



TESIS DOCTORAL

**DISEÑO DE UN SISTEMA EN TIEMPO REAL PARA EL PROCESAMIENTO A
BORDO DE IMÁGENES DE TELEDETECCIÓN UTILIZANDO ESTRUCTURAS DE
DATOS DE MATRIZ DIFUSA**

ALEJANDRO CRISTO GARCÍA

DPTO. TECNOLOGÍA DE LOS COMPUTADORES Y DE LAS COMUNICACIONES

2013



TESIS DOCTORAL

DISEÑO DE UN SISTEMA EN TIEMPO REAL PARA EL PROCESAMIENTO A BORDO DE IMÁGENES DE TELEDETECCIÓN UTILIZANDO ESTRUCTURAS DE DATOS DE MATRIZ DIFUSA

ALEJANDRO CRISTO GARCÍA

DPTO. TECNOLOGÍA DE LOS COMPUTADORES Y DE LAS COMUNICACIONES

Conformidad de los directores:

Fdo: Dra. Rosa M^a Pérez Utrero

Fdo: Dr. David Valencia Corrales

2013

Este proyecto ha sido financiado por el Gobierno de Extremadura y el Fondo Social Europeo a través de una beca-contrato de Formación de Personal Investigador (FPI). Expediente PRE08046.

GOBIERNO DE EXTREMADURA



UNIÓN EUROPEA

Fondo Social Europeo

Una manera de hacer Europa

*Dedicado a mi mujer Luz Marina,
a mis padres, a mi hermano,
y a mi familia política.*

Agradecimientos

A la Dra. Rosa M^a Pérez y al Dr. Pablo Martínez, por ser mis guías durante todos estos años, convirtiéndose en grandes apoyos y amigos, y por enseñarme el fantástico mundo de la investigación.

Al Dr. David Valencia Corrales, por sus aportaciones, ideas y sugerencias.

Al Gobierno de Extremadura y al Fondo Social Europeo, por creer en este proyecto y financiarlo.

Al Dr. Antonio Baeza y a Yolanda Miralles por acogerme en el Laboratorio de Radiactividad Ambiental de la Universidad de Extremadura (LARUEX) en mi nueva etapa postdoctoral.

Al Dr. Maciel Zortea, a Mario Fornaroli, a Juan Jesús Merino, al Dr. Pedro Luis Aguilar, a Carmen Cantero y al resto de antiguos y actuales integrantes del Grupo de Redes Neuronales y Procesamiento de la Señal (GRNPS), por amenizar siempre mi trabajo y por hacer del laboratorio mi segundo hogar.

Al Dr. José Moreno, de la Universidad de Valencia, por sus desinteresados consejos y por enseñarme tantas cosas sobre procesamiento espacial en nuestras pequeñas conversaciones en los congresos.

A José Antonio Gómez, del Instituto Nacional de Técnica Aeroespacial (INTA), por aclararme tantas dudas respecto a la adquisición de datos desde plataformas aerotransportadas y suministrarnos tanta información e imágenes para la elaboración del proyecto.

Al Dr. Robert Green, de la National Aeronautics and Space Administration (NASA), por la enorme simpatía que le caracteriza, y por tener siempre un hueco para transmitirme sus valiosísimos conocimientos, aún siendo uno de los hombres más solicitados y ocupados de este mundo.

Al Dr. Thomas Schmid, del Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas (Ciemat) y a la Dra. Magaly Koch, de Boston University, por ser tan excelentes personas y por ayudarme durante el desarrollo de mis algoritmos, especialmente en los implementados en las dos tarjetas FPGA.

Al Dr. Anthony Gualtieri (y familia), a Kevin Fisher y a la Dra. Jacqueline LeMoigne, de la National Aeronautics and Space Administration (NASA), por acogerme con tanto cariño durante mi estancia en Estados Unidos y por enseñarme no sólo aspectos relativos a mis investigaciones, sino también al tipo de vida norteamericano.

A Tom Flatley, Gary Crum y al resto del equipo de SpaceCube (National Aeronautics and Space Administration, NASA), por aceptarme en su laboratorio durante mi estancia en Estados Unidos, y por ayudarme tanto en la emulación de este dispositivo.

Al Dr. Diego Fernández y al Dr. Roberto Sabia, de la European Space Agency (ESA), y al Dr. Marco Talone, de SERCO, por su gran acogida durante mi estancia en Italia, y por enseñarme tantas cosas sobre adquisición de mediciones oceánicas y errores instrumentales.

A Xavier Gorroño, Sergi Pons, Clara Linés, Philippe Sigal y al resto de compañeros de la European Space Agency (ESA), por enseñarme cada rincón de Roma durante mi estancia en Italia.

Al Dr. Miguel Ángel Vega y al Dr. José María Granado, del grupo Arquitectura de Computadores y Diseño Lógico (ARCO) de la Universidad de Extremadura, por ayudarme con la interconexión entre las dos tarjetas FPGA.

A Agustín Escribano, del Personal de Administración y Servicios (PAS) de la Universidad de Extremadura, por ayudarme a implementar de manera física el protocolo de comunicaciones entre las dos tarjetas FPGA.

A mi mujer Luz Marina, por ser literalmente la luz que ilumina mi vida, por sufrir de manera directa los daños colaterales que supone desarrollar una tesis doctoral, y por apoyarme y animarme en los momentos duros.

A mis padres, hermanos, familia política y amigos, por su incondicional apoyo, interés en el proyecto, y por aguantar mis largas e inevitables ausencias.



Escuela Politécnica

La Dra. ROSA M^a PÉREZ UTRERO, Profesora Titular del Departamento de Tecnología de los Computadores y de las Comunicaciones de la Universidad de Extremadura, y el Dr. DAVID VALENCIA CORRALES, Personal Científico e Investigador de la Universidad de Extremadura,

HACEN CONSTAR:

Que esta memoria, titulada *Diseño de un sistema en tiempo real para el procesamiento a bordo de imágenes de teledetección utilizando estructuras de datos de Matriz Difusa*, presentada por D. ALEJANDRO CRISTO GARCÍA, resume su trabajo de Tesis Doctoral y, considerando que reúne todos los requisitos legales, autorizan su presentación y defensa para optar al grado de Doctor en Tecnologías Informáticas por la Universidad de Extremadura.

Cáceres, Julio 2013

A handwritten signature in black ink, appearing to read 'Rosa Mª Pérez Utrero'.

Dra. Rosa M^a Pérez Utrero

A handwritten signature in blue ink, appearing to read 'David Valencia Corrales'.

Dr. David Valencia Corrales



VALORACIÓN EXTERNA PARA LA MENCIÓN DE “DOCTOR EUROPEO”
EXTERNAL ASSESSMENT FOR THE “DOCTOR EUROPAEUS” MENTION

I.D.	Apellidos (Family Name)	Nombre (Name)
F574775 – Italy	SABIA	ROBERTO
Universidad o Centro de Investigación (University or Investigation Center)	Departamento (Department)	
	European Space Agency (ESA)	EOP-SA

Datos de la tesis (Doctoral Thesis Data)	
Autor-Doctorando (Doctoral Student-Author)	Alejandro Cristo García
Título de la Tesis - Thesis Title	Diseño de un sistema en tiempo real para el procesamiento a bordo de imágenes de teledetección utilizando estructuras de datos de Matriz Difusa Design of a real-time system for on-board processing of remote sensing imagery using Diffused Matrix data structures.

Informe (Report)
(Additional pages could be added if necessary).

To whom it may concern,

I hereby endorse the work developed by Mr. Alejandro Cristo and support his candidacy for attaining a European Ph.D.

In his Ph.D. manuscript, the candidate presents an alternative data format to store the information collected by the sensors onboard any kind of platform (ground-based, airborne and spaceborne) and highlights the possibility of georeferencing each collected measurement in real time through an integrated positioning system, such as a GPS/INS.

The proposed data structure, called Diffused Matrix Format (DMF), is based on the measurement as a base unit. This unit is represented as a record called Diffused Matrix Record (DMR), which stores all the information concerning the measurement. Since the DMR is composed by a sequence of fields, where the main attributes to store are the space coordinates of the measurement itself and its instrumental value, it is possible enlarging it to consider additional data, such as ancillary flags, values provided by other instruments or computational processes results. Each DMR is placed over a grid called Diffused Matrix (DM) which represents the scanned scene or an area of interest. This is performed according to the real coordinates provided by the positioning system, unlike the conventional georeferencing processes, where each measurement is geolocated by its pixel position in the image. In this way, the spatial errors produced by conventional geocorrection processes are noticeably reduced, being only constrained by the quality and the accuracy of the positioning systems used.

The experiments developed by Mr. Alejandro Cristo in his Ph.D. Thesis regard different computation analyses in order to study the performance of the DMF format, and several adaptations to real scientific issues and needs.

In the first part of the experiments, the DMF format shows very good results, mainly due to its capability of storing only real measurements acquired by the sensors, and avoiding common

techniques such as nearest neighbor assignments, customarily used by conventional geocorrection algorithms. This feature renders the DMF a very good strategy since the size of the final image, and therefore its computation times, are largely decreased, allowing a faster distribution to the scientist community. Even more interesting is the capability of considerably reducing the spatial errors, since methodologies such as data fusion and change detection present a better global performance (it is possible to identify each measurement in different scenes in a straightforward way) when compared to their analogous conventional algorithms.

In the second part of the experiments, an onboard computation system is designed, targeting as main reference the SpaceCube device designed by NASA. This computation system is composed by two FPGA cards interconnected, simulating a processing module. Each FPGA card implements a different methodology at hardware level, and they need to be synchronized in order to optimize the communications between them and the outer part of the system. To perform this task, a study of the kinds of connectors that is possible to use, according to the hardware execution times, has been carried out, showing which ones provide an optimized data transfer, and which ones, in turn, would saturate the internal buffers provoking a global system failure. The outputs of the designed system are provided as DMR records, so a final georeferenced image is available as soon as it is received in a ground-station, maximizing all the advantages that the DMF format offers.

As an alternative way of storing and representing satellite data, the DMF format has the adequate flexibility to be adopted in several processing and analysis techniques. The DMF format could be useful to generate real-time results and higher-level image products, avoiding ground computation resources and minimizing the latency from data acquisition to distribution to the scientific community. As such, the DMF can easily be considered as one of the most promising data format and constitute an important alternative to be considered in satellite image processing.

In 2012, I had the chance of working personally with Mr. Alejandro Cristo, on a specific research task dealing with the data processing of the ESA SMOS satellite over oceans. The candidate programming skills and his fast learning curve in a topic contiguous but slightly different from his main area of expertise proved to be very helpful in addressing the issues faced during this research project. In particular, his experience with data sampling and data storage revealed crucial to obtain meaningful results, currently collected into a manuscript and under revision in the Geoscience and Remote Sensing Letter (GRSL) journal. On the personal side, I favorably noticed his commitment and team spirit in an international working dimension.

I therefore fully support the work of Mr. Alejandro Cristo and I firmly believe he has the adequate skills and knowledge background to be awarded a European Ph.D.

Yours faithfully,
Roberto Sabia, Ph.D.
Research fellow. European Space Agency

Favorable evaluation: Yes

Frascati (Italy), August 29th, 2013
Signature:

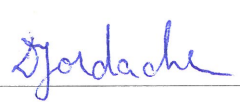




VALORACIÓN EXTERNA PARA LA MENCIÓN DE “DOCTOR EUROPEO”
EXTERNAL ASSESSMENT FOR THE “DOCTOR EUROPAEUS” MENTION

I.D.	Apellidos (Family Name)	Nombre (Name)
PX109606	Iordache	Marian-Daniel
Universidad o Centro de Investigación (University or Investigation Center)		Departamento (Department)
Flemish Institute for Technological Research (VITO)		Centre for Remote Sensing and Earth Observation Processes (TAP)

Datos de la tesis (Doctoral Thesis Data)	
Autor-Doctorando (Doctoral Student-Author)	Alejandro Cristo García
Título de la Tesis - Thesis Title	Diseño de un sistema en tiempo real para el procesamiento a bordo de imágenes de teledetección utilizando estructuras de datos de Matriz Difusa Design of a real-time system for on-board processing of remote sensing imagery using Diffused Matrix data structures.

Informe (Report)	
(Additional pages could be added if necessary).	
<p>The thesis introduces a new data structure intended to eliminate the typical spectral and spatial inaccuracies introduced by the geocorrection of data acquired by airborne and/or spaceborn sensors. The new data structure, called Diffused Matrix Format (DMF), takes into account the real location of each measurement acquired by the sensor. This format is based on an elemental unit called Diffused Matrix Record (DMR), which stores all the information corresponding to the acquired measurements, together with their UTM coordinates. In this way, this design allows to store the collected data with an accuracy limited to the GPS/INS system, avoiding conventional geocorrection processes which insert duplicated and false information. The thesis proves that DMF is an outstanding alternative in different fields, such as accurate spatial allocation of the measurements, data storage reduction, ground and onboard real-time processing and data fusion.</p> <p>The thesis is very well structured and very accurate from the point of view of mathematical formulation. Extensive studies are employed to prove the efficiency of the proposed data format, which is systematically compared to state of the art formats. It is shown that DMF requires less storage space compared to conventionally geocorrected data. Moreover, it can benefit from compression strategies to further reduce this space. The running time needed to process data in DMF format is also lower than the one corresponding to conventionally stored images. Simulations for data fusion applications using real images are also provided. Also, the DMF is validated as an effective onboard data acquisition format, having the potential to significantly reduce the data traffic.</p> <p>Overall, the ideas presented in the thesis are highly innovative, clearly exposed and deeply detailed. Regarding the content, the thesis is very well balanced between theoretical support and experimental evidence, while its novelty is undeniable. Given the high quality of the work, I enthusiastically recommend granting the author with the title “Doctor Europaeus”.</p>	
Informe favorable / Favorable evaluation: X Sí / Yes No / No	
Mol, Belgium, a 19 de Agosto de 2013	
Firma / Signature:	
Fdo: 	



esrin

Via Galileo Galilei
Casella Postale 64
00044 Frascati
Italy
T +39 06 9418 01
F +39 06 9418 0280
www.esa.int

Our ref. EOP-SA/0191/DFP-dr

Frascati, 6 June 2012

To whom it may concern,

This document certifies that Mr. Alejandro Cristo García, DNI 76037820-G, attended an internship at the European Space Research Institute (ESRIN) of the European Space Agency (ESA) from 1 March 2012 to 31 May 2012 (3 months), attached to the Science Strategy, Coordination and Planning Office (EOP-SA), Science, Applications and Future Technologies Department, Directorate of Earth Observation Programmes.

Mr. Alejandro Cristo García was the responsible for developing, testing and validating algorithms to improve/refine the processing of SMOS (Soil Moisture and Ocean Salinity) oceanographic data. The SMOS mission was launched in November 2009 to provide, over the oceans, remotely-sensed Sea Surface Salinity (SSS) estimates, with spatial and temporal coverage adequate for large scale oceanography. The work developed is placed at both Level 2 and Level 3 of the processing chain and devises adequate strategies to help in addressing some of the following issues:

- ⤴ Instrumental bias mitigation;
- ⤴ Wind speed / sea surface temperature auxiliary data uncertainties propagation;
- ⤴ Characterization of external noise sources (Galactic noise and Sun contamination);
- ⤴ Optimal averaging of the SSS product;
- ⤴ Data validation with various in-situ sources.

The results of this research have been presented and published in the proceedings of the international conference 2012 IEEE GOLD Remote Sensing Conference that took place in Rome on 4 June 2012.

A. Cristo, R. Sabia, M. Talone, D. Fernández-Prieto, M. Portabella, “Geophysical Effects at Increasing Latitudes in the SMOS L3 Salinity Products”, 2012 IEEE GOLD Remote Sensing Conference, Rome, Italy, 4 June 2012.

On behalf of the European Space Agency I would like to express my gratitude to Alejandro for his dedication, open collaboration and excellent work during his stage, where he



demonstrated an excellent capability to adapt to a new international environment and master in a short time new and challenging scientific and technical problems.

With my best regards,

A handwritten signature in black ink, appearing to be 'Diego Fernández Prieto', written over a faint, stylized graphic of a satellite or spacecraft.

Dr. Diego Fernández Prieto
EO Science, Applications and Future Technologies Department

Mención Doctorado Europeo

Esta tesis cumple los criterios para la obtención de la mención *Doctorado Europeo* concedida por la Universidad de Extremadura. Para ello se presentan los siguientes requisitos:

1. Aval mediante informes de idoneidad realizados por dos doctores de otros centros de investigación europeos.
 - Dr. Roberto Sabia. Investigador post-doctoral y Directivo Técnico en el centro European Space Research Institute de la European Space Agency (ESA), Frascati (Italia).
 - Dr. Marian Daniel Iordache. Investigador post-doctoral en el centro Flemish Institute for Technological Research (VITO), Mol (Belgium).
2. Estancia predoctoral realizada en otros países europeos:
 - Centro European Space Research Institute de la European Space Agency (ESA), Frascati (Italia), en el período del 1 de Marzo al 31 de Mayo de 2012 (3 meses), bajo la supervisión del Dr. Diego Fernández Prieto. Becado por la Ayuda Acción IV de la Universidad de Extremadura, Subprograma Banco Santander.
3. Documentación y defensa de la tesis realizados en dos idiomas europeos, español e inglés.
4. Constitución del tribunal por al menos un doctor procedente de un centro de educación superior europeo:
 - Dr. Mihai Datcu, del Deutsches Zentrum für Luft- und Raumfahrt (DLR, Centro Aeroespacial Alemán), Deutschland.
 - Dra. Paula Maria Ferreira de Sousa Cruz Redwei, de la Universidade de Lisboa, Portugal.

Índice general

1. Resumen / Summary	39
2. Motivaciones y objetivos	45
3. Introducción	49
3.1. Antecedentes.	50
3.2. Visión general de los sistemas de adquisición de imágenes.	51
3.2.1. Adquisición de datos desde plataformas y sensores.	56
3.2.2. Campañas de campo y sensores hiperespectrales en plataformas terrestres.	60
3.2.3. Sensores hiperespectrales en plataformas aero-transportadas.	62
3.2.4. Sensores hiperespectrales en plataformas espaciales.	64
3.2.5. Los efectos de la inestabilidad de la plataforma y el proceso de geocorrección.	66
3.2.6. Formatos de almacenamiento de imágenes hiperespectrales.	74
3.2.6.1. Formatos de datos.	75
3.2.6.2. Políticas en el almacenamiento del fichero de datos.	76
3.2.6.3. Formatos de datos estándar.	76
3.3. Sistemas de procesamiento en plataformas.	79
3.3.1. Sistemas basados en FPGA.	85
3.3.2. Sistemas FPGA a bordo de satélites.	88
3.3.2.1. SpaceCube como sistema de procesamiento modular.	90
3.3.2.1.1. La arquitectura Virtex-5.	95
3.3.2.1.2. El entorno de desarrollo de Xilinx.	100
4. Metodologías	101
4.1. Diffused Matrix Format (DMF).	105
4.1.1. Operaciones con el formato DMF.	111

4.1.1.1.	Creación de la matriz DMF.	111
4.1.1.2.	Visualización DMF.	114
4.1.1.3.	Almacenamiento DMF.	116
4.1.1.4.	Procesamiento DMF.	121
4.1.1.4.1.	Operaciones que no requieren vecindad.	121
4.1.1.4.2.	Operaciones que requieren vecindad.	123
4.1.2.	Adaptaciones y modificaciones del formato DMF.	125
4.1.2.1.	Adaptación del formato DMF para fusión de datos.	125
4.1.2.2.	Adaptación del formato DMF para detección de cambios.	126
4.2.	Simulación de un sistema SpaceCube.	128
4.2.1.	El módulo FIFO_E.	139
4.2.2.	El módulo FIFO_S.	141
5.	Resultados: Estudio del rendimiento del formato DMF	145
5.1.	Presentación de los datasets de imágenes utilizados en los experimentos.	146
5.2.	Construcción y visualización de la matriz DMF.	155
5.2.1.	Objetivos del experimento.	155
5.2.2.	Descripción del experimento.	158
5.2.3.	Resultados del experimento.	158
5.2.4.	Análisis de los resultados del experimento.	158
5.3.	Cálculo automático del tamaño de celda DMF.	162
5.3.1.	Objetivos del experimento.	162
5.3.2.	Descripción del experimento.	163
5.3.3.	Resultados del experimento.	165
5.3.4.	Análisis de los resultados del experimento.	173
5.4.	Espacio en disco.	178
5.4.1.	Objetivos del experimento.	178
5.4.2.	Descripción del experimento.	178
5.4.3.	Resultados del experimento.	179
5.4.4.	Análisis de los resultados del experimento.	188
5.5.	Tiempos de ejecución.	189
5.5.1.	Objetivos del experimento.	190

5.5.2.	Descripción del experimento.	190
5.5.3.	Resultados del experimento.	192
5.5.4.	Análisis de los resultados del experimento.	213
5.6.	Precisión espacial.	215
5.6.1.	Objetivos del experimento.	215
5.6.2.	Descripción del experimento.	215
5.6.3.	Resultados del experimento.	218
5.6.4.	Análisis de los resultados del experimento.	248
5.7.	Adaptación del formato DMF para campañas de campo.	253
5.7.1.	Objetivos del experimento.	254
5.7.2.	Descripción del experimento.	254
5.7.3.	Resultados del experimento.	256
5.7.4.	Análisis de los resultados del experimento.	257
5.8.	Fusión de datos.	258
5.8.1.	Objetivos del experimento.	258
5.8.2.	Descripción del experimento.	258
5.8.3.	Resultados del experimento.	263
5.8.4.	Análisis de los resultados del experimento.	265
5.9.	Detección de cambios.	269
5.9.1.	Objetivos del experimento.	269
5.9.2.	Descripción del experimento.	269
5.9.3.	Resultados del experimento.	271
5.9.4.	Análisis de los resultados del experimento.	272

6. Resultados: Adaptación del formato DMF a sistemas de adquisición 275

6.1.	Procesamiento a bordo en tiempo real.	276
6.1.1.	Objetivos del experimento.	276
6.1.2.	Descripción del experimento.	277
6.1.3.	Resultados del experimento.	278
6.1.4.	Análisis de los resultados del experimento.	281
6.2.	Primera aproximación a SpaceCube: El periférico MEDiC.	281
6.2.1.	Objetivos del experimento.	282

6.2.2.	Descripción del experimento.	282
	El módulo Sequential MEDiC.	283
	El módulo Parallel MEDiC.	284
6.2.3.	Resultados del experimento.	286
6.2.4.	Análisis de los resultados del experimento.	287
6.3.	Segunda aproximación a SpaceCube: Optimización del periférico MEDiC.	288
6.3.1.	Objetivos del experimento.	288
6.3.2.	Descripción del experimento.	288
	6.3.2.1. Primer diseño de la arquitectura hardware.	289
	6.3.2.2. Segundo diseño de la arquitectura hardware.	291
6.3.3.	Resultados del experimento.	292
6.3.4.	Análisis de los resultados del experimento.	293
6.4.	Simulación de un sistema SpaceCube para procesamiento a bordo.	295
6.4.1.	Objetivos del experimento.	295
6.4.2.	Descripción del experimento.	296
	6.4.2.1. Metodología desarrollada en la primera tarjeta FPGA.	297
	El módulo FIFO_E.	299
	El módulo Smooth.	300
	El módulo FIFO_S.	303
	6.4.2.2. Metodología desarrollada en la segunda tarjeta FPGA.	303
	El módulo FIFO_E.	305
	El módulo Statistics.	306
	El módulo FIFO_S.	308
6.4.3.	Resultados del experimento.	310
6.4.4.	Análisis de los resultados del experimento.	330
7.	Conclusiones / Conclusions	339
8.	Líneas futuras	347
8.1.	Consideración de la morfología del footprint.	347
8.2.	Visualización dinámica de la matriz DMF.	349
8.3.	Adaptación del formato DMF para procesamiento paralelo.	349
8.4.	Utilización del formato DMF para aislamiento de áreas de interés.	352

8.5. Optimización del módulo de procesamiento y de las comunicaciones.	353
A. Módulos de circuitería detallados	359
A.1. Segunda aproximación a SpaceCube. Optimización del periférico MEDiC.	359
A.1.1. Primer diseño de la arquitectura hardware.	359
El módulo MPMC Control Logic 1.	359
El módulo MPMC Controller.	360
El módulo MPMC Control Logic 2.	361
El módulo MPMC Control Logic 3.	362
El módulo Parallel MEDiC.	363
A.1.2. Segundo diseño de la arquitectura hardware.	364
El módulo MPMC Control Logic.	364
El módulo MPMC Controller.	365
A.2. Simulación de un sistema SpaceCube para procesamiento a bordo.	365
A.2.1. Metodología desarrollada en la primera tarjeta FPGA.	365
El módulo ToSingle256.	365
El módulo Filter.	366
El módulo Kernel.	369
El módulo Mean.	369
El módulo StdDev.	370
El módulo Threshold128.	371
El módulo Threshold.	372
El módulo FilterAdder.	373
El módulo ToInt256.	374
A.2.2. Metodología desarrollada en la segunda tarjeta FPGA.	375
El módulo ToSingle256.	375
El módulo Maximum.	375
El módulo Minimum.	376
El módulo Mean.	376
El módulo StdDev.	377
El módulo IR_Ratio.	377
El módulo NDVI.	377

Créditos de las imágenes	379
Bibliografía	381

Índice de figuras

3.1. Principales fuentes de imprecisiones presentes en las imágenes L0.	54
3.2. Representación del espectro electromagnético.	57
3.3. Firmas espectrales de diferentes minerales en el rango del visible e infrarrojo.	59
3.4. Componentes primarios de un sistema de teledetección electro-óptico.	59
3.5. Geometría de los sensores digitales.	60
3.6. Adquisición de mediciones y construcción del cubo de datos hiperespectral.	61
3.7. Adquisición de medidas utilizando sensores en Tierra.	61
3.8. Previsualización de una plataforma aero-transportada.	62
3.9. Previsualización de una plataforma espacial.	64
3.10. Proceso de construcción de una imagen L0 a partir de la adquisición de mediciones por un sensor de tipo <i>whiskbroom</i> desde la plataforma.	67
3.11. Relación entre la localización de las mediciones adquiridas por el sensor y la rejilla matricial que permitirá el correcto posicionamiento de los datos.	68
3.12. Proceso de geocorrección considerando y sin considerar el ángulo de vuelo.	70
3.13. Píxel con dos mediciones dentro de su área.	71
3.14. Políticas de selección de mediciones próximas cuando un píxel queda sin asignar.	71
3.15. Asignación de medición por vecino más próximo a un píxel vacío.	71
3.16. Error de localización arrastrado al considerar el píxel superior izquierda de la imagen geocorregida como referencia.	72
3.17. El problema del desplazamiento de las mediciones, respecto a su localización real, en el proceso de geocorrección.	73
3.18. Formatos básicos de ficheros de imágenes.	77
3.19. Políticas de almacenamiento del fichero de datos.	78
3.20. Procesador Fairchild F9450 (MIL-STD-1750 A).	82
3.21. Procesadores convencionales de propósito general RISC y CISC.	83
3.22. Diagrama de bloques de la arquitectura ERC32.	83
3.23. Diagrama de bloques de la arquitectura LEON3FT.	84

3.24. Vista interna de un chip FPGA.	86
3.25. Diagrama de un bloque CLB.	86
3.26. Comunicaciones entre los bloques CLB dentro de un chip FPGA.	87
3.27. Vista interna de un bloque DSP48E, un tipo de bloque DSP desarrollado por Xilinx.	88
3.28. Sistema FPGA híbrido Xilinx Virtex-II PRO.	88
3.29. Visualización general del sistema SpaceCube 1.0.	92
3.30. Especificaciones técnicas del sistema SpaceCube 1.0.	93
3.31. Diseño del sistema SpaceCube 1.5.	94
3.32. Visualización general del sistema SpaceCube 2.0.	96
3.33. Diseño del sistema SpaceCube 2.0 Mini.	96
3.34. Versiones del sistema SpaceCube.	97
3.35. Diagrama de bloques de la plataforma Virtex-5.	98
3.36. Vista detallada de los componentes presentes en las tarjetas FPGA Xilinx Virtex-5 ML507 y ML510.	99
4.1. Adquisición de mediciones sobre una determinada escena.	103
4.2. Representación de una misma escena utilizando diferentes tamaños de píxel.	104
4.3. Particionamiento de una determinada escena de manera irregular.	105
4.4. Estructuras básicas del formato DMF.	107
4.5. Alternativas al modo estándar de almacenamiento de los registros DMR en las respectivas celdas DMF.	109
4.6. Proceso de creación de la matriz DMF.	113
4.7. Visualización de una determinada banda de la imagen mediante el formato DMF.	115
4.8. Almacenamiento de la matriz DMF en fichero mediante la utilización de campos contadores.	118
4.9. Almacenamiento de la matriz DMF en fichero mediante la utilización de máscara y campos contadores.	119
4.10. Almacenamiento de la matriz DMF en fichero considerando únicamente la secuencia de registros DMR.	120
4.11. Procesamiento de la matriz DMF mediante operaciones que no requieren vecindad.	122
4.12. Optimización de la matriz DMF mediante la adición de una lista de accesos directos a todos los registros DMR almacenados.	123
4.13. Procesamiento de la matriz DMF mediante operaciones que sí requieren vecindad.	124
4.14. Adaptación del formato DMF para fusión de datos.	127
4.15. Adaptación del formato DMF para detección de cambios.	128

4.16. Contexto del sistema de procesamiento a bordo propuesto.	130
4.17. Emulación de un slice de procesamiento SpaceCube.	132
4.18. Configuración del paquete RS-232 8N1.	133
4.19. Configuración del paquete Ethernet.	134
4.20. Diagrama genérico del sistema hardware implementado en las tarjetas FPGA.	136
4.21. Diagrama detallado del periférico Metodología.	138
4.22. Diagrama general del módulo FIFO_E.	140
4.23. Diagrama general del módulo FIFO_S.	142
5.1. Recreación de la escena sintética (Banda 0) generada a partir de cuatro espectros referencia.	147
5.2. Obtención de espectros sobre la escena sintética para cada tipo de movimiento utilizando un sensor de tipo whiskbroom.	148
5.3. Imágenes L0 obtenidas mediante la simulación de vuelos sobre la escena sintética considerando los tipos de movimientos posibles con diferentes ángulos (Banda 0).	151
5.4. Imágenes geocorregidas a partir de los ficheros L0 e IGM obtenidos tras la simulación de adquisición (Banda 0).	152
5.5. Resolución espacial de las imágenes geocorregidas resultantes del vuelo simulado para cada uno de los movimientos.	153
5.6. Tamaño de píxel de las imágenes geocorregidas resultantes del vuelo simulado para cada uno de los movimientos.	154
5.7. Visualización del set de imágenes seleccionado para realizar los experimentos.	156
5.8. Visualización de las matrices DMF para el dataset de imágenes sintético (Banda 0).	159
5.9. Visualización de las matrices DMF para el dataset de imágenes real.	160
5.10. Proceso de estimación del tamaño de celda DMF.	164
5.11. Proceso de estimación del tamaño de celda DMF (Optimizado).	166
5.12. Tamaños de celda DMF en longitud y latitud proporcionados por la metodología de cálculo automático para cada uno de los movimientos de plataforma.	167
5.13. Visualización de las matrices DMF para el dataset de imágenes sintético, utilizando un tamaño de celda DMF calculado de manera automática (Banda 0).	168
5.14. Ángulos de la dirección del vuelo para el movimiento de plataforma yaw.	169
5.15. Tamaños de celda DMF en longitud y latitud proporcionados por la metodología de cálculo automático optimizada para el movimiento de plataforma yaw.	169
5.16. Visualización de las matrices DMF para el movimiento de plataforma yaw, utilizando un tamaño de celda DMF calculado de manera automática y considerando el sentido del vuelo (Banda 0).	169

5.17. Estudio de la distribución de los registros DMR sobre la matriz DMF para todos los ángulos de los tres tipos de movimiento de plataforma.	170
5.18. Estudio del número medio de registros DMR por celda DMF, para todos los ángulos de los tres tipos de movimiento de plataforma de acuerdo al tamaño de celda DMF.	171
5.19. Estudio del porcentaje de celdas DMF a Null, para todos los ángulos de los tres tipos de movimiento de plataforma de acuerdo al tamaño de celda DMF.	172
5.20. Visualización de las matrices DMF para el dataset de imágenes real, aplicando un tamaño de celda DMF calculado de manera automática por la metodología.	174
5.21. Estudio de la distribución de los registros DMR sobre la matriz DMF para todas las imágenes del dataset real.	176
5.22. Tamaños de disco de los formatos de almacenamiento DMF y convencional para cada uno de los movimientos de plataforma.	180
5.23. Imágenes geocorregidas a partir de los ficheros L0 e IGM correspondientes al movimiento de plataforma yaw, considerando el ángulo de vuelo (Banda 0).	182
5.24. Tamaños de disco de los formatos de almacenamiento DMF y convencional para el movimiento de plataforma yaw, considerando ángulo de vuelo y tamaño de celda DMF automático.	182
5.25. Tamaños de disco de los formatos de almacenamiento DMF y convencional para el dataset de imágenes real.	183
5.26. Visualización de las imágenes geocorregidas considerando el ángulo de vuelo para el dataset de imágenes real.	185
5.27. Tamaños de disco de los formatos de almacenamiento DMF y convencional para cada imagen del dataset real, considerando el ángulo de vuelo y un tamaño celda DMF automático.	187
5.28. Visualización de las imágenes del dataset sintético (formato geocorregido) tras aplicar la operación de umbralizado (Banda 0).	193
5.29. Visualización de las imágenes del dataset sintético (formato geocorregido) tras aplicar la operación de dilatación (Banda 0).	194
5.30. Visualización de las imágenes del dataset sintético (formato DMF) tras aplicar la operación de umbralizado (Banda 0).	195
5.31. Visualización de las imágenes del dataset sintético (formato DMF) tras aplicar la operación de dilatación (Banda 0).	196
5.32. Tiempos de ejecución de la operación de umbralizado para cada uno de los movimientos de plataforma del dataset sintético en sus versiones geocorregida, DMF y DMF optimizada.	197
5.33. Tiempos de ejecución de la operación de dilatación para cada uno de los movimientos de plataforma del dataset sintético en sus versiones geocorregida, DMF y DMF optimizada.	198
5.34. Tiempos de ejecución de la operación de umbralizado para el movimiento de plataforma yaw, considerando ángulo de vuelo y tamaño de celda DMF automático.	199
5.35. Tiempos de ejecución de la operación de dilatación para el movimiento de plataforma yaw, considerando ángulo de vuelo y tamaño de celda DMF automático.	199

5.36. Visualización de las imágenes del dataset real (formato DMF) tras aplicar la operación de umbralizado.	200
5.37. Visualización de las imágenes del dataset real (formato geocorregido) tras aplicar la operación de umbralizado.	202
5.38. Visualización de las imágenes del dataset real (formato geocorregido) tras aplicar la operación de umbralizado.	204
5.39. Visualización de las imágenes del dataset real (formato DMF) tras aplicar la operación de umbralizado.	206
5.40. Tiempos de ejecución de la operación de umbralizado para cada imagen del dataset real en sus versiones geocorregida, DMF y DMF optimizada.	208
5.41. Tiempos de ejecución de la operación de dilatación para cada imagen del dataset real en sus versiones geocorregida, DMF y DMF optimizada.	209
5.42. Tiempos de ejecución de la operación de umbralizado para cada imagen del dataset real en sus versiones geocorregida, DMF y DMF optimizada, considerando ángulo de vuelo y tamaño de celda DMF automático.	210
5.43. Tiempos de ejecución de la operación de dilatación para cada imagen del dataset real en sus versiones geocorregida, DMF y DMF optimizada, considerando ángulo de vuelo y tamaño de celda DMF automático.	211
5.44. Tiempos de ejecución de la operación de umbralizado para el movimiento de plataforma yaw, considerando un tamaño de celda DMF 10 veces menor que el proporcionado por el proveedor.	212
5.45. Tiempos de ejecución de la operación de dilatación para el movimiento de plataforma yaw, considerando un tamaño de celda DMF 10 veces mayor que el proporcionado por el proveedor.	212
5.46. Subimágenes de la escena de Cantera utilizadas para el estudio de la precisión espacial aplicado a un caso real.	217
5.47. Visualización de la localización de los píxeles de las imágenes geocorregidas de acuerdo a la manera en la fueron asignadas sus mediciones correspondientes (dataset sintético).	221
5.48. Porcentaje de tipo de asignación de mediciones presente en las imágenes geocorregidas y en sus versiones DMF análogas (dataset sintético).	222
5.49. Visualización de la localización de los píxeles de las imágenes geocorregidas de acuerdo a la manera en la fueron asignadas sus mediciones correspondientes, teniendo en cuenta el ángulo de vuelo y el tamaño de celda DMF calculado de manera automática (movimiento de plataforma yaw).	223
5.50. Porcentaje de tipo de asignación de mediciones presente en las imágenes geocorregidas y en sus versiones DMF análogas (dataset sintético), teniendo en cuenta el ángulo de vuelo y el tamaño de celda DMF calculado de manera automática (movimiento de plataforma yaw).	223
5.51. Visualización de las distancias entre el posicionamiento geocorregido de cada medición y el proporcionado por el sistema de posicionamiento (dataset sintético).	224
5.52. Distancias medias globales de las posiciones geocorregidas para cada píxel respecto a la posición original de la medición correspondiente (dataset sintético).	225

5.53. Visualización de las distancias entre el posicionamiento geocorregido de cada medición y el proporcionado por el sistema de posicionamiento (dataset sintético), teniendo en cuenta el ángulo de vuelo (movimiento de plataforma yaw).	226
5.54. Distancias medias globales de las posiciones geocorregidas para cada píxel respecto a la posición original de la medición correspondiente (dataset sintético), teniendo en cuenta el ángulo de vuelo (movimiento de plataforma yaw).	226
5.55. Visualización de la frecuencia de selección de las mediciones originales para la construcción de la imagen geocorregida correspondiente (dataset sintético).	227
5.56. Porcentaje de mediciones originales no seleccionadas para la construcción de la imagen geocorregida correspondiente (dataset sintético).	228
5.57. Número de veces que una medición ha sido seleccionada para para la construcción de la imagen geocorregida correspondiente (dataset sintético).	229
5.58. Visualización de la frecuencia de selección de las mediciones originales para la construcción de la imagen geocorregida correspondiente (dataset sintético), teniendo en cuenta el ángulo de vuelo (movimiento de plataforma yaw).	230
5.59. Porcentaje de mediciones originales no seleccionadas para la construcción de la imagen geocorregida correspondiente (dataset sintético), teniendo en cuenta el ángulo de vuelo (movimiento de plataforma yaw).	230
5.60. Número de veces que una medición ha sido seleccionada para para la construcción de la imagen geocorregida correspondiente (dataset sintético), teniendo en cuenta el ángulo de vuelo (movimiento de plataforma yaw).	230
5.61. Visualización de la localización de los píxeles de las imágenes geocorregidas de acuerdo a la manera en la fueron asignadas sus mediciones correspondientes (dataset real). . .	231
5.62. Porcentaje de tipo de asignación de mediciones presente en las imágenes geocorregidas y en sus versiones DMF análogas (dataset real), teniendo en cuenta el ángulo de vuelo y el tamaño de celda DMF calculado de manera automática.	233
5.63. Porcentaje de tipo de asignación de mediciones presente en las imágenes geocorregidas y en sus versiones DMF análogas (dataset real).	235
5.64. Porcentaje de tipo de asignación de mediciones presente en las imágenes geocorregidas y en sus versiones DMF análogas (dataset real), teniendo en cuenta el ángulo de vuelo y el tamaño de celda DMF calculado de manera automática.	235
5.65. Visualización de la localización de los píxeles de las imágenes geocorregidas de acuerdo a la manera en la fueron asignadas sus mediciones correspondientes (dataset real). . .	236
5.66. Distancias medias de las posiciones geocorregidas para cada píxel respecto a la posición original de la medición correspondiente (dataset real).	238
5.67. Visualización de las distancias entre el posicionamiento geocorregido de cada medición y el proporcionado por el sistema de posicionamiento (dataset real), teniendo en cuenta el ángulo de vuelo.	239
5.68. Distancias medias de las posiciones geocorregidas para cada píxel respecto a la posición original de la medición correspondiente (dataset real), teniendo en cuenta el ángulo de vuelo.	241

5.69. Visualización de la frecuencia de selección de las mediciones originales para la construcción de la imagen geocorregida correspondiente (dataset real).	242
5.70. Porcentaje de mediciones originales no seleccionadas para la construcción de la imagen geocorregida correspondiente (dataset real).	244
5.71. Número de veces que una medición ha sido seleccionada para para la construcción de la imagen geocorregida correspondiente (dataset real).	244
5.72. Visualización de la frecuencia de selección de las mediciones originales para la construcción de la imagen geocorregida correspondiente (dataset real), teniendo en cuenta el ángulo de vuelo.	245
5.73. Porcentaje de mediciones originales no seleccionadas para la construcción de la imagen geocorregida correspondiente (dataset real), teniendo en cuenta el ángulo de vuelo.	247
5.74. Número de veces que una medición ha sido seleccionada para para la construcción de la imagen geocorregida correspondiente (dataset real), teniendo en cuenta el ángulo de vuelo.	247
5.75. Visualización de los resultados de la precisión espacial para las subescenas de Cantera.	248
5.76. Formato DMF adaptado para campañas de campo.	255
5.77. Visualización de la matriz DMF adaptada a campañas de campo [Banda 10].	256
5.78. Visualización de la matriz DMF de acuerdo a los tipos de materiales analizados.	257
5.79. Representación de los instrumentos diseñados para la misión FLEX / Sentinel-3.	259
5.80. Información técnica de las líneas de escaneo de los instrumentos de Sentinel-3.	260
5.81. Visualización de la segunda imagen L0 sobre la escena de Cantera (Cantera2) [Banda 0].	261
5.82. Líneas de adquisición simuladas para los instrumentos OLCI y SLSTR, considerando caso $y=0$	262
5.83. Imágenes L0 simuladas para los diferentes orígenes.	263
5.84. Versiones DMF de las imágenes simuladas.	264
5.85. Análisis visual de la matriz DMF resultante tras introducir todas las mediciones adquiridas por FIMAS, OLCI, SLSTR_Nadir y SLSTR_Backward.	266
5.86. Densidad de mediciones de la matriz DMF resultante tras introducir todas las mediciones adquiridas por FIMAS, OLCI, SLSTR_Nadir y SLSTR_Backward.	267
5.87. Identificación de los orígenes de las mediciones en la matriz DMF resultante tras la fusión de datos.	268
5.88. Segunda subimagen de Cantera utilizada para el estudio de la detección de cambios.	270
5.89. Imagen DMF con las todas las mediciones de la primera subimagen (Timestamp=Instante1) y de la segunda subimagen (Timestamp=Instante2) almacenadas.	270
5.90. Visualización de los cambios detectados para las subescenas de Cantera.	271
6.1. Diseño del sistema de detección de nubes a bordo de satélite, con el formato DMF como formato de datos básico.	277

6.2. Visualización de la imagen Mediterráneo en formato DMF tras considerar diferentes niveles de presencia de nubes.	279
6.3. Diagrama del sistema hardware con el periférico MEDiC adjunto.	283
6.4. Circuitos diseñados para el cálculo de la distancia Euclídea multi-espectral.	285
6.5. Procesos de ejecución.	286
6.6. Diagrama de la conexión Memoria - Periférico - Procesador para el Diseño 1, en donde el periférico MEDiC hace uso de 7 puertos MPMC mediante una conexión NPI de 64 bits.	290
6.7. Diagrama del periférico MEDiC, mostrando los módulos del control del componente MPMC, las conexiones a los 7 puertos MPMC, y los registros de comunicaciones convencionales.	291
6.8. Diagrama de la conexión Memoria - Periférico - Procesador para el Diseño 2, en donde el periférico MEDiC hace uso de 4 puertos MPMC en modo Burst Read para la lectura de los datos, y un quinto para la escritura del resultado. Todos los puertos usan conexiones NPI de 64 bits.	293
6.9. Diagrama del periférico MEDiC, mostrando los módulos del control del componente MPMC, las conexiones a los 5 puertos MPMC, y los registros de comunicaciones convencionales.	294
6.10. Adaptación del diseño general para la implementación de las metodologías de suavizado espectral y cálculos estadísticos.	296
6.11. Diagrama detallado del periférico Smooth.	299
6.12. Diagrama del módulo FIFO_E.	300
6.13. Diagrama del módulo Smooth.	301
6.14. Diagrama del módulo FIFO_S.	304
6.15. Diagrama detallado del periférico Statistics.	306
6.16. Diagrama del módulo Statistics.	307
6.17. Diagrama del módulo FIFO_S.	309
6.18. Conexión física de las dos tarjetas FPGA emulando el sistema SpaceCube.	311
6.19. Ejemplo de resultado proporcionado por cada una de las tarjetas FPGA (Espectro (369,29) de la escena de Barrax).	313
6.20. Señalización de los puntos críticos que son necesarios sincronizar para el correcto funcionamiento del sistema propuesto.	315
6.21. Tasas de procesamiento y transmisión de mediciones para los puntos críticos del sistema.	320
6.22. Porcentaje de tiempo ocioso para cada uno de los dispositivos del sistema en función del tipo de conexión de entrada de datos seleccionado.	321
6.23. Visualización en formato DMF de los resultados obtenidos con el sistema SpaceCube para la escena de Cuprite.	325

6.24. Visualización en formato DMF de los resultados obtenidos con el sistema SpaceCube para la escena de Barrax.	326
6.25. Visualización en formato DMF de los resultados obtenidos con el sistema SpaceCube para la escena de Mediterráneo.	327
6.26. Visualización en formato DMF de los resultados obtenidos con el sistema SpaceCube para la escena de GMéxico.	328
6.27. Visualización en formato DMF de los resultados obtenidos con el sistema SpaceCube para la escena de Cantera.	329
6.28. Localización de la zona de aterrizaje Lida Junction Airstrip, único foco de alta vegetación en el desierto perteneciente a la escena de Cuprite.	337
8.1. Consideración de la morfología del footprint en el formato DMF.	348
8.2. Visualización interactiva de una determinada celda DMF de la matriz DMF (vista dual).	350
8.3. Representación de una escena simple mediante la estructura arbórea Quad-tree.	350
8.4. Adaptación del formato DMF para el procesamiento paralelo basado en el particionamiento irregular de la escena mediante variantes del árbol Quad-tree.	352
8.5. Representación de áreas de interés mediante el formato DMF para la escena de Mediterráneo (Banda 4).	354
8.6. Optimización del diseño del sistema basado en SpaceCube propuesto.	356
A.1. Diagrama del módulo MPMC Control Logic 1, responsable de obtener de memoria los 10 valores de espectro a través de cada puerto MPMC.	360
A.2. Señales de control del componente MPMC.	361
A.3. Diagrama del módulo MPMC Control Logic 2, responsable de obtener de memoria los 8 valores de espectro a través de cada puerto MPMC.	362
A.4. Diagrama del módulo MPMC Control Logic 3, responsable de obtener de memoria los 8 valores de espectro a través de cada puerto MPMC, y de escribir el resultado del cálculo de la distancia Euclídea multi-espectral.	363
A.5. Circuito paralelo optimizado para el cálculo de la distancia Euclídea multi-espectral.	364
A.6. Diagrama del módulo MPMC Control Logic, responsable de obtener de memoria los 16 valores de espectro a través de cada puerto MPMC y usando el modo Burst Read.	366
A.7. Señales de control del componente MPMC para realizar la lectura de un bloque de palabras de 64 bits de memoria, usando el modo Burst Read.	367
A.8. Diagrama del módulo ToSingle256.	367
A.9. Diagrama del módulo Filter.	368
A.10. Diagrama del módulo Kernel.	369
A.11. Diagrama del módulo Mean.	370
A.12. Diagrama del módulo StdDev.	372

A.13.Diagrama del módulo Threshold128.	373
A.14.Diagrama del módulo Threshold.	373
A.15.Diagrama del módulo FilterAdder.	374
A.16.Diagrama del módulo ToInt256.	375
A.17.Diagrama del módulo Maximum.	376
A.18.Diagrama del módulo Minimum.	377
A.19.Diagrama del módulo IR_Ratio.	378
A.20.Diagrama del módulo NDVI.	378

Índice de tablas

3.1. Bandas del espectro electromagnético.	58
3.2. Resumen de las características generales de los sensores más populares a bordo de plataformas aero-transportadas.	64
3.3. Resumen de las características generales de los sensores más populares a bordo de plataformas espaciales.	66
4.1. Diferencias principales entre las tarjetas Xilinx Virtex-5 ML507 y Virtex-5 ML510, considerando el sistema hardware reconfigurable propuesto.	131
5.1. Configuración y especificaciones técnicas de los vuelos realizados sobre la escena sintética.	150
5.2. Detalle de las características de las imágenes que conforman el dataset sintético.	155
5.3. Tamaños de celda DMF calculados de manera automática por la metodología y ángulos de vuelo para las imágenes del dataset real.	173
5.4. Diferencia de tamaños de espacio en disco (MB) de todos los formatos de almacenamiento respecto a la política DMF3 para cada movimiento de plataforma.	181
5.5. Diferencia de tamaños de espacio en disco (MB) de todos los formatos de almacenamiento respecto a la política DMF3 para el movimiento de plataforma yaw, considerando ángulo de vuelo y tamaño de celda DMF automático.	182
5.6. Diferencia de tamaños de espacio en disco (MB) de todos los formatos de almacenamiento respecto a la política DMF3 para el dataset real (dif).	184
5.7. Porcentajes de incremento de tamaños de espacio en disco de todos los formatos de almacenamiento respecto a la política DMF3 para el dataset real (Δ %).	184
5.8. Diferencia de tamaños de espacio en disco (MB) de todas las imágenes correspondientes al dataset real respecto a la política DMF3 (<i>dif</i>), considerando el ángulo de vuelo y un tamaño de celda DMF automático.	184
5.9. Porcentajes de incremento de tamaños de espacio en disco de todas las imágenes correspondientes al dataset real respecto a la política DMF3 (Δ %), considerando el ángulo de vuelo y un tamaño de celda DMF automático.	184
5.10. Valores de umbral $\overline{\ x\ ^2}$ seleccionados para cada uno de los datasets de imágenes.	191
5.11. Descripción de las imágenes utilizadas para el estudio de la precisión espacial aplicado a un caso real.	217

5.12. Ángulo espectral medio de las mediciones almacenadas en las imágenes geocorregidas y DMF, respecto a sus análogas en la imagen ground-truth.	248
5.13. Comparación de las características y funcionalidades de los diferentes sensores.	259
5.14. Detalle de las características de la segunda imagen L0 sobre la escena de Cantera (Cantera2).	261
5.15. Descripción de las subimágenes, correspondientes al segundo vuelo, utilizadas y generadas para el estudio de la detección de cambios.	271
5.16. Ángulo espectral medio obtenido tras realizar la detección de cambios usando versiones geocorregidas y el formato DMF.	271
6.1. Tamaños de la información adquirida por el satélite de acuerdo al nivel de presencia de nubes, incluyendo la reducción obtenida cuando el formato DMF se ha utilizado.	280
6.2. Tiempos empleados y speed-up de los circuitos secuencial y paralelo en el cálculo de la distancia Euclídea multi-espectral para espectros de 64 bandas.	287
6.3. Tiempos empleados y speed-up de los diseños propuestos en el cálculo de la distancia Euclídea multi-espectral para espectros de 64 bandas.	293
6.4. Bandas seleccionadas como valores R y NIR para cada una de las imágenes del dataset real.	304
6.5. Tiempos de ejecución de cada uno de los módulos de la metodología hardware propuesta.	314
6.6. Tiempos de ocio de cada una de las conexiones de salida de acuerdo a la conexión de entrada seleccionada.	323
6.7. Tiempo de procesamiento de una única medición en el sistema SpaceCube simulado.	330
6.8. Tiempos de procesamiento de las imágenes completas en el sistema SpaceCube simulado.	331
6.9. Porcentajes de dedicación a comunicaciones y procesamiento en el sistema SpaceCube simulado.	332

Capítulo 1

Resumen / Summary

El campo científico de la *teledetección* se centra en la observación y estudio de la corteza terrestre con el objetivo de entender el origen y funcionamiento de la dinámica de nuestro planeta, así como de garantizar la seguridad de los seres vivos que habitan en ella, y mejorar su calidad de vida. Esto implica el análisis de fenómenos meteorológicos, geológicos, atmosféricos, etc., para comprender las condiciones que los generan e influyen en su comportamiento. De esta forma, se hace posible el diseño e implementación de algoritmos que permitan una temprana detección y simulación de la evolución ante catástrofes naturales de gran magnitud, optimizando a su vez la generación de planes de prevención asistidos o automáticos y la estimación de daños tanto a priori, como a posteriori. Estas catástrofes naturales abarcan desde terremotos de gran magnitud, hasta maremotos, actividades volcánicas, componentes radiactivos presentes en la atmósfera, o incendios forestales, entre otros.

Aparte de los fenómenos naturales, otros parámetros adquieren gran importancia para este campo científico, como por ejemplo el social-demográfico, estudiando cómo el ser humano altera las condiciones actuales del planeta y su influencia en el cambio climático; o el militar, analizando el comportamiento y acciones de países o civilizaciones hostiles, y siempre teniendo como objetivo optimizar la protección de la población civil. El primer aspecto abarca diferentes técnicas como por ejemplo el seguimiento del crecimiento de las ciudades, detección de niveles de generación de contaminación por las mismas, o análisis de riesgos respecto a la situación de las poblaciones, bases, o nuevos asentamientos. El segundo aspecto implica métodos de detección de vehículos de guerra o armamento de destrucción masiva.

Para ello, se hace uso de sensores especiales utilizados por analistas en campañas de campo, o a bordo de plataformas situadas en vehículos terrestres, torretas, aviones o satélites. La tecnología que acompaña a estos sensores puede variar, así como sus características y geometrías, siendo los más utilizados aquellos que permiten el análisis espectral de los elementos de interés en diferentes frecuencias del espectro electromagnético. Los sensores utilizados en campañas de campo o a bordo de plataformas terrestres (como vehículos o torretas) permiten el estudio detallado de una determinada región muy localizada, así como la recolección de muestras espectrales para una optimización de las metodologías utilizadas. Las plataformas aerotransportadas están situadas en aviones o helicópteros, y permiten la recolección de datos de un área de interés de gran extensión, con una resolución espacial inigualable. Por otro lado, las plataformas espaciales están situadas a bordo de satélites, permitiendo el análisis de la corteza y atmósfera terrestre a un nivel global, con una resolución temporal y accesibilidad a regiones inalcanzables por expertos o vehículos tripulados.

Para el correcto funcionamiento de las metodologías y el cumplimiento de los objetivos de la misión, se hace necesario la optimización en la que la información correspondiente al área de interés es adquirida, procesada, representada, almacenada y distribuida.

Un primer parámetro a controlar es la estructura de datos utilizada, la cual debe permitir

la adecuada localización espacial de las mediciones adquiridas durante el proceso denominado *geocorrección*, garantizando la consistencia de la información y minimizando los errores espaciales y espectrales derivados de las características de los sensores y de las condiciones bajo las que han estado trabajando. Además, dicha estructura de datos debe permitir un procesamiento rápido, sencillo y eficaz, ya que la cantidad de mediciones recogidas puede ser elevada y multi-dimensional.

Un segundo parámetro abarca la manera en la que se almacenan y distribuyen los datos obtenidos de los sensores, por lo que un formato de datos óptimo debe implicar un salvado y cargado de la información de manera rápida, así como una minimización en el consumo de espacio en disco.

El tercer parámetro corresponde al diseño y configuración de los sistemas de adquisición y procesamiento, los cuales adquieren más importancia cuando implementan metodologías ejecutadas a bordo para la obtención de resultados en tiempo real. Los sistemas de procesamiento, responsables de analizar y generar un resultado a partir de las mediciones obtenidas por los sensores, deben ofrecer una respuesta rápida y fiable, así como una completa integración con el sistema de adquisición y otros dispositivos adicionales presentes en la plataforma. Esta integración requiere que todas las comunicaciones realizadas, tanto a un nivel interno como externo, estén perfectamente sincronizadas y permitan una transferencia de los datos en un tiempo mínimo para evitar *cuellos de botella*. Un tipo de dispositivo que está gozando de gran auge hoy en día son los Field Programmable Gate Array (FPGA) debido a las múltiples ventajas que ofrecen. Por un lado, permiten el desarrollo de las metodologías a un nivel de hardware, por lo que comparten las posibilidades correspondientes al nivel físico, como la estabilidad de los módulos y su velocidad. Por otro lado, se trata de una tecnología reconfigurable, por lo que permite la corrección de errores o la actualización de circuitos ya implementados. Estas características, sumadas a un bajo coste económico y un consumo de energía mínimos, y una gran adaptabilidad a técnicas hardware de detección y corrección de errores, las hace especialmente atractivas sobre todo para el procesamiento en entornos radiactivos, como en plataformas espaciales situadas en las capas superiores de la atmósfera, o fuera de ella.

Por tanto, teniendo en cuenta estos aspectos, este documento de tesis pretende proponer una solución a los problemas que surgen a la hora de adquirir, procesar, representar, almacenar y distribuir los datos obtenidos por los sensores, así como la optimización de las metodologías que permiten la realización de estas tareas. La línea principal de la propuesta se basa en el diseño e implementación de una estructura y formato de datos que permita una perfecta integración de la información, minimizando el error espacial derivado de agentes externos, como las turbulencias a las que está sometida la plataforma durante la adquisición de datos, o las imprecisiones espaciales procedentes de técnicas de preprocesamiento, como por ejemplo, la geocorrección. Esta estructura y formato de datos, denominada Diffused Matrix Format (DMF), se basa en la medición como unidad básica de almacenamiento, la cual es representada por una estructura con un número de campos variable denominada Diffused Matrix Record (DMR). El registro DMR almacena la localización espacial proporcionada por el sistema de posicionamiento presente en la plataforma, así como los valores instrumentales procedentes de los sensores. Los registros DMR son posicionados sobre una matriz georreferenciada, ocupando su celda DMF correspondiente de acuerdo a su localización espacial. Al igual que las imágenes, este sistema de representación permite disfrutar de las ventajas que proporcionan las matrices, como su sencillo procesamiento (basado en ecuaciones matemáticas) y el acceso directo. Además, el diseño permite el almacenamiento de ninguno, uno o más registros DMR en una misma celda DMF, por lo que se garantiza la utilización del 100 % de los datos obtenidos por los sensores, evitando la duplicación y preservación de los mismos. Al utilizar únicamente la información real recogida por los instrumentos, el formato DMF supone una minimización del consumo de espacio en disco, lo que supone una optimización del almacenamiento de la información y su distribución.

El formato DMF ha sido diseñado con el objetivo de ser integrada de manera fácil y sencilla a las diferentes técnicas de procesamiento de imágenes de teledetección. El hecho de almacenar los datos obtenidos por los sensores en una estructura matricial cuyas celdas pueden obtener un número de mediciones variables permite una fácil adaptación a metodologías más complejas mediante la simple adición de nuevos campos al registro DMF. De esta manera, es posible la inserción de datos procedentes

de diferentes orígenes e instantes en la misma matriz DMF, potenciando la eficiencia de métodos basados en la fusión de datos y la detección de cambios.

Por otro lado, y como se ha comentado con anterioridad, el sistema de procesamiento adquiere gran importancia a la hora de generar resultados a partir de los datos obtenidos por los instrumentos presentes en la plataforma. Por lo tanto, se ha realizado un detallado estudio de cómo integrar los modelos de computación con la estructura y formato de datos propuestos. Para ello se propone utilizar dispositivos FPGA como sistemas de procesamiento principales o auxiliares, se ha considerado como modelo referencia el sistema modular SpaceCube, desarrollado por la NASA, y compuesto principalmente por módulos de procesamiento conformados por dos tarjetas FPGA Xilinx Virtex-5 interconectadas. El módulo hardware propuesto hará uso de este mismo tipo de tarjetas, y proporcionará, en todo caso, los resultados generados en formato DMF. De esta manera, se ha realizado, primeramente, el diseño de un esquema de sincronización y transferencia de datos entre el sistema de procesamiento y el resto de dispositivos, a un nivel global. Posteriormente, se implementaron técnicas básicas de procesamiento a un nivel de hardware, para analizar cómo el formato DMF influye en los tiempos de ejecución de acuerdo a los diferentes tipos de conexiones utilizados para interconectar las dos tarjetas FPGA. Este experimento es muy importante, ya que detecta los puntos críticos del sistema, en los cuales una mala configuración puede suponer una saturación de los *buffers*, generando *cuellos de botella* a un nivel tanto interno como externo, y entorpeciendo el correcto procesamiento de los datos en tiempo real.

En resumen, el formato DMF propuesto no sólo permite solucionar los problemas más característicos y la optimización de técnicas basadas en la teledetección mediante la utilización de una estructura matricial a nivel de software, sino que su integración a sistemas hardware reconfigurables, como SpaceCube, potencia sus funcionalidades a la vez de optimizar la manera en la que los resultados son generados.

El resto del documento de tesis está estructurado de la siguiente manera: El Capítulo 2 detalla los principales motivos que llevaron a la ejecución de este proyecto, así como los objetivos definidos para su elaboración; el Capítulo 3 realiza un repaso por los antecedentes en el campo de la teledetección, presentando una introducción a los problemas principales en este campo, generalmente derivados del proceso de geocorrección, y dedicando especial atención al tipo de plataformas espaciales, y en concreto, al sistema SpaceCube; el Capítulo 4 presenta el formato DMF y su adaptación a diferentes necesidades y técnicas, así como un modelo general de sistema de computación basado en tarjetas FPGA que integra este formato como política de representación de los resultados generados; el Capítulo 5 presenta los *datasets sintético y real* y los experimentos realizados para analizar el funcionamiento y eficiencia del formato DMF, algunos abarcando técnicas y problemas derivados del procesamiento de imágenes convencionales, y otros resolviendo problemas específicos procedentes de necesidades de diferente índole; el Capítulo 6 incluye los experimentos elaborados y los resultados obtenidos cuando el formato DMF se adapta e integra en un sistema hardware reconfigurable con metodologías específicas implementadas; el Capítulo 7 realiza un análisis general de los resultados obtenidos, extrayendo las principales consideraciones y conclusiones; el Capítulo 8 presenta las líneas de investigación que han quedado abiertas y en las que se trabajarán en un futuro próximo; y finalmente el Anexo A presenta en detalle la circuitería de las metodologías implementadas a nivel de hardware.

The scientific field of *remote sensing* focuses on the observation and study of the Earth's crust, with the aim of understanding the origin and functioning of the dynamics of the planet, as well as ensuring the safety of the living beings that inhabit it and improving their quality of life. This involves analysing meteorological, geological and atmospheric phenomena in order to understand the conditions that cause them and influence their behaviour. This makes it possible to design and implement algorithms that enable early detection and evolution simulation with respect to natural

disasters of great magnitude, while optimising the creation of a priori and a posteriori assisted and automatic prevention plans and damage assessments. These natural disasters range from large magnitude earthquakes and tsunamis to volcanic activity, radioactive components present in the atmosphere and forest fires.

As well as natural phenomena, other parameters take on importance for this scientific field, such as social demographics involving the study of how humans alter the conditions of the planet and how this influences climate change, and military parameters involving analysis of the behaviour and actions of hostile countries or civilisations, always with the aim of optimising the protection of the civil population. The first aspect includes different techniques such as monitoring the growth of cities and their pollution levels, and risk analysis with respect to the circumstances of the population, bases and new settlements. The second aspect involves methods for detecting vehicles of war or weapons of mass destruction.

To this end, analysts use special sensors in field campaigns or on board platforms located on land vehicles, pylons, aircraft or satellites. The technology that accompanies these sensors can vary, as can its characteristics and geometry, and the most used is that which enables spectral analysis of the elements of interest at different frequencies of the electromagnetic spectrum. The sensors used in field campaigns or on board land platforms (such as vehicles or pylons) enable a detailed study of a certain highly-localised region and collection of spectral samples for the optimisation of the methodologies used. Airborne platforms are located on aircraft or helicopters and enable data to be collected from a large area of interest with an unsurpassable spatial resolution. Space platforms are located on satellites and enable analysis of the Earth's crust and atmosphere on a global level with a temporal resolution and accessibility to regions that cannot be reached by experts or manned vehicles.

To ensure the proper functioning of the methodologies and fulfilment of the mission's objectives, optimisation is necessary in which information relating to the area of interest is acquired, processed, represented, stored and distributed.

The first main parameter to control is the structure of the data used, which must enable appropriate spatial location of the measurements acquired during the process known as *geocorrection* to ensure consistency of information and minimisation of spatial and spectral errors arising from the characteristics of the sensors and the conditions under which they have been operating. In addition, this data structure must enable rapid, simple and effective processing, since the amount of measurements collected can be extensive and multi-dimensional.

The second main parameter is the manner in which the data obtained from the sensors is stored and distributed by means of an optimum data format that enables information to be saved and loaded quickly and disk space consumption minimised.

The third parameter is the design and configuration of the acquisition and processing systems, which are particularly important when implementing methodologies executed on board to obtain real-time results. The processing systems responsible for analysing and generating a result from the measurements obtained by the sensors must provide a rapid and reliable response and be completely integrated with the acquisition system and other additional devices located on the platform. For this integration, all internal and external communications have to be perfectly synchronised and must enable data transfer to take place in the shortest time possible to prevent *bottlenecks*. One such device that is becoming increasingly used today is the Field Programmable Gate Array (FPGA) on account of the many advantages it offers. One main advantage is that, as it enables the development of methodologies at hardware level, it shares the possibilities corresponding to the physical level, such as module stability and speed. Another advantage is that it consists of reconfigurable technology, thereby enabling errors to be corrected and existing circuits to be updated. These features, combined with low cost and minimal power consumption, and great adaptability to hardware detection techniques and error correction, makes it particularly appealing for processing in radioactive environments, such as space platforms located in the upper layers of the atmosphere or beyond.

Taking these aspects into account, this thesis document seeks to propose a solution to the problems that arise from acquiring, processing, representing, storing and distributing data obtained from sensors and suggest ways to optimise the methodologies that enable these tasks to be performed. The main idea behind the proposal involves designing and implementing a data structure and format that enables information to be perfectly integrated, thereby minimising spatial errors caused by external agents such as turbulence, which can affect platforms during data acquisition, or spatial inaccuracies resulting from pre-processing techniques, such as geocorrection. This structure and data format, known as the Diffused Matrix Format (DMF), is based on measurement as the basic unit of storage, which is represented by a structure with a variable number of fields, called a Diffused Matrix Record (DMR). The DMR stores the spatial location provided by the positioning system located on the platform and the instrumental values from the sensors. DMRs are positioned on a georeferenced matrix and occupy their corresponding DMF cell as per their spatial location. As with images, this representation system exploits the advantages that matrices provide, such as simple processing (based on mathematical equations) and direct access. Moreover, the design enables the storage of zero, one or more DMRs in the same DMF cell, thereby ensuring use of 100 % of the data obtained by the sensors, thus preventing its duplication and preservation. By using only the actual information collected by the instruments, the DMF format enables disk space consumption to be minimised and optimisation of information storage and distribution.

The DMF format has been designed with the aim of enabling easy and simple integration with different remote sensing image processing techniques. Storing the data obtained by the sensors in a matrix structure with cells that can obtain a variable number of measurements enables easy adaptation to more complex methodologies by simply adding new fields to the DMF record. By doing so, it is possible to insert data from various sources and moments into the same DMF matrix, thereby enhancing the efficiency of methods based on data fusion and change detection.

Moreover, as mentioned above, as the processing system acquires great importance when generating results from the data obtained by the instruments located on the platform, a detailed study has been carried out to find the best way of integrating the computation models with the proposed data structure and format. To this end, it is proposed to use FPGA devices as main or auxiliary processing systems and the NASA-developed SpaceCube modular system as a reference model, which is composed mainly of processing modules consisting of two interconnected Xilinx Virtex-5 FPGA cards. The proposed hardware module will use the same type of card and, in all cases, will provide the results in DMF format. Firstly, a global diagram for synchronisation and data transfer between the processing system and other devices was designed. Subsequently, basic processing techniques at hardware level were implemented to analyse how the DMF format influences execution times when different types of connection are used to interconnect the two FPGA cards. This experiment is very important as it detects the critical points of the system, in which a bad configuration can lead to *buffer* saturation, creating internal and external *bottlenecks* that hinder the processing of data in real time.

In summary, the proposed DMF format not only enables the most typical problems to be solved and ensures the optimisation of techniques based on remote sensing through the use of a matrix structure at software level, but its integration with reconfigurable hardware systems, such as SpaceCube, also enhances its functionality, while optimising the way in which the results are generated.

The remainder of this thesis document is structured as follows: Chapter 2 details the main reasons for undertaking this project and its defined objectives; Chapter 3 reviews previous work carried out in the field of remote sensing, including an introduction to its main challenges, usually arising from the geocorrection process, and paying special attention to space platform types, in particular, the SpaceCube system; Chapter 4 presents the DMF format and its adaptation to different needs and techniques, as well as a general computation system model based on FPGA cards that integrates with this format as a representation policy of the generated results; Chapter 5 presents the *synthetic and*

real datasets and the experiments carried out to analyse the functioning and efficiency of the DMF format, some of which include techniques and problems arising from the processing of conventional images, and others of which solve specific problems derived from needs of a different nature; Chapter 6 includes the experiments carried out and the results obtained when the DMF format is adapted to and integrated with a reconfigurable hardware system with specific methodologies implemented; Chapter 7 provides an overall analysis of the results obtained and extracts the main considerations and conclusions; Chapter 8 presents the lines of research that remain open and will be worked on in the near future; and, finally, the Annexes A present the detailed circuitry of the methodologies implemented at hardware level.

Capítulo 2

Motivaciones y objetivos

Fue en el curso académico 2006/2007 cuando ingresé en el Grupo de Redes Neuronales y Procesamiento de la Señal (GRNPS) para realizar el Proyecto Fin de Carrera (PFC) y obtener el título de Ingeniero en Informática. En el curso académico siguiente (2007/2008) me matriculé del doctorado en este mismo grupo, adjunto al Departamento de Tecnología de Computadores. A finales del año 2008, obtuve una beca de Formación de Personal Investigador (FPI) asociado a un proyecto regional, lo que me garantizó mi estancia en el grupo hasta Noviembre de 2012.

La línea de investigación principal en la que estuve trabajando, desde el inicio del doctorado a finales de 2007, fue el análisis de imágenes hiperespectrales, obtenidas a través de aviones o satélites, para la detección de *targets*. Es decir, mi trabajo consistía en el estudio de algoritmos que permitieran examinar la respuesta espectral de cada uno de los píxeles de la imagen con el objetivo de encontrar ciertos elementos de interés. Esto incluía la generación de mapas espectrales y clasificaciones a partir de espectros referencia o *endmembers*, o bien de mapas de abundancia, que indicaban el porcentaje presente de cada uno de estos espectros referencia para cada píxel de la imagen.

Fue realizando estas actividades cuando nos dimos cuenta que las detecciones de los elementos de interés tenían un mayor margen de error en los bordes de los objetos presentes en la imagen (por ejemplo de casas, o árboles), y que los resultados obtenidos a realizar un análisis subpíxel quedaban bastante lejos de la realidad si se comparaban con la escena referencia o *ground-truth*. En ocasiones, incluso el elemento de interés podía pasar totalmente desapercibido si su tamaño era demasiado pequeño, a pesar de conocer su localización exacta.

Analizando la posición de los elementos que se deseaban detectar mediante el uso de mapas *ground-truth*, e incluso imágenes de la misma escena, adquiridas por los mismos sensores pero durante vuelos a diferentes altitudes, pudimos comprobar que los píxeles sufrían un cierto desplazamiento espacial que estaba asociado con el proceso de geocorrección. En las imágenes originales (L0) adquiridas por los instrumentos, la localización de los elementos de interés sufrían un error espacial mínimo, asociado a las geometrías y configuraciones de los sensores, pero que se incrementaba de manera variable cuando se utilizaban metodologías de corrección geométrica convencionales. Mediante la realización de estudios posteriores, se comprobó que los movimientos a los que estaban sometidas las plataformas, también jugaban un papel a la hora de incrementar este error espacial (aún no siendo el factor determinante), ya que la eficiencia del proceso de geocorrección se decrementaba.

Por tanto, se empezó a idear el diseño de una nueva estructura de datos, sencilla de manejar y también basada en tipos matriciales, que permitiera tanto la correcta localización de los datos recogidos por los sensores, como la eliminación de problemas derivados, como la utilización parcial del número de mediciones originales, el duplicado de la información, o el desplazamiento de datos para rellenar píxeles vacíos. Debido a este comportamiento nada convencional, la estructura de datos, así como su formato a la hora de almacenar las mediciones, recibió el nombre de Diffused Matrix Format (DMF).

En el año 2009 recibimos la visita del investigador Dr. Anthony Gualtieri, procedente del centro NASA Goddard Space Flight Center (GSFC), el cual mostró gran interés en el proyecto que estábamos realizando, pues se encontraba con estos mismos problemas de desplazamiento generados por el proceso de geocorrección a la hora de utilizar algoritmos de segmentación de imágenes (basados en análisis espectrales) de elaboración propia. Debido al auge y reconocimiento que estaba teniendo el desarrollo de un sistema hardware reconfigurable basado en dispositivos Field Programmable Gate Array (FPGA), denominado SpaceCube, en el centro NASA GSFC, el Dr. Gualtieri nos recomendó elaborar el diseño de un nuevo modelo de sistema de computación que permitiera la integración del formato DMF. Dicho sistema consistiría en la simulación de un módulo de procesamiento SpaceCube compuesto por dos tarjetas FPGA Xilinx Virtex-5 interconectadas, trabajando en paralelo para generar un resultado (dependiente de las metodologías implementadas a nivel de hardware) ya georreferenciado.

Por tanto, la línea principal del proyecto de tesis se dividió en dos ramas diferentes, pero ligadas la una a la otra de manera muy estrecha. Por un lado, se requería la implementación de un tipo de estructura que permitiera minimizar los errores espaciales a la hora de georreferenciar cada una de las mediciones adquiridas por los sensores, que garantizara la integridad de la información, y que solventara los problemas derivados del proceso de geocorrección. Por otro lado, se requería el diseño de un nuevo modelo de sistema de computación hardware que se encargara de recoger los datos de los instrumentos presentes en la plataforma, los procesara en tiempo real, y ofreciera el resultado en la estructura anteriormente descrita, para que la distribución y visualización de la información fuera directa y en el mismo instante en el que ésta estuviera disponible.

De esta manera, se establecieron los siguientes objetivos:

- Diseño e implementación de un nuevo tipo de estructura de datos que permitiera eliminar las imprecisiones espaciales generadas por el proceso de geocorrección, siempre considerando los siguientes aspectos:
 - No debe tratarse de una estructura de datos compleja, para poder garantizar su implantación en cualquier tipo de sistema.
 - Debe garantizar un acceso a la información y procesamiento de los datos rápido y eficiente.
 - Debe minimizar el error espacial al introducido por el sistema de posicionamiento.
 - Debe garantizar el uso del 100 % de los datos recogidos por los instrumentos.
 - Debe evitar el duplicado de datos recogidos por los instrumentos.
 - Debe proporcionar una fácil adaptación a los nuevos requerimientos y necesidades de las misiones.
 - Debe posibilitar la inserción de información procedente de diferentes orígenes y momentos de adquisición, potenciando metodologías basadas en fusión de datos y detección de cambios.
 - Diseño e implementación de un formato de fichero que permita el almacenamiento y salvado de los datos presentes en la estructura de datos, considerando los siguientes aspectos:
 - El salvado y cargado de los datos debe ser lo más rápido posible.
 - El tamaño final de fichero debe estar minimizado a los datos procedentes de los instrumentos y el sistema de posicionamiento, reduciendo al máximo el almacenado de información adicional.
 - Debe permitir una fácil distribución a la comunidad científica.
 - Análisis y estudio del comportamiento y rendimiento de la estructura de datos propuesta de acuerdo a los niveles de movimiento de la plataforma.
 - Estudio del rendimiento de la estructura de datos en campañas de campo.
-

- Adaptación de la estructura de datos propuesta a metodologías más complejas como la fusión de datos, o la detección de cambios.
 - Estudio previo y análisis de viabilidad del uso de la estructura de datos propuesta en un sistema de procesamiento a bordo de plataformas aerotransportadas o espaciales.
 - Diseño de un sistema general de procesamiento hardware basado en FPGA, en el que queden detallados los siguientes parámetros:
 - El contexto y localización dentro del avión o satélite.
 - Conexiones y protocolos de comunicaciones utilizados para la recepción de los datos procedentes de los instrumentos y el sistema de posicionamiento.
 - Conexiones y protocolos de comunicaciones utilizados para la distribución de los resultados utilizando la estructura de datos propuesta.
 - Simulación de un módulo de procesamiento SpaceCube mediante la interconexión de dos tarjetas FPGA Virtex-5:
 - Adaptación de metodologías de teledetección a nivel de hardware.
 - Análisis y optimización de las comunicaciones realizadas entre los componentes situados en la FPGA.
 - Estudio del rendimiento de las conexiones externas para orientar las metodologías al procesamiento a bordo en tiempo real, utilizando la estructura de datos propuesta como el tipo de salida del sistema.
-

Capítulo 3

Introducción

A la hora de estudiar la corteza terrestre desde plataformas con sensores es necesario considerar varios aspectos que influyen directamente con la calidad y fiabilidad de los resultados generados por la metodologías. Uno de estos parámetros es el formato y estructura de datos en el que se va a almacenar, representar, visualizar y procesar la información recogida por los instrumentos. La elección de un formato de datos incorrecto o demasiado complejo puede implicar la inserción de imprecisiones y errores en la salida final de los procesos de ejecución correspondientes. Otro parámetro que es necesario considerar es el sistema de computación a bordo de la plataforma, y la integración del formato de datos diseñado como tipo de salida. Esta integración puede ser más o menos sencilla, pues se requiere una completa sincronización con el resto de dispositivos y una frecuencia de salida aceptable por las metodologías implementadas y por el sistema destino.

El objetivo de este capítulo es, en primer lugar, realizar una introducción a la manera en la que se adquieren las mediciones de un área de interés desde diferentes tipos de plataformas, el tipo de sensores que suelen ser utilizados para dicho propósito, las imprecisiones introducidas por los movimientos a los que pueden estar dichas propósitos, los errores espaciales cometidos durante el proceso de geocorrección de los datos, y los formatos de datos convencionales de almacenamiento de la información obtenida. En segundo lugar, presentar el contexto histórico de los sistemas de procesamiento utilizados hasta nuestros días, prestando especial atención a las últimas tendencias basadas en dispositivos reconfigurables a nivel de hardware, como son las tarjetas FPGA, concretamente el sistema de procesamiento SpaceCube.

Siguiendo esta línea, el capítulo está dividido en tres partes principales. En un primer lugar, la Sección 3.1 refleja, a un nivel general, el contexto en el que se sitúa el trabajo realizado. Posteriormente, la Sección 3.2 muestra una visión general del proceso de adquisición y tratamiento de la información recogida por los sensores, abarcando diferentes aspectos como los tipos de plataformas existentes, los efectos de inestabilidad producidos a la hora de georreferenciar los datos, o las políticas y formatos de almacenamiento utilizados de manera convencional. También se detallan las problemáticas presentes en cada uno de estos aspectos, y qué imprecisiones debería minimizar una estructura o formato de datos adecuado. La Sección 3.3 presenta el funcionamiento de los sistemas de procesamiento, dedicando especial atención en los situados a bordo de las plataformas, los cuales requieren una completa adaptación al sistema de adquisición y una rápida respuesta lo más cercana posible al tiempo real. Como caso concreto de estudio, se detallará un tipo de tecnología emergente que está siendo ampliamente reconocida por la comunidad científica como sistema de procesamiento a bordo de plataformas espaciales: los dispositivos Field Programmable Gate Array (FPGA), los cuales ofrecen la posibilidad de implementar las metodologías propias de la teledetección a través de hardware reconfigurable; y un tipo de sistema de procesamiento que apuesta fuertemente por esta tecnología: SpaceCube. Igual que en la sección anterior, es interesante realizar un estudio sobre las limitaciones en estos sistemas de procesamiento para el diseño de una estructura y formato de datos que maximice sus funcionalidades.

3.1. Antecedentes.

El estudio de la Tierra desde plataformas aero-transportadas (situadas en aviones) o espaciales (situadas en satélites) están ayudando considerablemente a la humanidad a comprender la dinámica natural de nuestro planeta [Vin97]. Las nuevas tecnologías son imprescindibles a la hora de realizar dicho análisis [LeB10], permitiendo grandes avances en diferentes subcampos de ciencias basadas en la naturaleza y medio ambiente, como la geología, la meteorología, la mineralogía, la arqueología, la geografía, la oceanografía, etc.

El desarrollo de las tecnologías y dispositivos que permiten el estudio de la corteza terrestre se enmarcan en el contexto de la *teledetección*, el cual se basa principalmente en la utilización de sensores y sistemas de computación para poder localizar y detectar elementos de acuerdo a su emisión o respuesta espectral (radiancia y reflectancia, respectivamente). Dichos sensores pueden ser clasificados en dos grupos, según su modo de actuación [Wan09]. Por un lado se encontrarían los sensores pasivos, los cuales estudian la radiación natural emitida o reflejada por los elementos situados en una determinada escena, considerando como fuente de radiación el Sol. Por el otro lado se encontrarían los sensores activos, cuyo principio es el mismo que para los sensores pasivos, pero siendo éstos los que generan la fuente de radiación para posteriormente estudiar la respuesta espectral de los elementos.

Un análisis de la corteza terrestre mediante sensores situados en plataformas a gran altitud ofrece un amplio abanico de posibilidades y aplicaciones, todos orientados a la comprensión de fenómenos naturales producidos en la superficie de la Tierra, en su atmósfera, o en sus océanos y mares; y con el objetivo principal de determinar planes antes, durante y después de las manifestaciones y apariciones de dichos fenómenos. Por tanto, la aplicación directa de la teledetección abarca multitud de funcionalidades [Cam02]:

- *Tecnología radar* [Hen98, Dav00]: Es un tipo de tecnología que se basa en el envío de pulsos de radio o microondas para analizar su alteración al incidir en un objeto de interés y ser reflejado de nuevo al origen. Tiene como principal utilidad el análisis y predicción de fenómenos atmosféricos y meteorológicos a gran escala, así como la posibilidad de elaborar modelos de elevación digital (Digital Elevation Models, DEM).
- *Tecnología basada en láser y altímetros radar* [Tsa82, Kra95]: Mediante el análisis de las perturbaciones producidas por la gravedad sobre la superficie de mares y océanos (longitud y altura de las olas) es posible determinar la velocidad y dirección del viento, así como la dirección de corrientes oceánicas superficiales.
- *Tecnología Light Detection and Ranging (LIDAR)* [Bec92, Lef02]: Es un tipo de tecnología basado en sensores activos, los cuales utilizan un láser como fuente de radiación sobre un determinado objeto de interés. Mediante el análisis de la respuesta espectral producida, es posible determinar su distancia, entre otros parámetros. Además, juega un gran papel en el estudio de la actividad vegetal.
- *Tecnología basada en radiómetros y fotómetros* [Str09, Ale02]: Es una de las más utilizadas, y permite el análisis del espectro de emisión y reflexión de un objeto de interés, generalmente en visible e infrarrojos, aunque también en microondas, rayos gamma y ultravioleta. Son de gran utilidad a la hora de determinar los compuestos químicos de los elementos analizados, así como de analizar las concentraciones de los mismos en la atmósfera terrestre.
- *Tecnología basada en fotografías aéreas y estereoscópicas* [Bai98, Bai99]: Este tipo de tecnología hace uso imágenes capturadas con cámaras convencionales o estereoscópicas (desde diferentes puntos de vista, propiciando el cálculo de distancias y potenciando la reconstrucción tridimensional) desde plataformas a gran altitud, por lo que se suele emplear a la hora de generar mapas topográficos.

- *Tecnología basada en imágenes multispectrales o hiperespectrales* [Lan03, Doz04]: Permite la generación de imágenes de la superficie terrestre, en donde las escenas quedan representadas por un número de filas y columnas de píxeles determinado (dependiendo de las especificaciones del sensor), almacenando este píxel la respuesta espectral de una determinada localización en un rango determinado del espectro electromagnético.

Los datos proporcionados por los sensores en el campo de la teledetección pueden adquirir una gran complejidad computacional [Lan02], debido a que suelen ser almacenados en estructuras de datos avanzadas con una gran dimensionalidad, lo que supone un alto coste en los tiempos de procesamiento y de almacenamiento de la información. Este coste puede ser incluso superior si se consideran técnicas basadas en el análisis temporal de los datos, como por ejemplo la detección de cambios, ya que se hace necesaria la georreferenciación, correlación espacial y estudio instrumental de diferentes datasets. Por ello, la tecnología está en constante desarrollo para poder optimizar la manera en la que tal cantidad de información es analizada y procesada, intentando reducir los costes computacionales requeridos durante el almacenamiento y el procesamiento.

Durante esta optimización, se hace necesario el control de dos parámetros principales cuya eficiencia influye directamente en el rendimiento global del sistema de procesamiento y análisis de los datos:

1. *El formato y estructura de datos* [PCI04]: La información recogida por los sensores necesita una distribución correcta y ordenada tanto a la hora de ser procesada (optimizando las estructuras de datos en memoria principal), como a la hora de ser almacenada (optimizando los formatos de datos en disco). Un mal diseño de las estructuras y los formatos correspondientes pueden llegar a suponer una desorganización de dicha información, implicando un bajo rendimiento del sistema de procesamiento general y la inclusión de imprecisiones y errores adicionales.
2. *El diseño del sistema de adquisición y cómputo* [Kom05]: El sistema de adquisición, formado principalmente por los sensores junto con dispositivos adicionales de georreferenciación de los datos, y el sistema de cómputo, encargado de recoger los datos obtenidos por el sistema de adquisición y convertirlos en información entendible por los expertos, necesitan un diseño acorde a las necesidades de las misiones. Deben ser máquinas optimizadas tanto a nivel hardware como software, capaces de recoger la gran cantidad de datos proporcionados por los sensores, procesarla en un tiempo considerable, y almacenarla eficientemente. Especial atención requieren los sistemas de cómputo a bordo de las plataformas, los cuales deben generar resultados en tiempo real a medida que los sensores proporcionan los datos.

Un correcto diseño e implementación de ambos parámetros es trivial para conseguir una generación de resultados y un procesamiento de los datos rápido y eficaz, proporcionando a su vez, una ordenada organización de la información, garantizando su calidad y acceso por parte de la comunidad científica.

3.2. Visión general de los sistemas de adquisición de imágenes.

Los sensores de observación de la Tierra producen una gran cantidad de información diariamente, la cual es almacenada usando diversos formatos de imágenes y requiriendo grandes cantidades de espacio en disco. Una parte de esta información es utilizada en el mismo día de adquisición con el objetivo de analizarla para conseguir un determinado fin. Sin embargo, existe otra gran parte de la misma que no será inmediatamente utilizada o procesada, sino almacenada para usos futuros, como aquéllas concernientes a estudios de detección de cambios [Bar03].

Este masivo espacio requerido para almacenar las imágenes obtenidas por estos sensores hacen necesarios sistemas físicos potentes, pero sobre todo el diseño y desarrollo de nuevos formatos estándar

que permitan la minimización y buen aprovechamiento del mismo. Una posibilidad sería incluir, si fuera necesario, fases de pre-procesamiento orientadas a la compresión de la información, pero que no supongan un impedimento o una reducción considerable en la precisión a la hora de recuperar las imágenes con el objetivo de realizar comparaciones temporales entre ellas o con una determinada imagen verdad-terreno (*ground-truth*), o bien en los tiempos de procesamiento.

Como se ha comentado con anterioridad, una de las aplicaciones principales que hacen amplio uso de las enormes cantidades de información referente a la observación de la Tierra es el estudio de cambios en la superficie de la misma [Lun99]. Sin embargo, uno de los mayores problemas de este tipo de estudio es la dificultad existente a la hora de comparar secuencias de imágenes temporales de una determinada región [Bru00], debido al hecho de que las condiciones de adquisición nunca llegan a ser exactamente las mismas, existiendo variaciones en las condiciones meteorológicas, diferente calibración de instrumentos, o modificaciones en la trayectoria y altura de la plataforma que transporta al sensor.

Los datos digitales de teledetección son almacenados como sets de ficheros que contienen la descripción de los datos (meta-datos), valores de píxel e información espacial [Hol11]. Los formatos de datos actuales almacenan toda esta información usando estructuras matriciales que no permiten técnicas avanzadas como la combinación de imágenes obtenidas por diferentes sensores, la comparación de imágenes para diferentes fechas de adquisición, vuelos, y/u órbitas, ni el procesamiento en tiempo real [Du03].

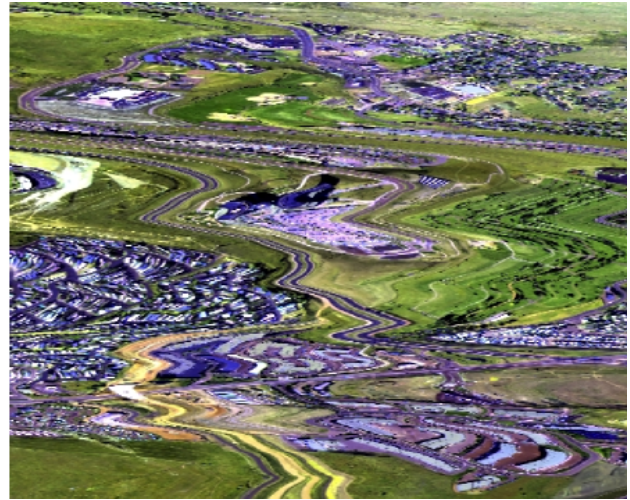
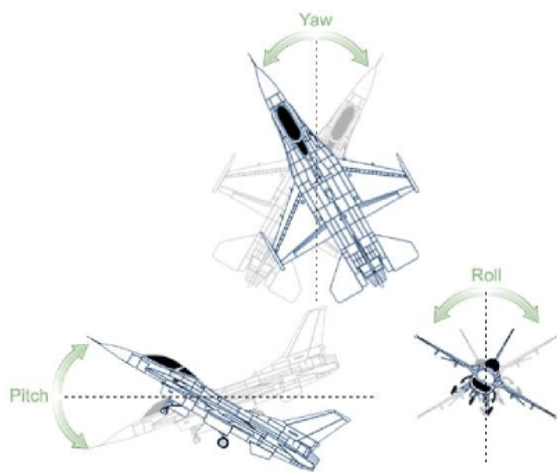
Los datos adquiridos por los instrumentos pueden encontrarse en diferentes niveles de producto que pueden ser definidos por los proveedores, pero que normalmente se basan de acuerdo a la propuesta del Committee on Earth Observation Satellites (CEOS) [CEOwww, EU12, Jus02]:

- *Nivel 0 (L0, RAW)*: Datos sin procesar reconstruidos procedentes de los instrumentos y *payload*, con una resolución completa y junto a información adicional para un correcto procesamiento posterior (por ejemplo efemérides, ventanas de sincronización, cabeceras y notificaciones de comunicaciones, eliminación de datos duplicados, etc.).
- *Nivel 1A (L1A)*: Datos sin procesar reconstruidos procedentes de los instrumentos, referenciados temporalmente e incluyendo coeficientes de calibración radiométrica y geométrica, así como parámetros de georreferenciación (por ejemplo, efemérides de la plataforma) ya procesados, pero no aplicados al nivel L0.
- *Nivel 1B (L1B)*: Unidades adquiridas por los sensores a los que se les ha aplicado los datos correspondientes al nivel L1A (no todos los instrumentos presentan un nivel L1B).
- *Nivel 2 (L2)*: Parámetros geofísicos derivados (por ejemplo, temperatura de la superficie del mar, índice de vegetación, etc.) con la misma resolución y localización que los datos a nivel L1.
- *Nivel 3 (L3)*: Datos o parámetros geofísicos posteriormente calculados que han sido espacial o temporalmente modificados (por ejemplo, derivados de productos de nivel L1 o nivel L2), normalmente completados con información adicional procedente de la aplicación de metodologías concretas, y que las confiere más consistencia. Dichas modificaciones podrían implicar promediados y composiciones.
- *Nivel 4 (L4)*: Salida de modelos o resultados de análisis a productos de datos de menor nivel (por ejemplo, parámetros que no han sido directamente medidos por los instrumentos, pero que derivan de dichas mediciones).

En líneas generales, los datos recogidos directamente por el sensor siempre se encuentran en un formato de imagen bruto, o de Nivel 0 (L0) [Man10]. El pre-procesamiento de imágenes incluye varias operaciones complementarias como la reducción de ruido, la conversión de radiancia y la corrección geométrica, que conforman la cadena de procesamiento [Sha02] dando lugar a diferentes niveles

diferentes, de tal manera que niveles más altos implican una mayor manipulación de los datos (L1, L2, L3...). La corrección de las fuentes de error en las imágenes L0 [Pum03] es una de las operaciones más importantes en las primeras etapas de la cadena de procesamiento, ya que es determinante para una óptima generación de resultados de las metodologías que serán ejecutadas en fases posteriores. El proceso de corrección es necesario, por tanto, para la generación de una proyección ajustada a la realidad de la imagen L0, ya que al haber sido adquirida desde un sensor en la plataforma de un avión o satélite está sometida a una serie de imprecisiones generadas por diferentes fuentes:

1. *Distorsiones originadas por la plataforma* [Ric13a, Ban95] (Figura 3.1(a)): Estas imprecisiones se producen cuando la plataforma en la que está situado el sensor que adquiere los datos está sometida a movimientos anormales e imprevistos, por ejemplo, turbulencias producidas por diferentes presiones en la atmósfera. Las plataformas a bordo de satélites minimizan este tipo de distorsión al considerar el Espacio un entorno *vacío*, por lo que éste está más presente en plataformas a bordo de aviones. En ellas, la referenciación de los datos que componen la imagen L0 se ve alterada al estar el sensor apuntando a una localización diferente a la establecida para cada una de las mediciones. En este contexto, se puede considerar la plataforma en un espacio tridimensional en el que son posibles tres tipos de perturbaciones principales:
 - *Cabeceo (Pitch)*: Cambios de ángulo en el eje lateral del avión, producidos por cabeceos del avión y la modificación de su altitud, y por tanto, generando alteraciones en el tamaño y distancias entre las mediciones obtenidas.
 - *Alabeo (Roll)*: Cambios de ángulo en el eje longitudinal del avión, produciendo alteraciones en el orden, distancia y separación entre las mediciones adquiridas.
 - *Giro lateral (Yaw)*: Cambios de ángulo en el eje vertical del avión que tienen como origen, de manera general, un cambio en el sentido del vuelo, produciendo alteraciones en la geometría de la escena escaneada por el sensor.
2. *Distorsiones originadas por la rotación terrestre* [Ric13b] (Figura 3.1(b)): Estas imprecisiones se generan en plataformas a mucha altitud de la corteza terrestre, en las que queda patente el movimiento de rotación de la Tierra. Es un tipo de alteración típico de las plataformas a bordo de satélites que influye directamente en la geometría de la escena escaneada, caracterizándolas con una morfología romboidal.
3. *Distorsiones provocadas por la atmósfera terrestre* [Gao09, For92] (Figura 3.1(c)): Alteraciones producidas por la presencia de diferentes tipos de aerosoles en la atmósfera terrestre. Se tratan de imprecisiones difíciles de corregir, ya que la señal procedente de los elementos presentes en la corteza terrestre se ve altamente modificada. Es un tipo de distorsión típico de plataformas a bordo de aviones a gran altitud, y de satélites.
4. *Distorsiones provocadas por el ángulo de la fuente de luz* [Xia08, Dox13] (Figura 3.1(a)): Este tipo de distorsión se produce cuando la fuente luminosa (el Sol) refleja la superficie terrestre con el mismo ángulo de incidencia con el que el sensor la está escaneando, produciendo la saturación de los colectores y la aparición de una franja luminosa en la imagen resultante (*sun glint*). Igual que con las perturbaciones introducidas por la atmósfera terrestre, este tipo de distorsión es bastante difícil de corregir.
5. *Distorsiones originadas por el sensor* [LeM12, Tei97] (Figura 3.1(e)): Imprecisiones generadas por las características y geometría propia del sensor que obtiene las mediciones. No sólo puede producir alteraciones en la manera en la que se obtienen los datos (a un nivel espacial) sino que también puede producirlos en la información en sí (a un nivel radiométrico), como por ejemplo la adición de ruido con una magnitud variable de acuerdo al canal de frecuencia, o posibles fallos en determinados colectores de los que conforman el sistema de adquisición.



(a)

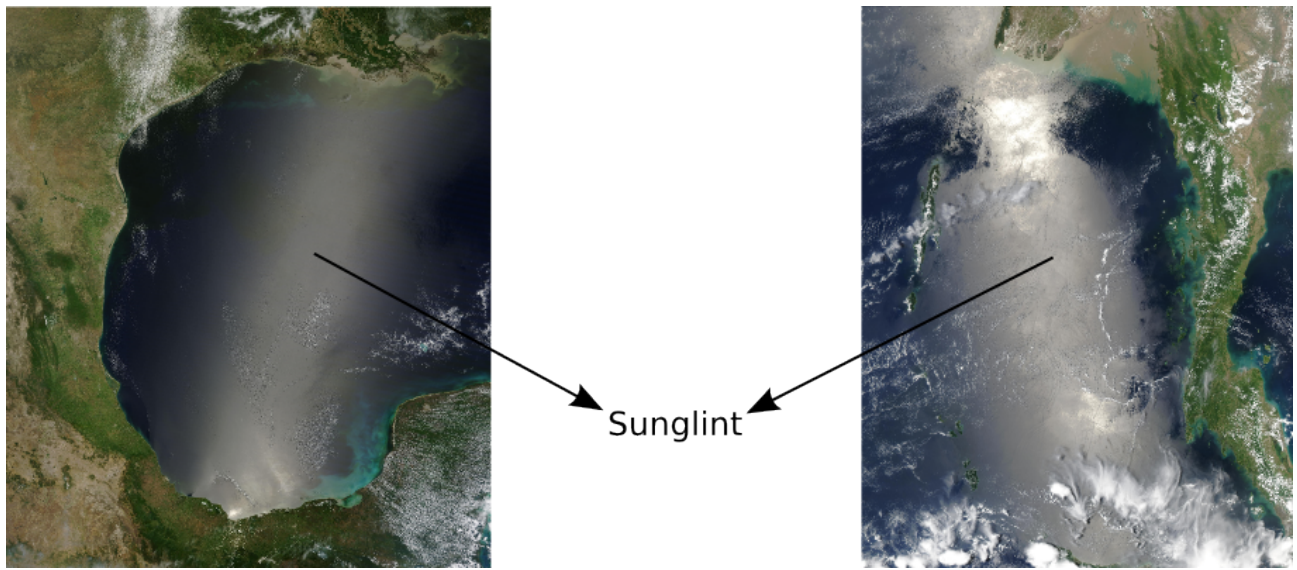


(b)

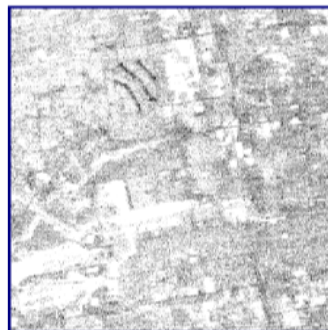
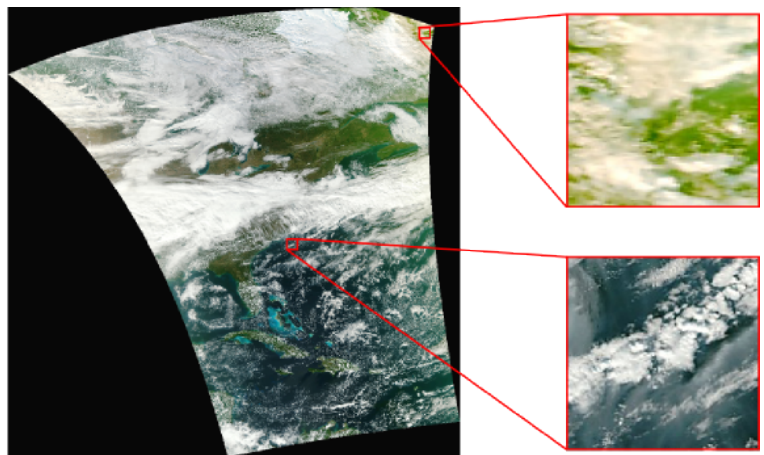
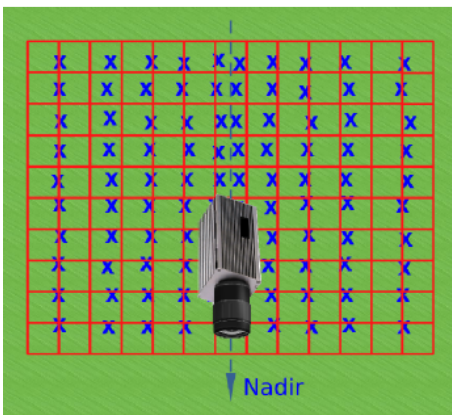


(c)

Figura 3.1: Principales fuentes de imprecisiones presentes en las imágenes L0.
 (a) Generadas por el movimiento de la plataforma. (b) Originadas por la atmósfera terrestre. (c) Provocadas por la atmósfera terrestre. (d) Provocadas por el ángulo respecto a la fuente de luz (sunlint). (e) Originadas por el sensor (Arriba: Efecto observable en el muestreo causado por la geometría del sensor, Abajo: Deficiencias introducidas por el incorrecto funcionamiento de los correspondientes colectores).



(d)



(e)

Figura 3.1: (Cont.)

Las dos últimas fuentes de error generan alteraciones en la información obtenida a un nivel de valor instrumental, con lo que se hacen necesarias metodologías complejas de corrección que suelen ser aplicadas generando diferentes niveles de producto superiores. Sin embargo, las tres primeras afectan a la localización de los datos, es decir, alteran el posicionamiento original de la información. Aunque esto no implica la modificación de los valores instrumentales de los datos, el correcto posicionamiento de las mediciones adquiridas respecto a su localización original es determinante para una óptima generación de resultados por parte de las metodologías que serán ejecutadas en fases posteriores. El procedimiento encargado de georreferenciar cada medición se denomina *proceso de geocorrección* [McW05, Sap12] y suele ser ejecutada en fases iniciales de la cadena de procesamiento [Len03, Cra00]. El proceso de geocorrección es necesario, por tanto, para la generación de una proyección ajustada a la realidad.

Para garantizar la correcta identificación del posicionamiento de cada medición adquirida, las plataformas de adquisición de datos suelen transportar, junto con los sensores correspondientes, dispositivos externos, como por ejemplo sistemas Ground Positioning System (GPS) y sistemas Internal Navigation System (INS) [Mos01, Sch93], los cuales en su conjunto son conocidos como *sistemas de posicionamiento*, y que permiten la localización de la plataforma en tiempo real. Se hace necesaria la sincronización de estos dispositivos con el sistema de adquisición (sensores), para una correcta geolocalización de los datos, debido principalmente a los siguientes motivos [Ska04]:

- El sistema de posicionamiento indica las coordenadas geográficas de la plataforma, pero no para cada medición. Es imprescindible un completo conocimiento del sistema de adquisición de datos y la realización de un módulo de generación de coordenadas para calcular la posición en Tierra a la que estaba apuntando el sensor en ese momento. Al tratarse de operaciones de proyección sobre un plano, se requerirán parámetros adicionales como la altura de la plataforma, velocidad, etc, que pueden ser proporcionadas por el sistema de procesamiento central de la nave, o configurados de manera estática antes de realizar la adquisición de datos.
- Debe existir una determinada sincronización entre el sistema de posicionamiento y el sistema de adquisición de datos, de tal manera que cuando este último obtenga una medición, le envíe una señal a aquél para la recepción del posicionamiento actual de la plataforma, y determinar de esta manera la localización espacial del dato recién adquirido.

Una correcta sincronización entre el sistema de adquisición y el sistema de posicionamiento permitirá, de esta manera, establecer una relación directa entre cada medición adquirida por los sensores y su localización real en la escena escaneada.

A continuación se detalla la manera en la que se realiza la adquisición de los datos de una determinada región de interés a partir de sensores hiperespectrales a bordo de plataformas (Sección 3.2.1), así como las características, ventajas e inconvenientes ofrecidos de acuerdo al tipo de plataforma: terrestre (Sección 3.2.2), aerotransportada (Sección 3.2.3) o espacial (Sección 3.2.4). También se detallarán las imprecisiones espaciales producidas durante la georreferenciación de la información a la hora de considerar estructuras de datos matriciales convencionales (Sección 3.2.5), junto con los formatos y políticas de almacenamiento de los datos pre- y post-procesados convencionales (Sección 3.2.6).

3.2.1. Adquisición de datos desde plataformas y sensores.

La ciencia que permite el estudio y adquisición de imágenes de la corteza terrestre desde plataformas a determinadas altitudes posee un contexto histórico breve, pero con un transcurso rico acompañado del desarrollo de instrumentos y metodologías que permitían, cada vez más, optimizar el análisis de ciertas áreas o regiones de interés [Dou06]. Empezando por el uso temprano y rudimentario de cámaras en globos y palomas durante el siglo XIX, y terminando con el desarrollo de dispositivos de adquisición de datos a bordo de aviones y satélites, tal y como es conocido hoy en día, estableciendo el

término de *teledetección*. Un impulso tecnológico marcado por grandes hitos [Den12] como la utilización de la fotografía desde globos militares durante la Guerra de la Triple Alianza (1864), la posibilidad de poder conservar el negativo de la imagen sin necesidad de tener que ser revelada al momento (1871), o las primeras fotografías realizadas desde un avión sobre Centocelli, Italia (1909).

Hoy en día, la ciencia de la teledetección define el análisis y la interpretación de las mediciones de la radiación electromagnética que es reflejada o emitida por una fuente, observada y/o grabada desde un determinado punto del espacio por un observador o instrumento, la cual no está en contacto con ella [Jos05]. La observación terrestre (Earth Observation, EO) aplicado al campo de la teledetección representa la interpretación y comprensión de mediciones adquiridas por instrumentos a bordo de aviones (*airborne*) o satélites (*spaceborne*) [Net01] capaces de capturar la radiación electromagnética que es reflejada o emitida por los objetos situados sobre la superficie terrestre (nieve, roca, océano, etc.) o bien dentro de la propia atmósfera. Un principio importante a considerar cuando se aplican técnicas propias de la teledetección orientada a EO, es que los diferentes objetos, situados sobre la superficie de la Tierra o bien en su atmósfera, reflejan, absorben, transmiten o emiten energía electromagnética en diferentes regiones espectrales y con diferente magnitud, permitiendo la identificación de los elementos de los que están compuestos [Goe85]. Los sensores a bordo de plataformas, tanto en aviones como en satélites, son capaces de capturar la magnitud del flujo energético emitida o reflejada por las componentes que conforman la superficie terrestre mediante el análisis de la magnitud y distribución de la energía a lo largo de las diferentes frecuencias del *espectro electromagnético*.

El espectro electromagnético (Figura 3.2) [NAS10] es la distribución energética del conjunto de ondas electromagnéticas en todas las posibles frecuencias. Para el caso de un objeto particular, el espectro electromagnético hace referencia a la distribución energética característica que es emitida (*espectro de emisión*) o absorbida (*espectro de absorción*) por el objeto [Phi12]. Para su estudio, el espectro electromagnético se divide en diferentes bandas de acuerdo a su frecuencia (f), longitud de onda (λ), o energía de fotón (E), tal y como se indica en la Tabla 3.1.

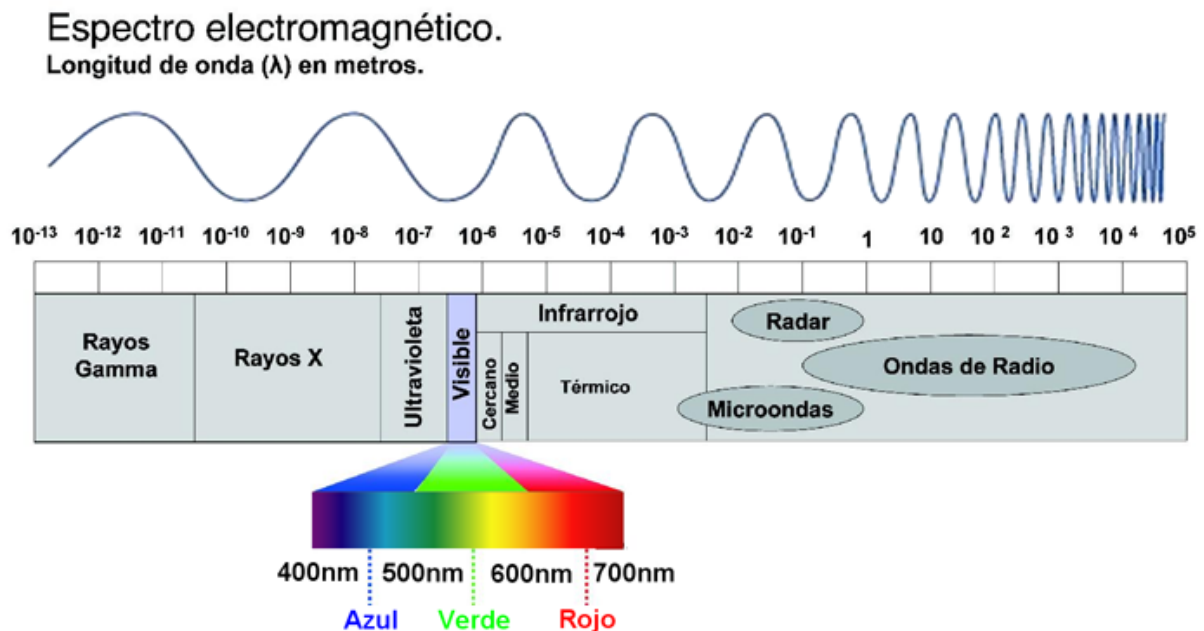


Figura 3.2: Representación del espectro electromagnético.

Por tanto, un determinado objeto puede emitir, absorber, o reflejar energía a lo largo del espectro electromagnético con más o menos intensidad en diferentes rangos. De esta manera sería posible encontrar un objeto que emite mucha radiación (energía) en el rango de los rayos X, pero muy poca en infrarrojos.

Dentro del espectro electromagnético destaca el rango del visible (desde los 380 nm hasta los 780

Banda	λ	f	E
Rayos gamma	< 10 pm	> 30 EHz	> 20×10^{-15} J
Rayos X	< 10 nm	> 30 PHz	> 20×10^{-18} J
Ultravioleta extremo	< 200 nm	> 1.5 PHz	> 993×10^{-21} J
Ultravioleta cercano	< 380 nm	> 789 THz	> 523×10^{-21} J
Luz visible	< 780 nm	> 384 THz	> 255×10^{-21} J
Infrarrojo cercano	< 2.5 μ m	> 120 THz	> 79×10^{-21} J
Infrarrojo medio	< 50 μ m	> 6 THz	> 4×10^{-21} J
Infrarrojo lejano / submilimétrico	< 1 mm	> 300 GHz	> 200×10^{-24} J
Microondas	< 30 cm	> 1 GHz	> 2×10^{-24} J
Radio frecuencia ultra alta	< 1 m	> 300 MHz	> 19.8×10^{-26} J
Radio frecuencia muy alta	< 10 m	> 30 MHz	> 19.8×10^{-28} J
Radio onda corta	< 180 m	> 1.7 MHz	> 11.22×10^{-28} J
Radio onda media	< 650 m	> 650 kHz	> 42.9×10^{-29} J
Radio onda larga	< 10 km	> 30 kHz	> 19.8×10^{-30} J
Radio frecuencia muy baja	> 10 km	< 30 kHz	< 19.8×10^{-30} J

Tabla 3.1: Bandas del espectro electromagnético.

nm), que es el intervalo de energía que el ser humano puede percibir con el ojo, el cual actúa como sensor natural. Todos los objetos que emitan o reflejen energía en dicho intervalo pueden ser percibidos con la vista, diferenciando los diferentes niveles de frecuencias como colores. Los niveles de energía con longitud de onda menor a 380 nm y mayor a 780 nm es imposible percibirla con el ojo humano, y es necesario utilizar sensores.

Uno de los aspectos más importantes en el campo de la teledetección es la *reflectancia* [Mob99]. La reflectancia es el porcentaje de radiación reflejado por un objeto al incidir sobre él una determinada fuente de luz. La energía de la luz reflejada se ve modificada respecto a la energía de la luz incidente en todo el espectro electromagnético, dependiendo del tipo del material del que está compuesto el objeto. Esta respuesta es única para cada tipo de material, y su magnitud puede ser representada a lo largo del espectro electromagnético (Figura 3.3), lo que se conoce como *firma espectral*. De esta forma, es posible determinar la composición de un objeto en concreto mediante el análisis de su firma espectral [Cla89].

Los sistemas de adquisición de datos utilizados en teledetección permiten obtener una imagen de la superficie terrestre en diferentes longitudes de onda o frecuencias, utilizando como fuente de luz natural el Sol. La manera en la que se realiza dicha adquisición puede variar de acuerdo a la tecnología del instrumento, pero de manera general está compuesto por un sistema óptico y un prisma que permita dispersar en diferentes longitudes de onda la luz solar que llega a nuestro dispositivo tras ser reflejada en la escena. Mediante un detector formado por n sensores (cada uno de ellos encargado de capturar la imagen en una determinada frecuencia) es posible adquirir una medición en n longitudes de onda diferentes [Spa07]. La magnitud adquirida por los sensores en cada longitud de onda se denomina Nivel Digital (ND).

Un sensor electro-óptico puede ser modelado por el conjunto de procesos mostrado en la Figura 3.4 [Sch07]. Las operaciones de escaneo convierten la radiancia espacial adquirida a una señal óptica continua y variante en el tiempo. Los detectores, en cambio, convierten esa señal óptica en una señal electrónica variante en el tiempo, la cual es amplificada y procesada por la electrónica del sensor. En el conversor analógico/digital (A/D), la señal procesada es sampleada en tiempo y cuantizada en valores ND representando los píxeles espaciales de la imagen.

La geometría de la adquisición depende principalmente de la configuración del sensor, pudiendo ser posible de esta manera diferenciar diferentes tipos de sistema de captura [Por03], tal y como se muestra en la Figura 3.5. En dicha figura se han representado en color rojo las mediciones que están siendo capturadas en un mismo instante t , y en color amarillo las mediciones ya capturadas. El resto de la escena aparece en color verde. De acuerdo a la geometría de adquisición, podemos encontrar,

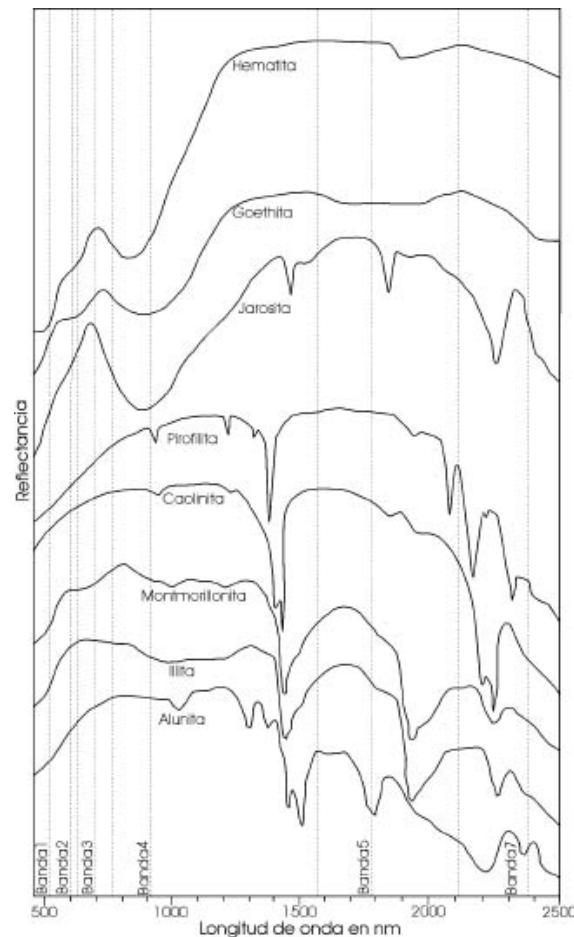


Figura 3.3: Firmas espectrales de diferentes minerales en el rango del visible e infrarrojo.

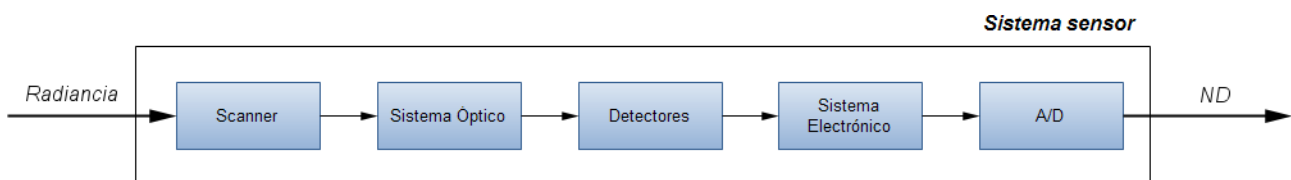


Figura 3.4: Componentes primarios de un sistema de teledetección electro-óptico.

por tanto, los siguientes tipos de sistema de adquisición:

- Sensores *whiskbroom* [Bre00]: Son de tipo electromecánico. Tienen un único detector, el cual escanea la escena mediante la rotación de un prisma o espejo que mueve el *Campo de Visión Instantáneo* (Instantaneous Field Of View, IFOV) hacia el objetivo deseado (Figura 3.5(a)). De esta manera, una única medición de la escena es tomada en cada instante t . La geometría resultante es cilíndrica. El sistema de coordenadas usado por este sensor es hemisférico y el tamaño del área de integración del píxel o Campo de Visión Instantáneo en Tierra (Ground Instantaneous Field Of View, GIFOV) del sensor depende de la distancia de la plataforma al objetivo en tierra.
- Sensores *pushbroom* [Gup97]: Poseen una serie de detectores lineales, que son colocados transversalmente a los movimientos de la plataforma (Figura 3.5(b)). De esta manera, una línea de n mediciones (para n detectores) de la escena es tomada en cada instante t . El sistema de coordenadas para este tipo de sensor es cilíndrico y el tamaño del GIFOV depende de la distancia del eje cilíndrico al objetivo en tierra.
- Sensores matriciales (*frame cameras*) [Kau12]: Poseen una matriz de detectores (Figura 3.5(c)).

Todos los elementos sensoriales están dispuestos en el plano focal y presentan la particularidad de que adquieren la imagen digital en un solo instante. De esta manera, una matriz de $n \times m$ mediciones (para $n \times m$ detectores) de la escena es tomada en cada instante t . El sistema de coordenadas para este tipo de sensor es cartesiano y el tamaño del GIFOV depende de las coordenadas i y j del objetivo.

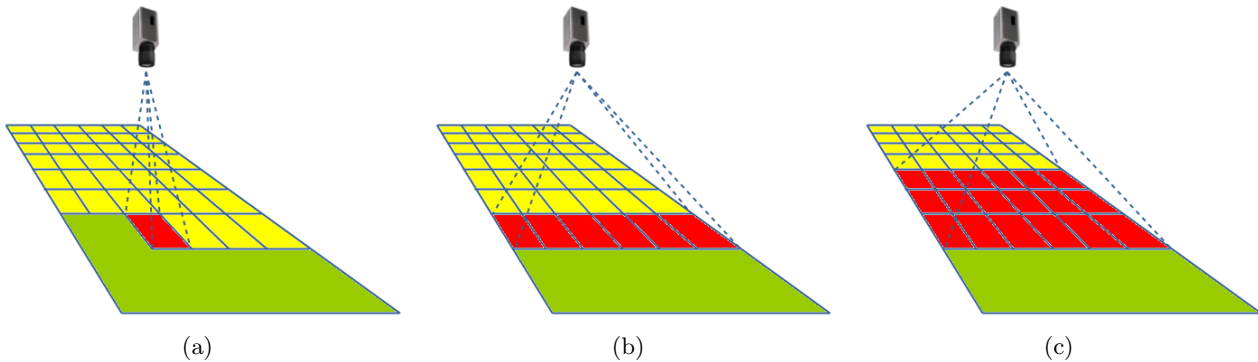


Figura 3.5: Geometría de los sensores digitales.
 (a) Sensor whiskbroom. (b) Sensor pushbroom. (c) Sensor matricial.

El resultado del escaneo de la superficie terrestre por el sistema de adquisición es una sucesión de mediciones que conforman un cubo de datos denominado *imagen hiperespectral* (Figura 3.6) definiendo la imagen L0 [Mur05]. Los ejes x e y hacen referencia a la dimensión espacial, que consistirá en la sucesión de mediciones en el orden en el que fue adquirida por el detector (en una imagen L0), o bien las mediciones georreferenciadas acorde a su latitud y longitud (en una imagen geocorregida). El eje z hace referencia a los espectros de las mediciones. Este cubo de datos cobra gran importancia en el campo de la teledetección. Como habíamos comentado con anterioridad, un tipo de material responde a la luz incidente mediante un determinado espectro. Si se conoce la respuesta espectral de un determinado componente (por ejemplo, hierro, acero, algún tipo de piedra, algún tipo de vegetación, etc.), podemos rastrear todas las mediciones adquiridas por el sensor para buscar cuáles presentan un comportamiento espectral como el del componente buscado. Si existen mediciones con el mismo comportamiento espectral, mediante sus localizaciones espaciales podremos determinar en qué regiones de la escena este elemento está presente. Técnicas más avanzadas de teledetección, como por ejemplo el desmezclado espectral [Kes02], permiten estimar el porcentaje de material de cada medición a partir de sus comportamientos espectrales.

La información que representa una única imagen hiperespectral está constituida por una gran cantidad de datos, y su tamaño dependerá directamente del número de mediciones adquiridas y de la resolución espectral. Debido a que la traza de la imagen suele cubrir una distancia considerable, y a que la resolución espectral de las cámaras es cada vez mejor, el resultado es una imagen de bastante tamaño y muy costosa de procesar computacionalmente [Gel04]. Es por esto que el formato de almacenamiento de la misma tanto en disco, como en memoria RAM (para procesamiento), debe presentar un diseño que permita una manipulación óptima, ligera y eficaz de la misma [Cha07a].

3.2.2. Campañas de campo y sensores hiperespectrales en plataformas terrestres.

Las campañas de campo consisten en la planificación de sucesivas salidas al campo por parte de analistas y expertos para la adquisición de información detallada de una zona determinada, la cual posee varios tipos de componentes (piedras, minerales, vegetación, etc.) de interés [Mor04]. El objetivo es utilizar espectrómetros o sensores en plataformas (Figura 3.7) para la obtención de la respuesta espectral de dichos componentes de interés (o bien la recolección de los mismos para posteriores pruebas

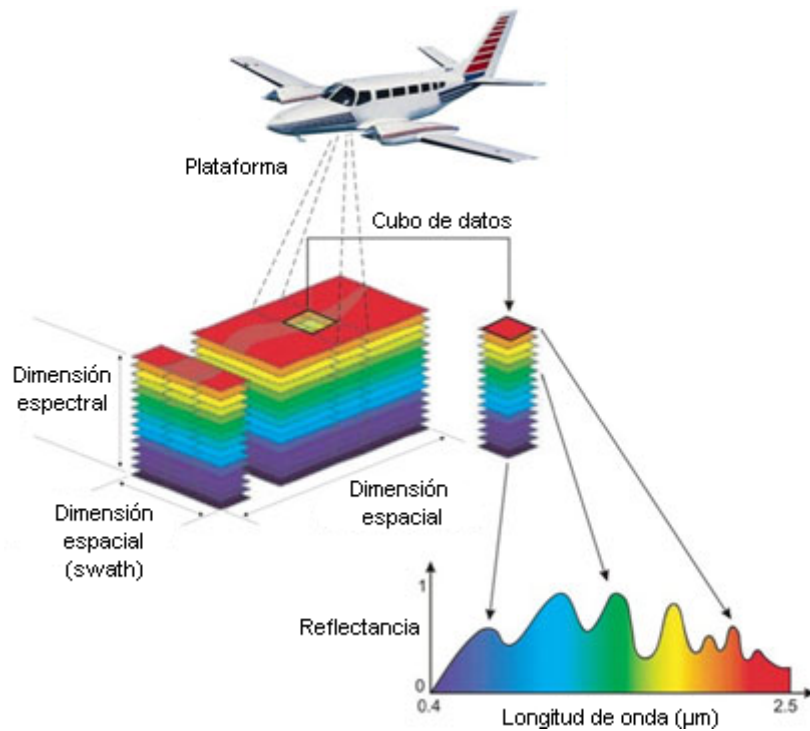
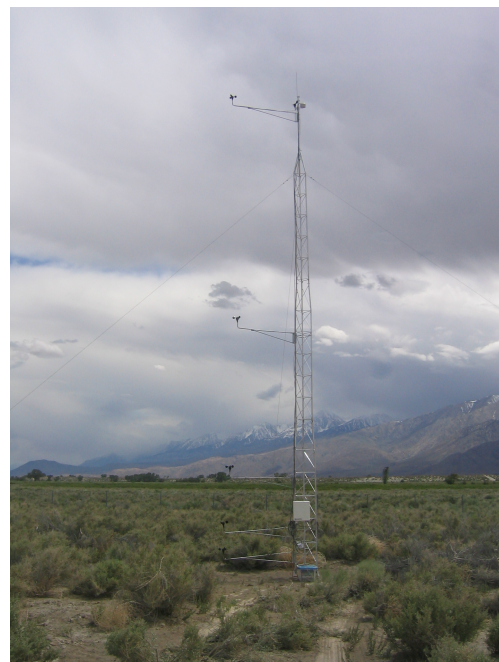


Figura 3.6: Adquisición de mediciones y construcción del cubo de datos hiperespectral.

de laboratorio) y así poder construir una serie de espectros referencia (librería espectral) [Boj03] que puede ser utilizada para optimizar los procesos y metodologías aplicados en fases posteriores sobre imágenes adquiridas desde aviones o satélites. En el caso de utilizar espectrómetros, es el propio analista el que se desplaza por la escena recogiendo espectros de los elementos de interés. En cambio, las plataformas terrestres pueden no requerir la atención constante de los expertos y pueden llegar a tener cierto nivel de autonomía, aunque pueden ser móviles o fijas [Mil80].



(a)



(b)

Figura 3.7: Adquisición de medidas utilizando sensores en Tierra.

(a) Analista analizando la respuesta espectral de diferentes elementos durante una campaña de campo. (b) Plataforma terrestre situada en una torreta móvil.

Las ventajas e inconvenientes que presentan las plataformas terrestres son variables, y dependen directamente del lugar en el que esté instalada la plataforma. Por ejemplo, plataformas situadas en vehículos [Fog03] presentan una gran alternativa, ya que la rapidez en la que se realiza el escaneo de la superficie es regulable, de acuerdo a la velocidad del vehículo, y que pueden obtenerse mediciones hasta a una altura de 15 metros, pero posee un inconveniente importante, y es que pueden existir zonas inaccesibles en las que el vehículo es incapaz de entrar, y su rango de efectividad se limita por tanto a carreteras, caminos y alrededores. Plataformas terrestres instaladas en mástiles portables [Pap01] permiten el estudio de un determinado área desde mayores altitudes, y permiten un cambio de situación fácil mediante pequeños remolques. Sin embargo, son bastante inestables frente a condiciones climáticas adversas. Una solución más estable es utilizar torretas [Kum95], las cuales son mucho más estables, al ser generalmente construidas con materiales más duros y percederos. Pero su limitación es que son fijas y sólo son rentables si se desea realizar un estudio continuo durante un periodo largo de tiempo.

3.2.3. *Sensores hiperespectrales en plataformas aero-transportadas.*

Las plataformas aero-transportadas, tal y como su nombre indica, están situadas a bordo de helicópteros o aviones [Bir94] (Figura 3.8), los cuales sobrevuelan la zona de interés para realizar su escaneo. Este tipo de adquisición se caracteriza principalmente por su alta resolución espacial, y por su grado de dificultad a la hora de georreferenciar los datos, al poder estar el avión sometido a turbulencias de la atmósfera. Se tratan, por tanto, de plataformas que están sometidas a inestabilidades y su consecuente imprecisión a la hora de geolocalizar la información recogida. Para aligerar y solventar en parte este problema, las plataformas aero-transportadas suelen ir acompañadas de giróscopos que permiten detectar los movimientos *pitch*, *roll* y *yaw* a los que está sometida.



Figura 3.8: Previsualización de una plataforma aero-transportada.

A rasgos generales, este tipo de plataformas tienen las siguientes ventajas:

- El avión puede volar a altitudes relativamente bajas pudiendo permitir una resolución espacial por debajo de 1 metro.
- Los aviones pueden fácilmente cambiar su planificación de vuelo para evitar malas condiciones meteorológicas como nubes, los cuales pueden bloquear la vista del suelo de un sensor pasivo.
- Pueden realizarse cambios de última hora para ajustar la iluminación del sol, la localización del área a escanear, y visitas adicionales a esa localización.

- El mantenimiento y reparación de los sensores pueden ser realizadas de manera fácil.
- Las trayectorias de vuelo de un avión no conoce límites, únicamente los límites políticos.

Pero también poseen las siguientes desventajas:

- Obtener un permiso de vuelo para introducirse en un espacio aéreo extranjero puede ser un proceso largo.
- La baja altitud de los vuelos del avión establece un zona de escaneo (*swath*) demasiado estrecha, por lo que para áreas de interés grandes se requieren varias pasadas del avión para cubrir la zona al completo.
- El tiempo transcurrido entre que se adquieren los datos, hasta que son proporcionados a la comunidad científica correspondiente se retrasa al tener que esperar que el avión vuelva al aeropuerto antes de transferir la imagen L0 al proveedor para el preprocesamiento.

Algunos de los instrumentos aero-transportados más populares hoy en día para la captura de imágenes terrestres de la superficie terrestre son los que se muestran a continuación:

- *Airborne Visible/Infrared Imaging Spectrometer (AVIRIS)* [AVIwww, Vane93, Gre98]: Es un instrumento de probada eficacia en el campo de la teledetección. Consta de un único sensor que proporciona imágenes calibradas de radiancia en 224 canales (bandas) espectrales contiguos con unas longitudes de onda desde 400 nm a 2500 nm. Utiliza un sensor tipo *whiskbroom* que permite obtener un total de 614 píxeles por oscilación. La señal medida por cada detector se amplifica y se codifica utilizando 12 bits.
- *79-channel Digital Airborne Imaging Spectrometer (DAIS 7915)* [DAIwww, Cha93, Sob04]: Los 79 canales de alta resolución del DAIS 7915 permiten recoger información de la superficie de la superficie terrestre en un rango de longitudes de onda de 0.4 μm a 12.3 μm en un formato digital basado en palabras de 16 bits. El tipo de mecanismo utilizado es un sistema *Kennedy* donde un espejo poligonal cúbico escanea el terreno bajo el avión a través de una ventanilla abierta en la parte inferior del mismo. Las rotaciones realizadas por dicho espejo permite capturar las mediciones en un carácter horizontal mientras que el desplazamiento del avión permite obtenerlas en un carácter vertical.
- *Reflective Optics System Imaging Spectrometer (ROSIS)* [ROSwww, Kun91, Hua10]: Es un espectrómetro aero-transportado compacto, de tipo *pushbroom*. Presenta dos modos operacionales, el modo imagen y el modo espectral. En el modo espectral se ofrecen 81 canales (bandas) en el rango 440 nm - 852 nm, con una resolución espectral de 12 nm en el rango 440 nm - 572 nm, y de 4 nm en el rango 580 nm - 852 nm. En el modo imagen sólo están disponibles 29 de los 81 canales, pero se ofrece una resolución espacial tres veces mayor que en el modo espectral.
- *Airborne Hyperspectral Scanner (AHS)* [AHSwww, Gom07, Rej04]: Es un espectrómetro lineal aero-transportado tipo *whiskbroom* que ofrece imágenes de la superficie terrestre compuestas por un total de 80 bandas: 63 de ellas en la parte reflectiva del espectro electromagnético, 7 bandas en el rango de 3 μm a 5 μm y 10 bandas en el rango de 8 μm a 13 μm .

La Tabla 3.2 muestra una comparativa de las características generales de los sensores comentados anteriormente.

Tabla 3.2: Resumen de las características generales de los sensores más populares a bordo de plataformas aero-transportadas.

Sensor	Rango Espectral	Bandas	Píxels/Línea	Tipo	Resolución Radiométrica
AVIRIS	0.4 - 2.5 μm	224	512	<i>Whiskbroom</i>	12 bits
DAIS 7915	0.5 - 12.5 μm	79	512	<i>Whiskbroom</i>	15 bits
ROSIS	0.43 - 0.86 μm	115	512	<i>Pushbroom</i>	14 bits
AHS	0.44 - 13.17 μm	80	750	<i>Whiskbroom</i>	12 bits

3.2.4. Sensores hiperspectrales en plataformas espaciales.

Las plataformas espaciales son aquéllas que van instaladas en satélites [Zho13] (Figura 3.9), y que permiten el estudio desde el Espacio. Se caracterizan por la adquisición de datos desde gran altitud, lo que supone una baja resolución espacial pero una mayor cobertura (*swath*), así como la capacidad de visitar zonas ya escaneadas con anterioridad, con una resolución temporal que depende directamente de las características de la órbita del satélite. Esto potencia el desarrollo de metodologías basadas en este parámetro, como la monitorización de elementos en la superficie terrestre o la detección de cambios [Smi99, Pal97].



Figura 3.9: Previsualización de una plataforma espacial.

A rasgos generales, este tipo de plataformas tienen las siguientes ventajas:

- Gran cobertura espacial.
- Frecuente y repetitiva cubierta de un área de interés.
- Medición cuantitativa de elementos sobre la corteza terrestre usando sensores calibrados radiométricamente.
- Procesamiento y análisis computerizado semi-automático.
- Coste menor relativo por unidad de cubierta.

Pero también poseen las siguientes desventajas:

- Por normal general, la resolución espacial no es tan elevada como en las plataformas aerotransportadas, aunque existen excepciones.

- El mantenimiento y reparación de los instrumentos a bordo de la plataforma es difícil y costoso.
- Generación de una gran cantidad de datos que es necesario gestionar para un óptimo almacenamiento y procesamiento.

Algunos de los instrumentos espaciales más populares hoy en día para la captura de imágenes terrestres de la superficie terrestre son los que se muestran a continuación:

- *Moderate-resolution Imaging Spectroradiometer (MODIS)* [MODwww, Xio09, Yan06]: Es un instrumento científico a bordo de los satélites Terra y Aqua. Captura datos en 36 bandas en el rango de frecuencias $0.4 \mu\text{m}$ - $14.4 \mu\text{m}$ a diferentes niveles de resolución espacial (2 bandas a 250 m, 5 bandas a 500 m, y 29 bandas a 1 km). Con un instrumento a bordo de cada satélite se consigue escanear la superficie terrestre al completo en sólo 1 ó 2 días. Fueron diseñados con el propósito de proporcionar mediciones sobre la dinámica global a gran escala, incluyendo cambios en la cubierta de nubes de la Tierra, intercambios de energía y procesos ocurriendo en océanos, tierra y atmósfera baja. Tres calibradores a bordo (un difusor solar combinado con un monitor de estabilidad de difusor solar, un ensamblaje de calibración radiométrica espectral, y un *cuerpo negro*) proporcionan calibración a bordo.
- *Enhanced Thematic Mapper Plus (ETM+)* [ETMwww, Bar99, Ekl01]: Se trata de un radiómetro multispectral tipo *whiskbroom* de 8 bandas a bordo del satélite Landsat-7, capaz de proporcionar imágenes de alta resolución de la superficie terrestre. Detecta radiación filtrada espectralmente en los rangos VNIR, SWIR, LWIR y bandas pancromáticas de la cara iluminada de la Tierra con un *swath* de 183 km a una altitud de 705 km.
- *Quickbird* [Quiwww, Tou02, Yam05]: Es un sistema de adquisición de datos a bordo del satélite de mismo nombre, el cual adquiere imágenes pancromáticas con una resolución de 60 cm, e imágenes multispectrales con una resolución de 2.4 a 2.8 m. A fecha de hoy, se trata del satélite que proporciona las imágenes de mayor tamaño y el que dispone de mayor capacidad de almacenamiento a bordo. La resolución espacial que ofrece, excepcional para una plataforma espacial, permite tal nivel de detalle que es posible identificar perfectamente edificios y otro tipo de estructuras.
- *TerraSAR-X Antena SAR X-banda* [Terwww, Wer04, Mas10]: Sensor activo a bordo del satélite TerraSAR-X trabajando a una longitud de onda de 31 mm (frecuencias de 9.6 GHz). Permite la obtención de imágenes radar de alta calidad del planeta entero con una órbita polar a 514 km de altitud. Dicha órbita polar fue planificada de tal forma que el satélite siempre se encuentra en la cara iluminada por el Sol, asegurando una óptima recolección de energía a través de sus paneles solares. Este instrumento proporciona imágenes radar con una resolución espacial de 1 m.
- *Compact High Resolution Imaging Spectrometer (CHRIS)* [Chrwww, Mol04, Ver08]: Se trata de un instrumento hiperespectral cuyo objetivo es analizar las capacidades de los espectrómetros en plataformas de satélite pequeñas y ágiles. Proporciona imágenes con un número de bandas programable desde 19 a 63 en el rango VNIR (400 - 1050 nm) con una resolución espacial de 18 m para un número de bandas mínimo, y de 36 para un número máximo. Cada imagen proporcionada conforma un área de estudio de 13×13 km.

La Tabla 3.3 muestra una comparativa de las características generales de los sensores comentados anteriormente.

Tabla 3.3: Resumen de las características generales de los sensores más populares a bordo de plataformas espaciales.

Sensor	R. Espectral	Bandas	Swath	Tipo	R. Espacial	R. Temporal
MODIS	0.4 - 14.4 μm	36	2330 km	<i>Whiskbroom</i>	250 - 1000 m	1 - 2 días
ETM+	0.45 - 12.5 μm	8	183 km	<i>Whiskbroom</i>	30 - 120 m	16 días
Quickbird	450 - 900 nm	6	16.5 km	<i>Pushbroom</i>	0.61 - 2.88 m	1 - 3.5 días
TerraSAR-X	31 mm	1	10 - 100 km	Varios	1 - 18 m	3 - 11 días
CHRIS	415 - 1050 nm	19 - 63	14 - 18 km	<i>Pushbroom</i>	18 - 36 m	7 días

3.2.5. Los efectos de la inestabilidad de la plataforma y el proceso de geocorrección.

Mientras un sensor captura una imagen, la plataforma en el que está instalado puede estar sujeto a movimientos externos aleatorios provocando que las posiciones reales de los centroides de las mediciones adquiridas por éste no coincidan con ningún patrón regular de la escena [Phi13]. La Figura 3.10(a) y la Figura 3.10(b) muestran un ejemplo de cómo un sensor de tipo *whiskbroom* toma las mediciones cuando la plataforma está sujeta a este tipo de movimientos. Los círculos representan el área de integración del GIFOV. Cada uno de los GIFOV ha sido numerado para identificar el orden de adquisición. A la hora de almacenar las mediciones adquiridas por el sensor, éstas son colocadas de acuerdo a la posición relativa de una matriz sin considerar su localización real espacial, construyendo de esta manera la imagen L0 (Figura 3.10(c)). Es decir, la primera medición, independientemente de su localización real es asignada en el primer píxel de la matriz L0; la segunda medición en el segundo píxel; la tercera medición en el tercer píxel, y así sucesivamente. Debido a que las mediciones no se adquieren con regularidad espacial debido a los movimientos de la plataforma, el análisis visual de las imágenes L0 siempre da la sensación de que éstas están distorsionadas.

Debido a este fenómeno, la posición espacial de los centroides de las mediciones adquiridas en una imagen no corresponderían a los de otra imagen sobre la misma escena, incluso si ambos han sido capturados por el mismo sensor, sobre la misma plataforma y a la misma altura. Por lo que es prácticamente imposible conseguir dos imágenes L0 exactamente iguales de un mismo área aunque todos los parámetros de vuelo y captura sean idénticos. Por tanto, la construcción de la imagen L0 final (y su calidad) depende directamente de la aleatoriedad de los movimientos de la plataforma durante la adquisición de la misma.

Para un correcto análisis visual de la imagen, y para un óptimo procesamiento de la información recogida por los sensores teniendo en cuenta la localización de las mediciones, se hace necesario un proceso de geocorrección en las primeras fases de la cadena de procesamiento [Kla94]. Este proceso de geocorrección permite la generación de una imagen espacialmente corregida mediante la proyección de las mediciones adquiridas, de acuerdo a las coordenadas proporcionadas por el sistema de posicionamiento (y dispositivos y/o herramientas auxiliares) sobre una estructura de datos de carácter matricial [Liu13, Zit03, Gos05, Sim07, Dom08] (Figura 3.11). De esta manera, la imagen geocorregida se tratará siempre de una matriz tridimensional en el que la dimensión X representará la coordenada en longitud, la dimensión Y la coordenada en latitud, y la dimensión Z el valor instrumental (un espectro para el caso de sensores multi/hiperespectrales). El proceso de geocorrección es un procedimiento en el que entran en juego diferentes parámetros:

- El valor instrumental de las mediciones, o información recogida por el sensor.
- Las coordenadas de las mediciones proporcionadas por el sistema de posicionamiento.
- El tamaño de píxel (σ), el cual puede ser configurable durante el proceso de geocorrección, pero que generalmente es proporcionado por el proveedor al disponer los datos a la comunidad científica, previo cálculo a partir de variables como altitud, velocidad y dirección de la plataforma, o características electrónicas y ópticas del sensor.

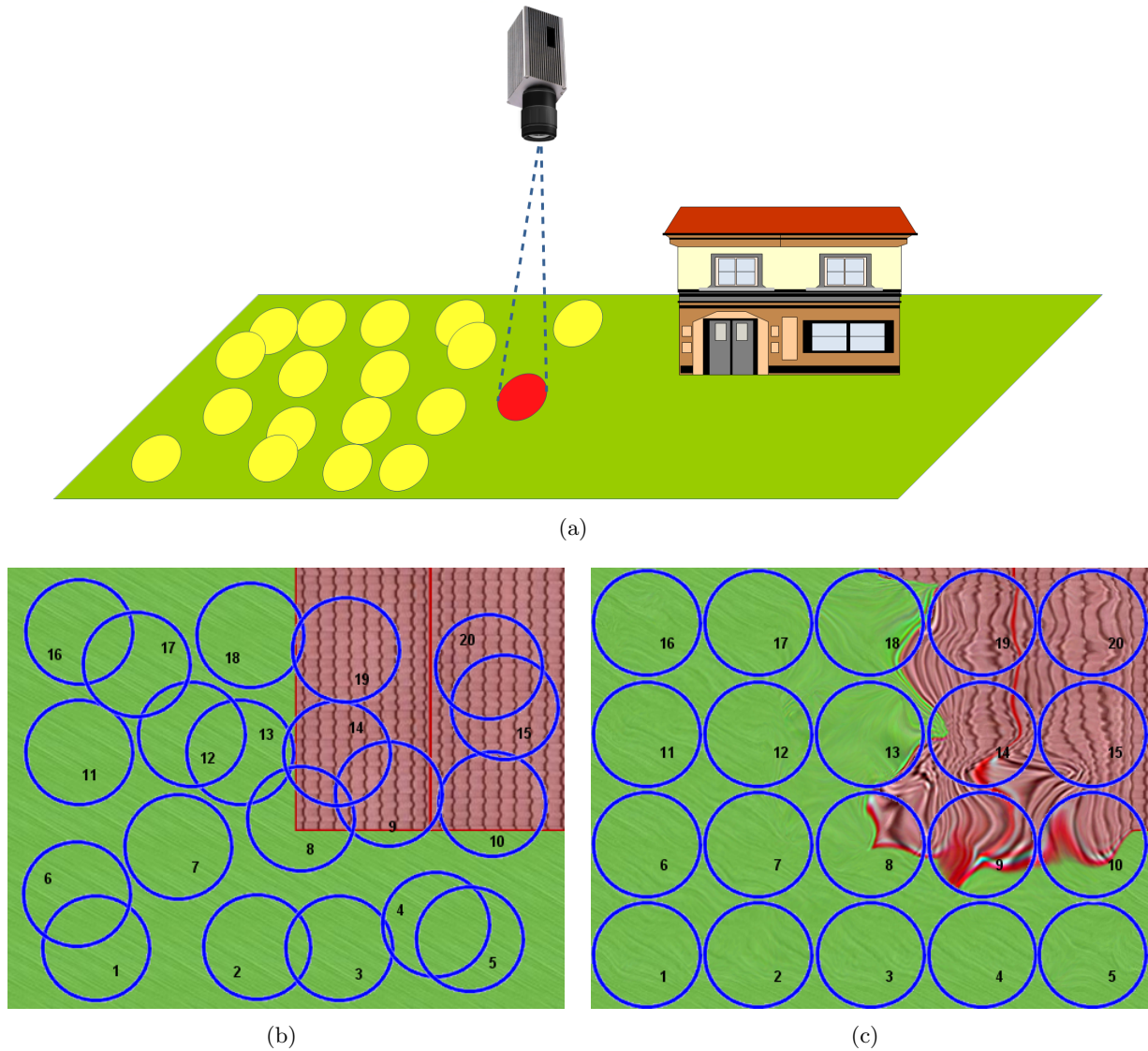


Figura 3.10: Proceso de construcción de una imagen L_0 a partir de la adquisición de mediciones por un sensor de tipo whiskbroom desde la plataforma.

(a) Sensor en plataforma obteniendo mediciones. (b) Localización real de las mediciones adquiridas por el sensor. (c) Construcción de la imagen L_0 .

- Ángulo de la trayectoria de la plataforma (α), que permite la orientación de los datos respecto a la dirección N/S.

La primera tarea a realizar durante el proceso de geocorrección es la generación de la matriz tridimensional que almacenará los datos recogidos por los sensores. Es necesario determinar la localización del área de interés a partir del tamaño de píxel considerado estableciendo el número de columnas (N_c) y filas (N_f) de la matriz geocorregida tal como:

$$\begin{aligned} N_c &= \left\lceil \frac{M_L - m_L}{\sigma} \right\rceil \\ N_f &= \left\lceil \frac{M_l - m_l}{\sigma} \right\rceil \end{aligned} \quad (3.1)$$

Donde M_L , m_L , M_l , m_l , son la longitud máxima, longitud mínima, latitud máxima y latitud

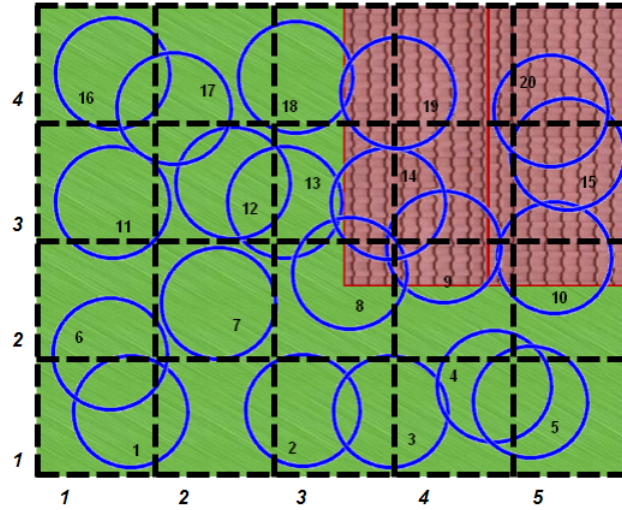


Figura 3.11: Relación entre la localización de las mediciones adquiridas por el sensor y la rejilla matricial que permitirá el correcto posicionamiento de los datos.

mínima proporcionadas por el sistema de posicionamiento, y σ el tamaño de píxel proporcionado por el proveedor.

Tras la creación de la matriz geocorregida, se recorren todas las mediciones presentes en la imagen L0, y para cada una de ellas, y a partir de las coordenadas proporcionadas por el sistema de posicionamiento, se calculan sus posiciones $[x, y]$ correspondientes mediante:

$$\begin{aligned} x &= \left\lfloor \frac{L - m_L}{\sigma} \right\rfloor \\ y &= \left\lfloor \frac{l - m_l}{\sigma} \right\rfloor \end{aligned} \quad (3.2)$$

Donde L y l son las coordenadas de la medición en longitud y latitud, m_L y m_l la longitud y latitud mínimas de la escena de interés, y σ el tamaño de píxel proporcionado por el proveedor.

El resultado es una imagen geocorregida que almacena los datos adquiridos por el sensor y proyectados en dirección N/S. En el caso de que la plataforma no haya seguido dicha dirección, sino que su trayectoria haya estado definida por un ángulo α respecto a la misma, la imagen resultante abarcará espacialmente la escena escaneada y los alrededores, cuyas mediciones no han sido adquiridas. Estos píxeles situados fuera del área escaneada son representados por espectros con valores a 0. Esta información es completamente inservible y puede llegar a incrementar el tamaño en disco de la imagen de manera considerable. Con el objetivo de minimizar el número de píxeles situados fuera de la escena, y por consiguiente, el tamaño total de la imagen, los proveedores de la imagen suelen considerar una geocorrección considerando el ángulo α respecto a la dirección N/S [Boa99]. De esta forma, la Ecuación 3.1 que definía el tamaño de la matriz que contenían los datos queda determinada como:

$$\begin{aligned} N_c &= \left\lceil \frac{(M_L - m_L)\cos(\alpha) - (M_l - m_l)\sin(\alpha)}{\sigma} \right\rceil \\ N_f &= \left\lceil \frac{(M_L - m_L)\sin(\alpha) + (M_l - m_l)\cos(\alpha)}{\sigma} \right\rceil \end{aligned} \quad (3.3)$$

Mientras que la Ecuación 3.2 que definía el posicionamiento $[x, y]$ de cada medición dentro de la estructura matricial queda determinada por:

$$\begin{aligned} x &= \left\lfloor \frac{(L - m_L)\cos(\alpha) - (l - m_l)\sin(\alpha)}{\sigma} \right\rfloor \\ y &= \left\lfloor \frac{(L - m_L)\sin(\alpha) + (l - m_l)\cos(\alpha)}{\sigma} \right\rfloor \end{aligned} \quad (3.4)$$

Todo este proceso queda resumido y representado de manera visual en la Figura 3.12.

Por tanto, el proceso de geocorrección de los datos puede considerarse como una transformación de las coordenadas originales de las mediciones $[L, l]$, proporcionadas por el sistema de posicionamiento, a un tipo específico de coordenada $[x, y]$ que permite localizar cada medición dentro de la estructura matricial geocorregida.

El proceso de geocorrección puede suponer un gran impacto en diferentes metodologías del procesamiento de imágenes, ya que este tipo de política puede generar una serie de imprecisiones que pueden repercutir seriamente en procedimientos y metodologías posteriores en la cadena de procesamiento [Wan09, Ale07], como por ejemplo el análisis sub-píxel [Sma00]. Dichas imprecisiones, que aparecen durante y tras el proceso de geocorrección, proceden principalmente de asignar información distribuida espacialmente de manera aleatoria (y que no sigue un patrón regular) a un tipo de estructura que considera una distancia entre mediciones fija y constante (σ). Éstas se detallan a continuación:

- *Imprecisión 1)*: A dos o más mediciones les corresponden el mismo píxel. En este caso la solución es asignar el promedio de las mediciones correspondientes, o bien seleccionar aquélla más cercana al centro del píxel. Para la primera opción, se estaría mezclando información de datos que, aunque cercanos, pueden ser diferentes, mientras que con la segunda opción, se estarían descartando mediciones que podrían no ser asignadas a ningún otro píxel, perdiendo información. La Figura 3.13 muestra un ejemplo de un píxel $(3,3)$ que posee dos mediciones (*medición 13* y *medición 14*) dentro de su área. Uno de los dos, o bien la media de los valores instrumentales de ambas mediciones (según la política escogida), representará el centro del píxel.
- *Imprecisión 2)*: Un píxel puede quedar sin asignar (vacío), debido a que no se ha adquirido ninguna medición dentro del subárea que abarca. En este caso la solución es considerar alguna de las diferentes políticas asignación por vecindad más comunes [Bab10] (Figura 3.14):
 1. *Vecino más próximo (Nearest Neighbor)*: Al píxel se le asigna directamente la medición más próxima.
 2. *Interpolación lineal (Bilinear interpolation)*: Al píxel se le asigna el promedio de los valores instrumentales de las 4 mediciones más próximas. Dicho promedio puede ser ponderado, considerando un mayor peso a mediciones más cercanas al centro del píxel.
 3. *Convolución cúbica (Cubic convolution)*: Al píxel se le asigna el promedio de los valores instrumentales de las 16 mediciones más próximas. Dicho promedio puede ser ponderado, considerando un mayor peso a mediciones más cercanas al centro del píxel.

Es fácil darse cuenta de que el proceso de asignación de mediciones cercanas a píxeles no asignados supone la inserción de información falsa en la imagen geocorregida, ya que se está considerando un determinado valor instrumental a las áreas comprendidas por el píxel que no han sido medidas, y en las que no es posible determinar sus contenidos a través de los datos originales. Para el caso de la política *Vecino más próximo*, además se produce un duplicado de la información, ya que la misma medición se está asignado a dos píxeles diferentes. La imprecisión introducida por las políticas *Interpolación lineal* y *Convolución cúbica* puede llegar a ser muy elevada si las mediciones están muy separadas espacialmente entre sí, además de generar un valor instrumental ficticio (resultado de los promedios) no obtenido por el sistema de adquisición. Por este motivo,

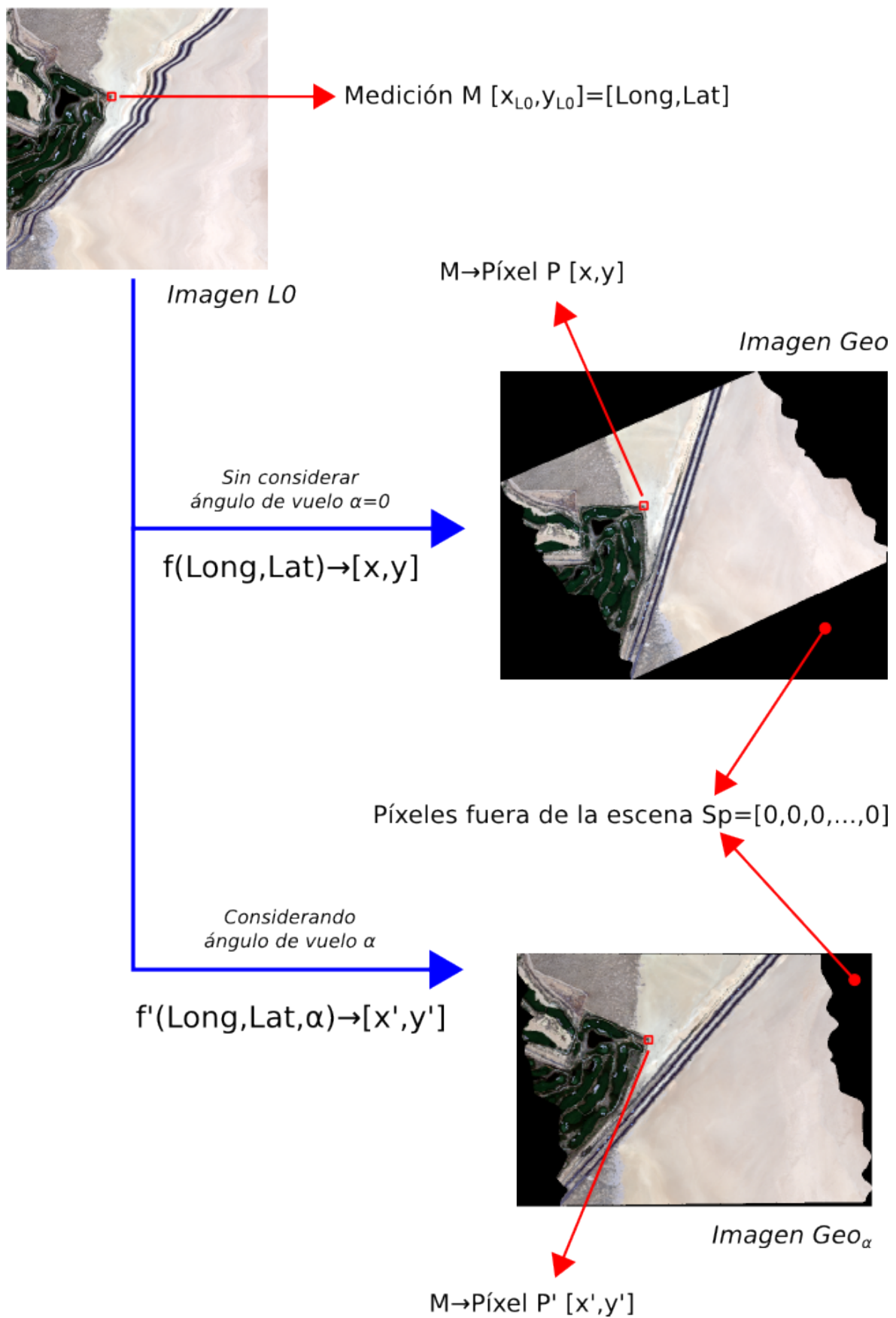


Figura 3.12: Proceso de geocorrección considerando y sin considerar el ángulo de vuelo.

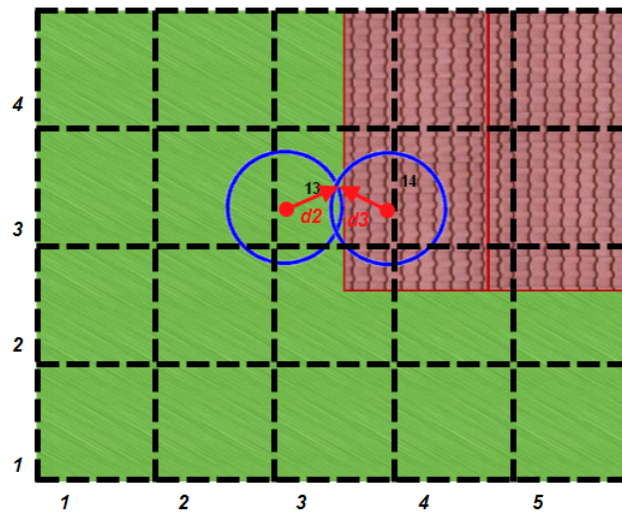


Figura 3.13: *Píxel con dos mediciones dentro de su área.*

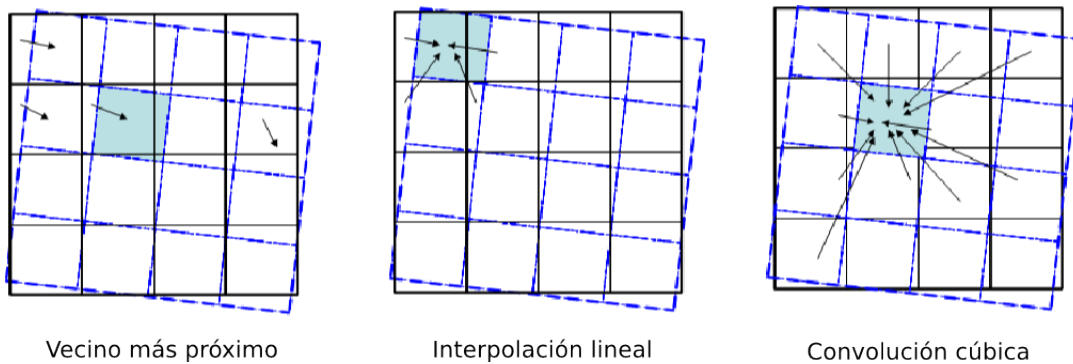


Figura 3.14: *Políticas de selección de mediciones próximas cuando un píxel queda sin asignar.*

siempre se considerará una política de *Vecino más próximo* para el resto de la documentación, al ser considerada como la que menos imprecisiones produce durante el proceso de geocorrección. La Figura 3.15 muestra un ejemplo de píxel vacío (1,2) al que se le asigna otra medición (medición 2), correspondiente a otro área de píxel diferente (1,3), siguiendo la política de *Vecino más próximo*.

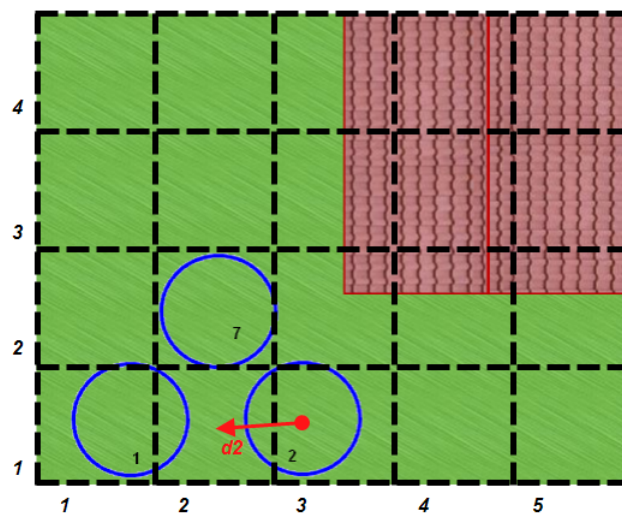


Figura 3.15: *Asignación de medición por vecino más próximo a un píxel vacío.*

- *Imprecisión 3*): Los píxeles de las imágenes geocorregidas suelen ser georreferenciados mediante las coordenadas geográficas del píxel superior izquierda de la imagen, así como del tamaño de

píxel. De esta manera, la localización de los centros del resto de píxeles que conforman la imagen geocorregida $[L, l]$ viene determinada por:

$$\begin{aligned} L &= L_{ref} + \sigma * x \\ l &= l_{ref} + \sigma * y \end{aligned} \quad (3.5)$$

Siendo L_{ref} y l_{ref} las coordenadas de referencia (píxel superior izquierda) en longitud y latitud, σ el tamaño de píxel proporcionado por el proveedor, y x e y las posiciones del píxel en la matriz (columna y fila).

El inconveniente de esta política de georreferenciación es que siempre se estará cometiendo el arrastre de un error debido a la limitación de la precisión del formato numérico a la hora de definir la localización del píxel referencia y el tamaño del píxel, tal y como se muestra en la Figura 3.16.

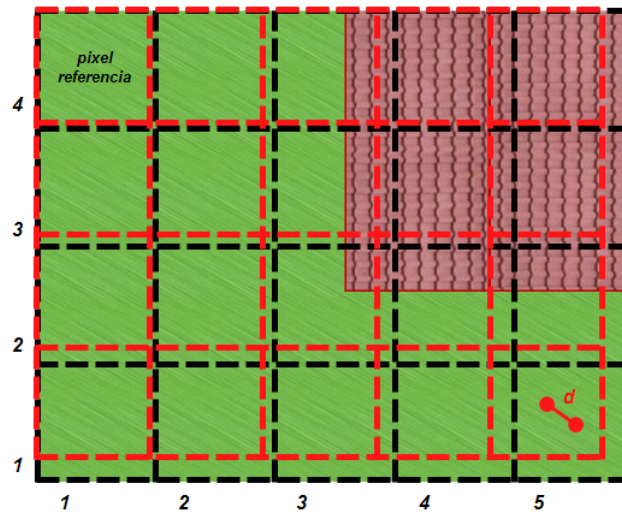


Figura 3.16: Error de localización arrastrado al considerar el píxel superior izquierda de la imagen geocorregida como referencia.

- *Imprecisión 4*): Las mediciones sufren un desplazamiento a la hora de ser asignados al centro de un píxel. Los píxeles de la imagen geocorregida tienen una distribución regular, mientras que las localizaciones de las mediciones adquiridas por el sensor no. El hecho de tener que georreferenciar cada medición por el centro del píxel en el que reside en vez de su localización original provoca un desplazamiento d cuya magnitud dependerá del tamaño del píxel considerado. De esta manera, siempre se estará aplicando con un determinado error que hace imposible que algoritmos como el desmezclado espectral trabajen con precisión.

Como puede observarse en la Figura 3.17, la *medición 13* fue obtenida sobre un área cubierta únicamente por hierba. Esto significa que la composición espectral que puede ser derivada de esta medición por desmezclado espectral debería ser 100 % hierba y 0 % de otros componentes. Sin embargo, la composición espectral en la celda de la imagen geocorregida $(3,3)$, donde la *medición 13* ha sido asignada tras la geocorrección, muestra un 55 % de hierba y un 45 % de tejado y además, no corresponde al espectro asignado de 100 % de hierba. Esta discrepancia es debido al hecho de que la *medición 13* haya sido desplazada una distancia d_{13} de su posición real, y con él la información espectral haya sido desplazada también.

Obviamente, este problema de re-localización no afecta únicamente a un sólo píxel, sino a la mayoría de píxeles de la imagen geocorregida. Por ejemplo, la *medición 18* tiene una composición de 90 % de hierba y un 10 % de tejado, y tras la geocorrección la posición de píxel $(4,3)$ tiene una composición de 55 % y 45 %; o la *medición 9*, cuya superficie está compuesta por un 10 % de hierba y un 90 % de tejado que sustituirá la verdadera composición de 65 % de hierba y 35 % de tejado en el píxel $(2,4)$.

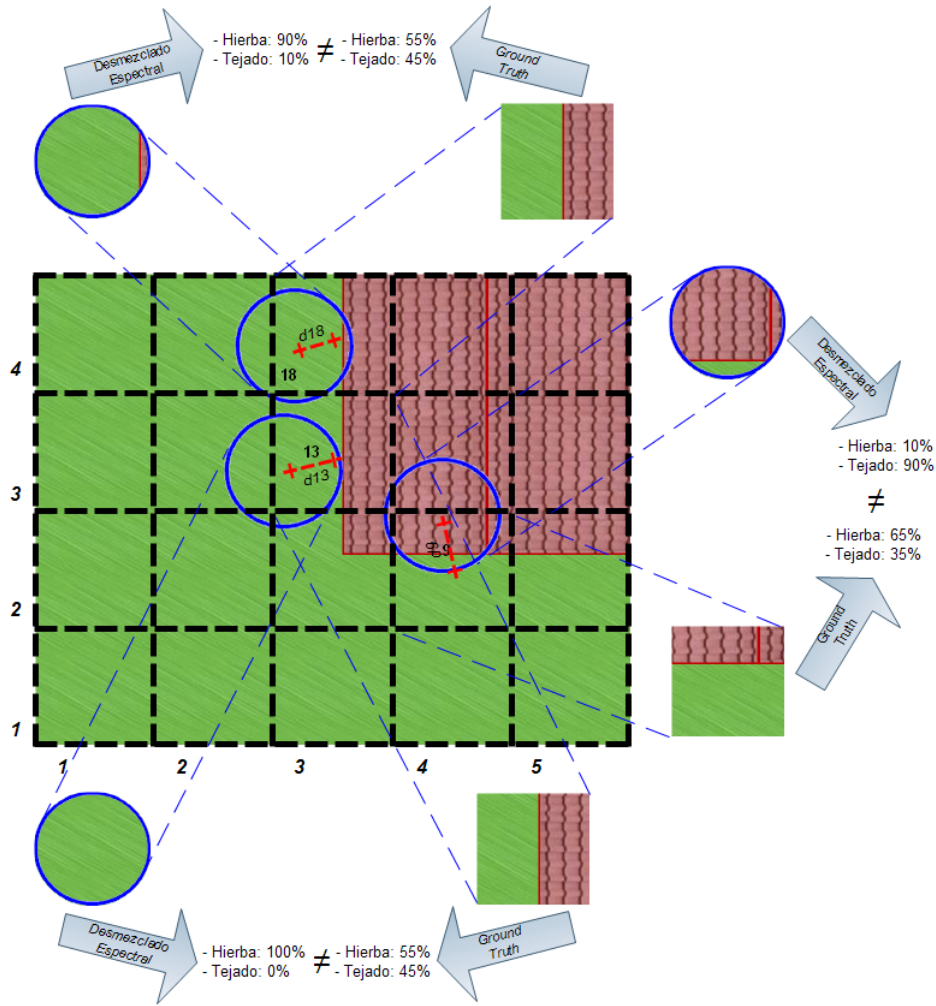


Figura 3.17: El problema del desplazamiento de las mediciones, respecto a su localización real, en el proceso de geocorrección.

Por supuesto este problema se agrava cuando se consideran políticas de asignación de mediciones a píxeles no asignados, como las detalladas en la *Imprecisión 2*.

Desde el punto de vista del usuario las imágenes geocorregidas son más fáciles de entender y comparar con otras imágenes. Las imágenes L0, por el contrario, pueden ser difíciles de analizar y de interpretar por expertos humanos [Tou04]. En este contexto, el proceso de geocorrección puede ser considerado como una técnica opcional que, entre otras funciones avanzadas (como una mejor registración de imágenes en metodologías basadas en detección de cambios o fusión de datos) hace de las imágenes un tipo de información más fácilmente legible y manipulable por el analista.

Desde el punto de vista de la máquina, el proceso de análisis y e interpretación de la imagen adquirida por el sensor puede ser llevado a cabo de dos maneras diferentes:

- *Geocorregir la imagen primero y después aplicar el algoritmo de procesamiento:* Ésta es la manera en la que se suele llevar siempre a cabo el proceso de análisis, pues los proveedores suelen entregar las imágenes ya georreferenciadas a la comunidad de expertos. En este caso, existirían todos los problemas de imprecisión detallados con anterioridad, y se estaría siempre considerando un determinado porcentaje de error correspondiente a la inserción de información falsa y redundante, a la no utilización del set de datos original completo, y a un error de precisión espacial, los cuales son introducidos durante el proceso de geocorrección. En caso de escoger esta política, es fácil pensar que una manera sencilla de solucionar los problemas que se han detallado con

anterioridad sería utilizar un tamaño de píxel de imagen geocorregida mínimo (por ejemplo la menor distancia entre mediciones), de tal manera que a cada píxel se le asigne una única medida, y el desplazamiento que sufran las mediciones sea mínimo. Pero entonces surgirían nuevos problemas, como por ejemplo que el número de píxeles sin asignar se incrementaría notablemente, siendo asignados mediciones por vecino más próximo, e introduciendo más información falsa a la imagen; que el error de localización cometido por la limitación del formato numérico al estar únicamente el primer píxel (superior izquierdo) georreferenciado sería mayor; y que el tamaño de la imagen final podría ser demasiado grande debido al incremento de la resolución, implicando un mayor cómputo y una peor manipulación.

- *Aplicar el algoritmo de procesamiento primero y después aplicar la geocorrección:* Es posible para una máquina procesar la imagen L0 usando técnicas convencionales de análisis digital de imágenes, y dejando el proceso de geocorrección para el final, cuando se requiere intervención humana para dirigir la comparación de imágenes y el análisis visual. La ventaja de esta política es que se puede llevar a cabo un procesamiento de imágenes con más precisión, ya que se está utilizando la información espectral y espacial real tal y como fue adquirida por el sensor al completo. Es justamente en este esquema cuando surge la necesidad de un formato de almacenamiento para imágenes hiperespectrales que permita este procesamiento preservando la precisión, pero que permita al usuario intervenir de manera cómoda mediante una representación de la escena fiel a la realidad.

Los problemas de imprecisión detallados con anterioridad, derivados del proceso de geocorrección, así como la gran cantidad de espacio de almacenamiento que exigen las imágenes temporales, sobre todo en campos científicos como el de detección de cambios, requieren un nuevo diseño de formato de almacenamiento que permita preservar la precisión espacial adquirida por el sensor así como una minimización de la información tanto en disco como en memoria RAM (durante el procesamiento), y que permita a los expertos un análisis visual mediante una proyección fiel a la realidad [Kle07].

Ya existen algunas soluciones básicas que intentan reducir dichos problemas. Una de ellas es el uso de una máscara con dimensiones análogas a las de la imagen hiperespectral geocorregida [LeM11], donde cada píxel indica si a su correspondiente en la imagen geocorregida le corresponde la medición asignada por localización espacial, o bien la medición ha sido asignada de otro píxel cercano por vecino más próximo. De esta manera, y usando dicha máscara, podemos únicamente procesar la información que consideremos de mayor validez. Aunque esta solución resuelve la *Imprecisión 2*, se seguirían cometiendo los errores de desplazamiento al considerar el centro del píxel como la localización espacial de la medición asignada (*Problema 4*), así como la no utilización de todo el set de datos original obtenido por el sensor. Un ejemplo de dicha máscara es el fichero GLT que en ocasiones acompaña al fichero que contiene la imagen hiperespectral geocorregida, el cual se detalla en la Sección 3.2.6.1.

3.2.6. Formatos de almacenamiento de imágenes hiperespectrales.

Como se ha comentado anteriormente, los datos digitales de teledetección adquiridos por los sensores suelen ser almacenados como un set de varios ficheros. Algunos de ellos contienen la descripción de los datos (metadatos) [Ras11], otros contienen los valores de las mediciones, y otros contienen todo lo referente a la información espacial. Aún así, la forma en el que los datos residen en los ficheros puede variar, según el diseño del proveedor, por lo que hoy en día es posible encontrar un gran número de formatos [Boa10]. El comité CEOS lleva años dedicando un gran esfuerzo en el desarrollo de formatos estándar con resultados bastante prometedores. A pesar de encontrar multitud de alternativas a la hora de almacenar la información, uno de los formatos que se están imponiendo actualmente es el HDF, diseñado por la NASA, debido principalmente a su sencillez y eficiencia.

A pesar de este esfuerzo realizado por instituciones y usuarios en la creación y estandarización de formatos de almacenamiento de datos digitales de teledetección, los diseños de los nuevos formatos

de imágenes deberían no únicamente limitarse a tratar de minimizar el espacio de almacenamiento de los mismos, sino también en potenciar otros aspectos como la preservación de la precisión de la información espacial ofrecida por el sistema GPS/INS de la plataforma, el análisis temporal de imágenes, el almacenamiento de imágenes adquiridas por diferentes sistemas de adquisición (diferentes sensores, en diferentes plataformas), o el procesamiento en tiempo real [Bur11, Fol10]. Es posible que estos nuevos formatos necesiten el desarrollo de nuevos algoritmos de procesamiento de imágenes o la modificación de los ya existentes, para procesar y analizar la información contenida en los mismos [Cha13, Bla12].

A continuación se presentan los tipos de ficheros que suelen utilizarse a la hora de distribuir la información recogida por los sensores a bordo de la plataforma (Sección 3.2.6.1); las diferentes políticas utilizadas para organizar la información instrumental a lo largo de los ficheros de datos (Sección 3.2.6.2); y los formatos y contenedores de datos estándar más utilizados de manera general (Sección 3.2.6.2).

3.2.6.1. Formatos de datos.

Actualmente uno de los problemas existentes al trabajar con imágenes de teledetección es la ausencia de un formato estándar para el almacenamiento de la información capturada por el sensor. Los proveedores ofrecen sus productos en los formatos nativos diseñados por la compañía, agencia o institución que adquirió los datos. Aún así, generalmente una imagen hiperespectral es almacenada usando cuatro tipos de ficheros diferentes, aunque este esquema puede variar ligeramente dependiendo del proveedor de los datos [Vin99]:

- *Fichero cabecera* (Figura 3.18(a)): Contiene descripciones numéricas y textuales (meta-datos) sobre la manera en la cual está almacenada la información en los tres ficheros restantes (de codificación binaria), así como información adicional como los detalles de la compañía o agencia proveedora, fecha y hora del vuelo, tipo de cámara empleada, altura del avión durante la adquisición de datos, etc. Suele tener una codificación de texto plano, facilitando la información referente a las condiciones bajo las que fue realizada la obtención de mediciones de manera sencilla y rápida.
- *Fichero IGM* (Figura 3.18(b)): Fichero con las mismas dimensiones espaciales que el fichero de datos L0, donde cada píxel contiene la información correspondiente a las coordenadas espaciales de los correspondientes píxeles del fichero de datos L0. Para ello, consta de dos bandas, donde la primera hace referencia a las coordenadas en longitud, mientras que la segunda hace referencia a las coordenadas en latitud.
- *Fichero GLT* (Figura 3.18(c)): Ofrece información sobre cómo ha sido construida la imagen geocorregida. Posee las mismas dimensiones espaciales que ésta, y cada píxel almacena la posición (i,j) del píxel asignado de la imagen L0. Es decir, si el píxel $(3,4)$ de la imagen GLT contiene un valor $(2,3)$, quiere decir que al píxel $(3,4)$ de la imagen geocorregida se le ha asignado la medición correspondiente al píxel $(2,3)$ de la imagen L0. Como el fichero IGM, el fichero GLT consta de dos bandas, la primera referenciando las posiciones de píxel en x y la segunda referenciando las posiciones de píxel en y . Si el valor de la posición de píxel es positivo, significa que es una medición a la que realmente le corresponde ese píxel en la imagen geocorregida, según sus coordenadas. Si el valor es negativo, significa que a ninguna medición le correspondía ese píxel en la imagen geocorregida, siendo asignada otra medición correspondiente a otro píxel por vecino más próximo. Valores a $(0,0)$ indican que no hay medición actual para ese píxel (fuera de la escena de interés).
- *Fichero de datos*: Almacena los valores de todas las bandas para cada píxel. El método usado para almacenar esta información depende del diseño desarrollado por el proveedor, pero comúnmente estos diseños tienen como base tres políticas: el *Band Sequential* (BSQ), el *Band*

Interleave by Line (BIL) y el *Band Interleave by Pixel* (BIP), los cuales se detallan en la Sección 3.2.6.2. El fichero de datos puede almacenar tanto la información correspondiente a la imagen L0 (Figura 3.18(d)), como a la geocorregida (Figura 3.18(e)).

3.2.6.2. Políticas en el almacenamiento del fichero de datos.

Independientemente del soporte físico de la imagen, es necesario conocer el formato de grabación de los datos, de cara a poder reconstruir posteriormente la información detectada por el sensor [Chu07]. En este sentido, conviene considerar dos aspectos:

- *La codificación aplicada*: Los ND de la imagen se graban en código binario. Es decir, cada valor almacenado (que corresponde a un valor de una medición en una determinada longitud de onda) se representa por una serie de bits. El número de bits por el cual es representado un ND determina su *resolución radiométrica*. Tradicionalmente, los sensores han codificado los ND en grupos de 8 bytes (1 byte), pero hoy en día podemos encontrar varios sensores que codifican la información en un rango más amplio.
- *La organización lógica de los datos*: Un valor de una imagen hiperespectral viene determinado por su posición espacial en la misma (píxel, ejes x e y), y la longitud de onda en el espectro electromagnético (banda, eje z). Al ser una imagen hiperespectral un conjunto de mediciones con una localización espacial, donde cada medición posee un vector de datos que corresponde a la información espectral, podemos encontrar diferentes políticas a la hora de organizar los datos correspondientes en el fichero. Las más comunes son BSQ, BIL y BIP:
 - *Bandas secuenciales (Band Sequential, BSQ)*: Los ND de cada banda se disponen uno a continuación del otro hasta completar todos los píxeles que la forman. Tras ellos, se sitúan los ND de la siguiente banda y así sucesivamente hasta completar todas las bandas espectrales de la imagen (Figura 3.19(b)).
 - *Bandas intercaladas por línea (Band Interleaved by Line, BIL)*: Los ND se organizan por líneas en lugar de por bandas, disponiéndose consecutivamente los píxeles correspondientes a todas las bandas para cada línea, antes de empezar con la siguiente (Figura 3.19(c)).
 - *Bandas intercaladas por píxel (Band Interleaved by Pixel, BIP)*: Los ND se organizan por píxeles en lugar de por bandas, disponiéndose consecutivamente todos los píxeles con todas sus bandas, antes de empezar con el siguiente (Figura 3.19(d)).

Estas políticas de almacenamiento de ficheros de datos no son las únicas que existen [Gen08], sirviendo en muchas ocasiones como base para desarrollar otras alternativas por parte de los proveedores y agencias que distribuyen las imágenes. Entre los formatos más utilizados hoy en día para el almacenamiento de imágenes hiperespectrales podemos encontrar el HDF, el CDF, el NetCDF, el GeoTIFF, el JPEG2000 o el FITS, los cuales han sido detallados en la Sección 3.2.6.3. De todos ellos, el que se está convirtiendo en más popular es el HDF, usado ampliamente por la NASA para el programa Earth-Observing System (EOS).

3.2.6.3. Formatos de datos estándar.

A continuación se listan algunos de los formatos estándar más utilizados para el almacenamiento de imágenes hiperespectrales:

- *HDF (Hierarchical Data Format)* [HDFwww, Koz10, Yan05]: Es un formato de fichero de datos diseñado por el National Center for Supercomputing Applications (NCSA) con el objetivo de

```

ENVI
description = {
  Create New File Result [Thu Oct 14 11:21:54 2010];
  samples = 407
  lines = 417
  bands = 224
  header offset = 0
  file type = ENVI Standard
  data type = 2
  interleave = bsq
  sensor type = Unknown
  byte order = 0
  map info = {UTM, 1.000, 1.000, 654738.505, 4012104.558, 1.6600000000e+001, 1.6600000000e+001,
  11, North, units=Meters, rotation=-9.00000000}
  wavelength units = Unknown
  band names = {
    Georef (Resize (Resize (Band 1:raw):[Memory1]):raw),
    Georef (Resize (Resize (Band 2:raw):[Memory1]):raw),
    Georef (Resize (Resize (Band 3:raw):[Memory1]):raw),
    Georef (Resize (Resize (Band 4:raw):[Memory1]):raw),
  }

```

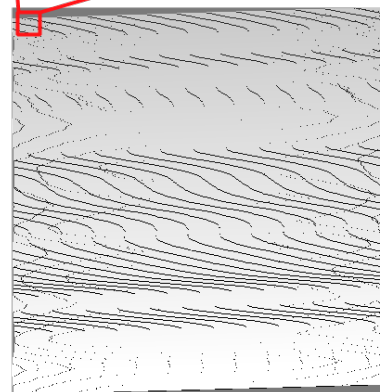
(a)

	169	170	171	172	173
603	655343.71	655360.49	655377.25	655384.01	655410.77
604	655339.05	655355.82	655372.59	655389.35	655406.11
605	655333.94	655350.71	655367.48	655384.24	655401.00
606	655328.53	655345.30	655362.07	655378.83	655395.59
607	655322.98	655339.75	655356.52	655373.29	655390.05
608	655317.57	655334.35	655351.12	655367.89	655384.65
609	655312.76	655329.54	655346.31	655363.08	655379.84
610	655309.47	655326.25	655343.02	655359.79	655376.55
611	655305.97	655322.75	655339.52	655356.29	655373.06
612	655302.34	655319.12	655335.90	655352.67	655369.43
613	655298.81	655315.59	655332.36	655349.13	655365.90
614	655295.75	655312.53	655329.31	655346.08	655362.85
615	655293.73	655310.50	655327.28	655344.05	655360.82



(b)

	4	5	6	7	8
10	0	0	0	0	0
11	0	0	0	0	0
12	0	0	0	0	0
13	567	567	567	567	567
14	568	568	568	568	568
15	569	569	569	569	569
16	570	570	570	570	570
17	-571	-571	-571	-571	-571
18	571	571	-571	-571	-571
19	572	572	572	572	572
20	573	573	573	573	573
21	574	574	574	574	574
22	575	575	575	575	575



(c)



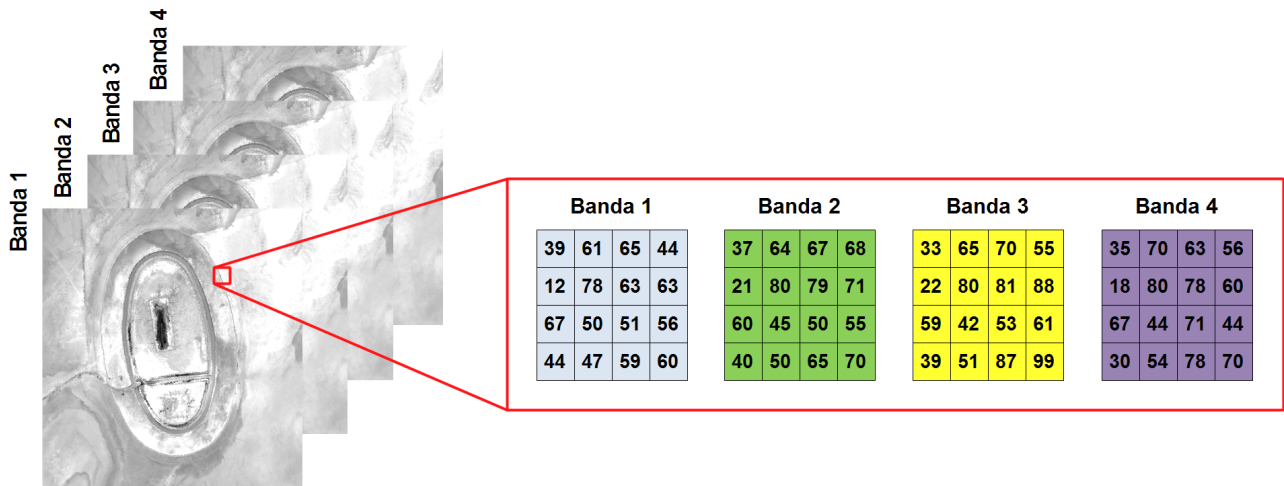
(d)



(e)

Figura 3.18: Formatos básicos de ficheros de imágenes.

- (a) Fichero cabecera. (b) Banda de longitudes del fichero IGM. (c) Banda de posiciones en X del fichero GLT.
- (d) Banda de imagen del fichero de datos L0. (e) Banda de imagen del fichero de datos geocorregido.



(a)

39	61	65	44
12	78	63	63
67	50	51	56
44	47	59	60
37	64	67	68
21	80	79	71
60	45	50	55
40	50	65	70
33	65	70	55
22	80	81	88
59	42	53	61
39	51	87	99
35	70	63	56
18	80	78	60
67	44	71	44
30	54	78	70

(b)

39	61	65	44
37	64	67	68
33	65	70	55
35	70	63	56
12	78	63	63
21	80	79	71
22	80	81	88
18	80	78	60
67	50	51	56
60	45	50	55
59	42	53	61
67	44	71	44
44	47	59	60
40	50	65	70
39	51	87	99
30	54	78	70

(c)

39	37	33	35
61	64	65	70
65	67	70	63
44	68	55	56
12	21	22	18
78	80	80	80
63	79	81	78
63	71	88	60
67	60	59	67
50	45	42	44
51	50	53	71
56	55	61	44
44	40	39	30
47	50	51	54
59	65	87	78
60	70	99	70

(d)

Figura 3.19: Políticas de almacenamiento del fichero de datos.

(a) Detalle de las bandas de una imagen hiperespectral. (b) Formato BSQ. (c) Formato BIL. (d) Formato BIP.

ayudar a los usuarios en el almacenamiento y manipulación de datos científicos a través de diferentes sistemas operativos y máquinas. Soporta una gran variedad de tipos de datos: vectores de datos científicos, tablas, anotaciones de texto, así como varios tipos de imágenes *raster* con sus respectivas paletas de colores. Los ficheros HDF son también auto-descriptivos. Por cada objeto de datos almacenado, existen etiquetas pre-definidas que identifican dicha información con el tipo de datos, la cantidad de datos, sus dimensiones, y su localización dentro del fichero.

- *CDF (Common Data Format)* [CDFwww, TIB06, Dhr03]: Es un formato de ficheros auto-descriptivo y multi-plataforma desarrollado por el Goddard Space Flight Center de la NASA, y orientado al almacenamiento y manipulación de datos escalares y multi-dimensionales. Una característica importante de CDF es que puede manejar set de datos que pueden ser tanto multi-dimensionales como escalares. Para hacer esto, CDF agrupa los datos por *variables* cuyos valores están conceptualmente organizados en vectores. Una variable CDF es un nombre genérico de un elemento u objeto que representa los datos, y que no tiene ningún contexto científico asociado a ella. Por ejemplo, datos representando una variable independiente, una variable dependiente,

un valor de fecha y hora, o cualquier otro dato (una imagen, un fichero XML, etc.). En otras palabras, una variable no contiene ningún significado oculto más que el dato en sí. Se puede describir una variable, o la relación con otra variable o variables, mediante *atributos*.

- *NetCDF* (*Network Common Data Form*) [Netwww, Han05, Rew90]: Es un formato de datos desarrollado por la University Corporation for Atmospheric Research (UCAR), originalmente basado en el modelo conceptual del formato CDF de la NASA, pero actualmente se encuentra tan evolucionado que no es compatible con él. Básicamente presenta las mismas estructuras que CDF, pero además se han añadido algunas más, como los *grupos* y tipos definidos por el usuario.
- *GeoTIFF* [Geowww, Rit00, Rit97]: Este formato apareció como solución a las limitaciones de representación de estructuras para el procesamiento de información geográfica del formato TIFF. La especificación del GeoTIFF se basa en un set de etiquetas TIFF que describen toda la información cartográfica asociada a imágenes TIFF originadas en sistemas de adquisición de imágenes desde satélite, obtención de fotografías aéreas, obtención de mapas, modelos de elevación digital, o resultados de análisis geográficos.
- *JPEG2000* [JPEwww, Buc08, Zab06]: Es un protocolo basado en compresión de imágenes estándar y sistema de codificación. Fue creado por el comité Joint Photographic Experts Group en el año 2000 con la intención de reemplazar el estándar original JPEG (creado en el año 1992) basado en la Transformada Discreta de Coseno, con un nuevo método basado en la Transformada de Wavelets. Los mercados y aplicaciones a los que es dirigido el formato JPEG2000 es amplio: dispositivos multimedia, comunicaciones cliente/servidor, análisis de imágenes médicas, teledetección, cine digital, etc.
- *FITS* [FITwww, Wel80, Hu08]: Es un formato orientado al almacenamiento de información de diferente naturaleza, como imágenes multispectrales, tablas, o texto plano. Un fichero de este tipo está formado por una serie de bloques denominados HDUs (Header Data Units), cada uno de ellos compuesto por una cabecera de texto en ASCII, y datos codificados en binario. Cada una de estas cabeceras almacena información sobre su bloque de datos. La primera del fichero (la cabecera de la primera sección) es denominada *cabecera principal* y contiene información sobre el fichero en general y su bloque de datos, también principal. Las siguientes secciones se pueden denominar *extensiones*. Las cabeceras contienen infinidad de información acerca de sus respectivos bloques de datos, tal como tamaño del mismo, origen, formato, comentarios, historial de datos, etc. Además, permite la posibilidad de que el usuario pueda definirse sus propios campos.

3.3. Sistemas de procesamiento en plataformas.

El tratamiento a gran escala de imágenes en el campo de la teledetección, las cuales incluyen metodologías avanzadas como la detección de cambios, monitorización del clima, entre otras, están ayudando a la humanidad a mejorar su estilo de vida y su conocimiento sobre la Tierra y el Universo [Gro13]. Estas técnicas son constantemente analizadas y optimizadas, con el objetivo de conseguir resultados cada vez más precisos. A medida que estas técnicas se hacen más complejas, se incrementa la necesidad de adaptar los sistemas de computación para maximizar el rendimiento global [Key04].

El procesamiento a bordo de la plataforma (tanto en avión como en satélite) proporciona muchas posibilidades desde un punto de vista de la computación, siendo muy útil para el análisis en tiempo real y para el pre-procesamiento de datos, sobre todo cuando existen ciertas limitaciones en la transferencia de datos de la plataforma a Tierra [Pia99, Cha01, Cet04]. La adaptación de técnicas de monitorización de la corteza terrestre a un nivel de computación en tiempo real permite la creación de redes de alerta que pueden mejorar considerablemente la calidad de vida en el planeta [Lew09], por lo que en ocasiones se convierten en fases clave de la cadena de procesamiento.

Las características de los sistemas de procesamiento a bordo de aviones difiere completamente de los sistemas de procesamiento a bordo de satélites [Kno07]. Únicamente es necesario estudiar y analizar el entorno en el que van a realizar sus operaciones para corroborar este hecho:

- Los sistemas situados en plataformas a bordo de aviones trabajan a gran altitud donde las turbulencias de diferentes zonas de la atmósfera proporcionan el mayor porcentaje de imprecisión en la recogida de los datos. Sin embargo, los sistemas situados en plataformas espaciales no se ven afectados por dichas turbulencias al encontrarse fuera de la atmósfera terrestre, pero realizan sus operaciones en un entorno radiactivo. Este entorno es generado por las fusiones nucleares llevadas a cabo en el núcleo del Sol y de las estrellas (radiación cósmica), las cuales liberan partículas de alta energía que pueden alterar la electrónica del satélite, y como consecuencia, los resultados generados por las metodologías.
- La altitud de las plataformas situadas en aviones es mucho menor que las de las plataformas situadas en satélites, lo que implica diferentes resoluciones espaciales para las imágenes adquiridas. Por tanto, metodologías que necesiten una alta resolución espacial (análisis subpíxel, detección de *targets*, etc.) mostrarán un mayor rendimiento en sistemas de procesamiento aerotransportados, dependiendo de las necesidades de la misión. Sin embargo, metodologías que necesiten una baja resolución espacial pero una gran cobertura para un análisis global de la superficie terrestre (análisis y predicción meteorológicas, estudio de aerosoles en atmósfera, análisis de la salinidad de mares y océanos, temperatura superficial terrestre, etc.) necesitan estar situadas en plataformas espaciales a gran altitud.
- La resolución temporal de cada tipo de plataforma difiere también. Los aviones permiten la adquisición de una determinada región de interés durante un tiempo limitado (el cual puede abarcar varios escaneos de la misma zona), por lo que se hace necesaria una planificación previa del vuelo y de la trayectoria a seguir. Si se especifica en los objetivos de la misión, se pueden establecer diferentes fechas de visita para un análisis temporal del área de interés, pero depende de las necesidades e intereses, y la resolución temporal en este caso es un factor variable. Para el caso de plataformas situadas en satélites, la resolución temporal es constante, abarcando de varios días a varias semanas, dependiendo de las características de la órbita definida durante la especificación de la misión espacial. Esto potencia el desarrollo de metodologías basadas en la detección de cambios o la fusión de datos.
- La energía disponible para el sistema de computación y el resto de dispositivos presentes en cada tipo de plataforma es, en ambos casos, limitada, pero su obtención, almacenamiento y distribución es más sencilla en un avión que en un satélite, en el cual además es necesario calcular, a partir de su órbita, las horas en las que estará expuesto al Sol y en las que estará oculto, siempre jugando con las limitaciones de los paneles solares y las baterías para no perder la autonomía durante el transcurso de la misión.

Por tanto, tras analizar en detalle las diferencias en los entornos en los que trabajarán las plataformas y los dispositivos tanto a bordo de aviones como de satélites, es posible comprobar que las características de los sistemas de procesamiento serán totalmente diferentes.

Para el caso de las plataformas a bordo de aviones, la alta capacidad de autonomía y gestión de la energía, así como la posibilidad de albergar analistas y expertos durante el vuelo como tripulantes, hace que los sistemas de computación utilizados sean de lo más variado [Fal02]. Para empezar, se podría decir que no existe un módulo físico diseñado e implementado de manera específica de acuerdo a los instrumentos que obtendrá la información, restos de dispositivos y necesidades de la misión, sino que el elemento central recae en el sistema de adquisición, el cual proveerá los datos de la región de interés [Sch05]. Existen sistemas de adquisición proporcionados por empresas e instituciones, cuyas características pueden suponer una ventaja o inconveniente de acuerdo a los objetivos que se quieran cumplir. Estos sistemas de adquisición están formados principalmente por los instrumentos (sensores),

que se encargarán de la obtención de la información, y adicionalmente por el sistema de posicionamiento GPS/INS [Cra01, Mos00]. La electrónica presente en estos sistemas está compuesta por módulos hardware / software que se encargan de la correcta sincronización entre la información instrumental y espacial a partir de parámetros de vuelo configurables [Ror12].

Al seguir este esquema, uno de los puntos más importantes, en el que se centra gran parte del esfuerzo durante el desarrollo, es la manera en la que se van a transferir los datos al exterior, o la manera en la que van a ser accesibles. Algunos sistemas de procesamiento contienen una memoria interna en la que almacenan los datos obtenidos para su posterior distribución [Gon12], pero si se desea realizar procesamiento en tiempo real es necesario que los datos estén disponibles durante el vuelo. Por ello, es necesario diseñar un formato de datos y un protocolo de comunicaciones para que el envío de la información adquirida sea óptimo [Yan08].

Este tipo de tecnología, junto a la autonomía propia que ofrece un avión, permite la instalación de una computadora que se encargue de la recolección de la información proporcionada por el sistema de adquisición y posicionamiento de datos, así como del procesamiento de los datos en tiempo real, de acuerdo a las metodologías implementadas [Ska10]. Esta computadora puede haber sido diseñada específicamente para la misión por una entidad externa al fabricante del sistema de adquisición y posicionamiento de datos [Bev92], o bien puede tratarse de una computadora de propósito general (PC, portátil, etc.). Esta última opción es muy atractiva, ya que permitiría la posibilidad de implementar algoritmos utilizando herramientas de programación convencionales con un bajo coste. Además, sería posible la implementación de metodologías y técnicas avanzadas de procesamiento, como la utilización de varias computadoras para el tratamiento paralelo de la información [Kur10].

En cuanto a las plataformas a bordo de satélites, hay que considerar que los sistemas de procesamiento correspondientes estarán limitados a la capacidad energética del satélite [Rey09] y que no estarán supervisados y controlados por ninguna tripulación, por lo que se requiere que el sistema sea lo más autónomo e inteligente posible [Ste01]. A estos parámetros hay que añadirles el entorno radiactivo en el que llevará a cabo sus funciones exige que éste sea robusto y tolerante a fallos [Cza05]. Además, igual que ocurría con los sistemas aero-transportados, la política de comunicaciones es muy importante debido a que todos los datos recogidos por los instrumentos son enviados a Tierra para su posterior almacenamiento y procesamiento [Air09, Bez12, Sta01]. Pero lo que hace realmente atractiva la posibilidad de estudiar y analizar la corteza terrestre desde el Espacio es la capacidad de poder generar resultados desde el propio satélite y enviarlos directamente a Tierra [Bre10, Liu12, Hal11]. Por este motivo, y por las diversas metodologías que pueden albergar (dependiendo de los objetivos de la misión espacial), por normal general los sistemas de procesamiento a bordo de plataformas espaciales son sistemas de propósito específico, los cuales deben considerar los siguientes criterios [Eic12]:

- Aparte de las metodologías que permitirán la generación de resultados desde el Espacio, también existirán módulos adicionales de gestión (como control del satélite, del *payload*, y otras funciones), por los que se hará necesario compartir espacio y energía.
- Deben ser lo suficientemente robustos para aguantar vibraciones y movimientos bruscos, los cuales se producen principalmente durante la fase de lanzamiento.
- Deben ser robustos al entorno radiactivo del Espacio. Además, es necesario considerar que este aspecto no se limita únicamente al impacto producido por partículas altamente energéticas, sino que éstas tienden a acumularse a lo largo de la vida del satélite.
- Deben ser robustos a las diferencias termales producidas por el hecho de tener sólo una parte del satélite expuesta al Sol.
- Deben ser robustos a condiciones químicas extremas, presentes en las capas más altas de la atmósfera terrestre (especialmente satélites con órbita de altitud baja). Un ejemplo es el Oxígeno atómico, el cual produce una degradación, erosión y contaminación de los materiales con los que son construidos los satélites, acortando su tiempo de vida.

- La cantidad de energía consumida debe ser mínima, puesto que los paneles solares y las baterías presentes en el satélite (junto con la gestión realizada por el sistema de generación de energía) la produce de manera limitada.
- Es imprescindible la adición de módulos de detección y corrección de errores, ya que el entorno radiactivo del Espacio puede alterar el funcionamiento de la electrónica presente en el satélite.

En este contexto, han surgido diferentes propuestas de sistemas de computación a bordo de satélites, los cuales pretenden ser los más generales posibles, optimizando su configurabilidad para adaptarlos a las necesidades de las misiones espaciales [Zha10]. En un principio, los procesadores MIL-STD-1750 A/B (Figura 3.20) dominaron el diseño de los satélites durante más de una década [Pas95]. Este tipos de procesadores, combinados con dispositivos adicionales, como las unidades de gestión de memoria básicas como Pyramid Semiconductor P1753 [Pyr05] y Dynex MA31751 [DYN99], o más avanzadas como Direct Memory Access Controllers (DMAC) [Kam08]. Estos dispositivos conformaban una estructura modular multi-chip que fueron utilizados en diferentes satélites como Intelsat VII [Los12], European Remote Sensing ERS 1 y 2 [Ver01], o Envisat [Dub01].

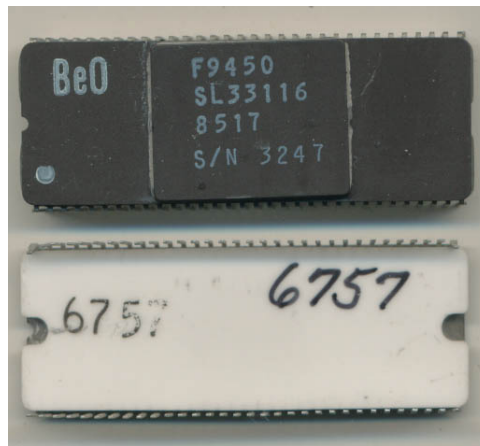


Figura 3.20: *Procesador Fairchild F9450 (MIL-STD-1750 A).*

Sin embargo, al tratarse de componentes aislados cuyo trabajo en conjunto formaban el sistema de procesamiento, se requería una alta compatibilidad y una especial atención a la manera en la que se transferían los datos, por lo que la complejidad de la electrónica era bastante elevada [Cve93]. Teniendo como objetivo su reducción y la generación de una serie de diseños que apostaran por un sistema de procesamiento compacto, se empezaron a introducir procesadores Reduced Instruction Set Computer (RISC) [Dan03] (Figura 3.21(a)), de propósito general, los cuales disponían de un básico repertorio de instrucciones, pero muy apto para la realización de cálculos desde el Espacio [Gil05, Vla04]. Esto era debido a que cada instrucción era ejecutada en un único ciclo de reloj y además permitía frecuencias de reloj muy altas. De esta manera, se potenció el desarrollo de las metodologías a un nivel de software, más sencillas, rápidas y fiables de programar, aunque menos livianas. Diferentes alternativas fueron estudiadas posteriormente, como la inclusión de procesadores Complex Instruction Set Computers (CISC) [Sin10] (Figura 3.21(b)), también de propósito general, con un repertorio de instrucciones mucho más amplio y rápidas. Pero la simplicidad del repertorio de instrucciones de los procesadores RISC, así como su tecnología basada en un número muy bajo de transistores (que lo conformaban como un sistema robusto contra la radiación y los efectos electromagnéticos) [Zar96], permitió que fuera este tipo de procesadores el más utilizado en los sistemas de procesamiento espaciales [Com94].

La tendencia en el diseño de los sistemas de procesamiento a bordo de plataformas espaciales es la misma que para los sistemas de procesamiento de propósito general: la integración de todos los módulos implicados en la computación en un único chip o circuito integrado (System-on-Chip, SoC) [Gre11, Fan04]. De esta manera, la funcionalidad ofrecida por los procesadores RISC pudo ser rápidamente completada mediante la adición de diferentes módulos, como memorias gestionadas por

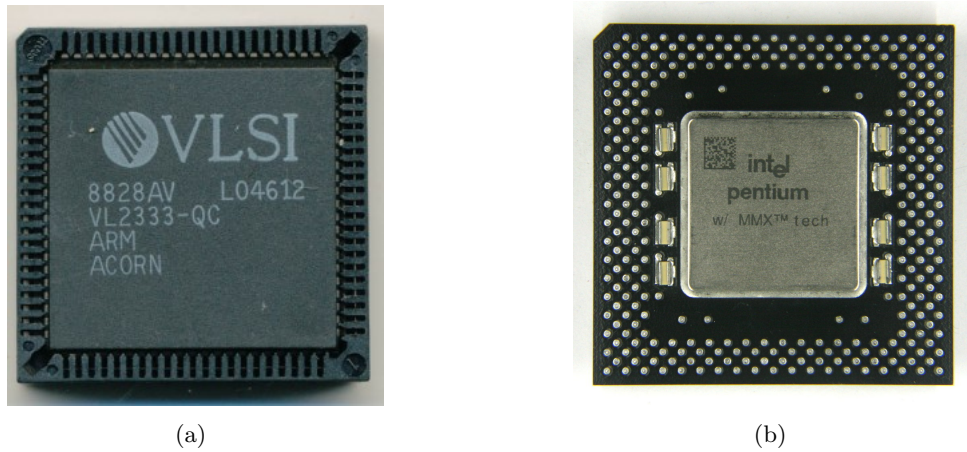


Figura 3.21: *Procesadores convencionales de propósito general RISC y CISC.*
 (a) *Procesador VLSI VL2333-QC (RISC).* (b) *Procesador Intel FV80503200 (CISC).*

controladoras Direct Access Memory (DMA), registros, operadores matemáticos dedicados (bloques DSP), etc., permitiendo una evolución a sistemas completos integrados basados en procesadores RISC [Buj09].

Uno de los primeros sistemas destacables fue el modelo ERC32 [Ram97, Tha00, Gai94], el cual estaba formado por tres chips principales: una unidad de enteros, una unidad de coma flotante y un controlador de memoria. Las tres unidades estaban interconectadas entre sí mediante dos buses para datos e instrucciones y que permitían la transferencia de la información con el exterior mediante buffers dedicados (Figura 3.22).

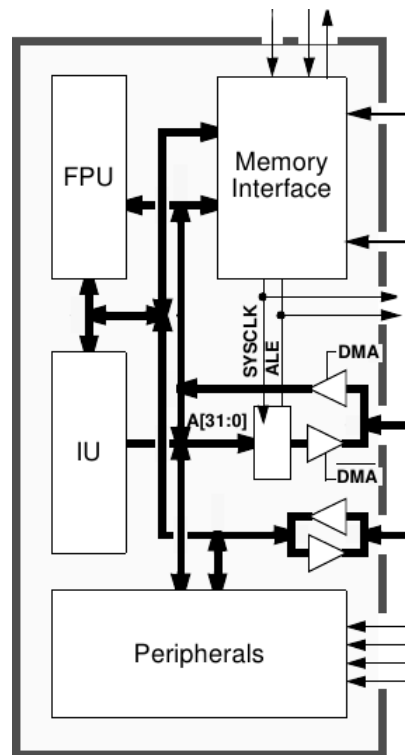


Figura 3.22: *Diagrama de bloques de la arquitectura ERC32.*

Entre estos tipos de sistemas destaca la conocida arquitectura LEON3FT [Aer13], una aproximación a la generación de procesadores LEON SPARC V8 [Gai07], pero con una tolerancia a fallos maximizada y una circuitería optimizada para ser ejecutada en el Espacio [Gai05]. Esta arquitectura se basa en una serie de módulos interconectados por una serie de buses (Figura 3.23), en

los que se encuentran [Aer10]:

- Procesador LEON3 SPARC V8 con repertorio de instrucciones de 8 KB y caché de datos de 8 KB.
- Unidad de punto flotante IEEE-754.
- Controlador de memorias PROM y SRAM externas con detección y corrección de errores.
- Controlador de memorias SDRAM externas con detección y corrección de errores.
- Controlador de puertos de entrada / salida de propósito general (General Purpose I/O, GPIO) de 16 bits.
- Unidad temporizador y *watchdog* (reseteo en caso de fallo).
- Controlador de interrupciones.
- Interfaces de conexión UART y JTAG.
- 4 módulos SpaceWire.
- Hasta 2 controladoras Controller Area Network (CAN).
- Controladora Ethernet.
- Interfaz de comunicaciones CompactPCI.

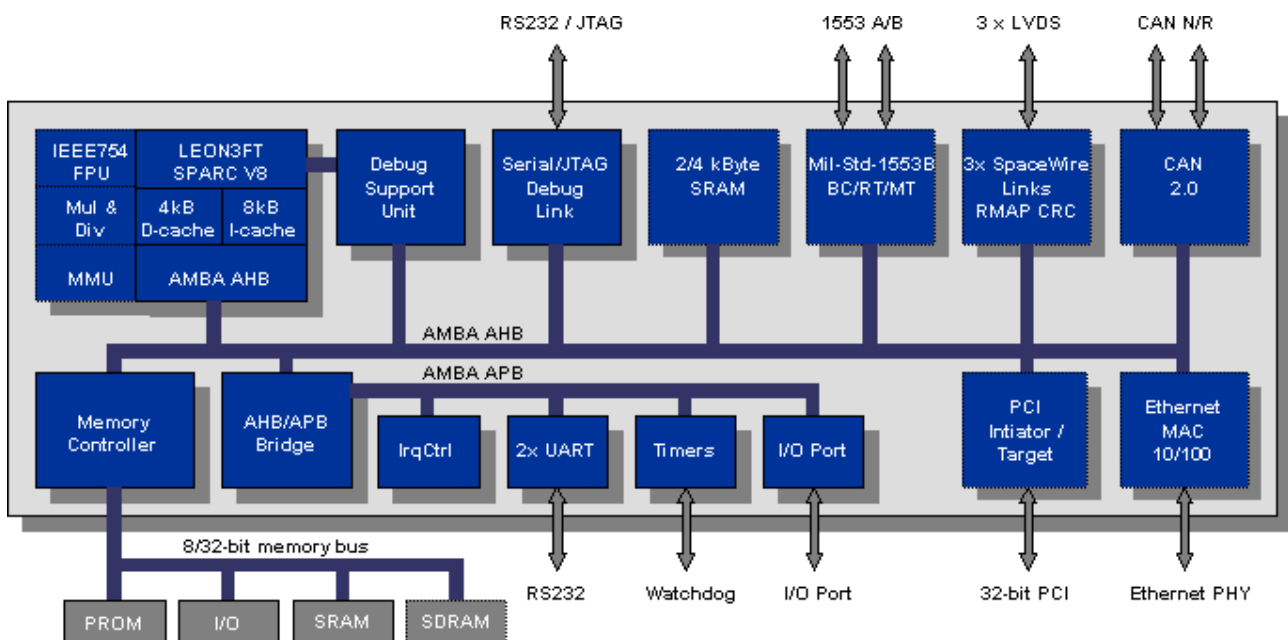


Figura 3.23: Diagrama de bloques de la arquitectura LEON3FT.

A pesar de que sistemas SoC, como ERC32 y LEON3FT son ampliamente utilizados como sistemas de procesamiento central de plataformas espaciales, existe un gran interés general en utilizar arquitecturas hardware reconfigurables [Mar04]. Como su nombre indica, este tipo de tecnologías se basan en un tipo de hardware que puede ser re-diseñado, re-implementado y re-configurado tanto de manera local como remota. La tecnología más representativa de este tipo de arquitectura son las Field Programmable Gate Array (FPGA) [Ros93, Sad11], las cuales gozaron de gran reconocimiento en el ámbito aero-espacial con la elaboración y aprobación del documento RTCA/DO-254 [Mar09, Cer07]. En este documento, desarrollado por la organización RTCA Incorporated [RTCwww], se establecían las

pautas para analizar, diseñar e implementar sistemas de computación hardware, con especial atención en los dispositivos FPGA [DeL11]. Aunque inicialmente el documento estaba enfocado a plataformas aerotransportadas, pronto se adaptarían a las plataformas espaciales [Xil11a, Syn12]. De esta manera, los dispositivos FPGA se establecerían como una gran alternativa en todo lo referente a los sistemas de computación reconfigurables, debido a principalmente a sus características [Gra03]:

- *Bajo consumo de energía:* El uso de una tarjeta FPGA supone un consumo de energía mucho menor que dispositivos hardware análogos implementando la misma metodología. Este es un aspecto importante ya que la energía disponible en un satélite es limitada y además debe ser compartida con el resto de dispositivos.
- *Tecnología barata:* Las tarjetas FPGA son un tipo de tecnología barata de fabricar en serie. Además, debido a la gran oferta de tarjetas FPGA existente actualmente en el mercado, es posible encontrar modelos simples y muy baratos, o más complejos y más caros, por lo que el precio puede ajustarse de acuerdo a las necesidades de la misión espacial. Existen modelos de FPGA orientados al ámbito militar y aeroespacial, que presentan componentes mucho más robustos, así como chasis especiales antirradiación y módulos de detección y corrección de errores. Pero sus precios superan a las tarjetas FPGA convencionales con un alto orden de magnitud, por lo que es interesante poder adaptar las de propósito general para realizar procesamiento a bordo.
- *Hardware reconfigurable:* Las tarjetas FPGA ofrecen una velocidad de procesamiento rápida, característica de las implementaciones a muy bajo nivel, pero con la flexibilidad y la reprogramabilidad propias de las implementaciones software. La posibilidad de poder reconfigurar el hardware de manera remota supone una gran ventaja. Es posible realizar modificaciones, nuevas versiones, parches, ampliaciones y detección y corrección de errores del sistema de manera remota una vez el satélite está ya en órbita, reduciendo el coste del mantenimiento de manera considerable. Además, otra opción muy atractiva es la posibilidad de poder re-utilizar satélites cuando las misiones para las que fueron desarrollados y lanzados terminen. De esta manera, sería posible reconfigurarlo para nuevos propósitos y de acuerdo a las necesidades y requisitos de nuevas misiones espaciales.

Las características que ofrecen los dispositivos FPGA hacen de éstas un sistema de computación capaz de ejecutar metodologías complejas a muy bajo nivel, con gran rapidez y a bajo coste, potenciando la paralelización de los datos durante su procesamiento y potenciando las técnicas que requieren una respuesta en tiempo real. Además, su reconfigurabilidad las hace muy atractivas, ya que sería posible la corrección de errores o la actualización de la circuitería a un nivel de hardware, así como la redefinición de nuevas misiones una vez finalizadas las anteriores, reaprovechando el satélite y su *payload*.

A continuación se muestra de manera detallada la composición y funcionamiento de los dispositivos FPGA (Sección 3.3.1), y su adaptación a los sistemas de procesamiento en plataformas espaciales (Sección 3.3.2), dedicando especial atención a un tipo de sistema reconfigurable basado en esta tecnología: SpaceCube.

3.3.1. Sistemas basados en FPGA.

Un sistema FPGA (Field Programmable Gate Array) es un dispositivo semiconductor que contiene bloques de lógica cuya interconexión y funcionalidad puede ser configurada *in situ* mediante un lenguaje de descripción especializado. La lógica programable puede reproducir desde funciones tan sencillas como las llevadas a cabo por una puerta lógica o un sistema combinacional hasta complejos sistemas en un chip.

Los dispositivos FPGA se utilizan en aplicaciones similares a los circuitos integrados de propósito específico (Application-Specific Integrated Circuit, ASIC) [Fra00] sin embargo son ligeramente más lentos, tienen un mayor consumo de potencia y es difícil que puedan abarcar sistemas tan complejos como ellos [Hus10]. A pesar de esto, los dispositivos FPGA tienen como ventajas ser reprogramables (lo que añade una enorme flexibilidad al flujo de diseño), sus costes de desarrollo y adquisición son mucho menores y el tiempo de desarrollo es también muy inferior.

El chip FPGA está compuesto por una estructura matricial de bloques lógicos configurables (Configurable Logic Block, CLB) [Kuo07] que son los responsables de implementar una determinada metodología (Figura 3.24). A pesar de que existen infinitos diseños, como modelo general los bloques CLB se presentan como un circuito conformado por una tabla LUT (Look-Up Table) de 4 entradas que será programada para generar una determinada salida a partir de dichas entradas (Figura 3.25). Además, se integrará un *flip-flop* (biestable) tipo D para sincronizar la salida de acuerdo a la frecuencia del reloj (CLK). Cada bloque CLB presenta un total de 5 registros (*pines*) que permitirá las comunicaciones con los bloques CLB vecinos, dedicando 4 de ellos para entradas, y 1 para salida. Modelos de sistemas FPGA más innovadores amplían las tablas LUT a 6 entradas, modos asíncronos y síncronos, modos aritméticos adicionales, 6 pines (4 para entradas y 2 para salidas), etc., incrementando la complejidad de la circuitería del bloque CLB, pero a la vez su rendimiento y eficacia.

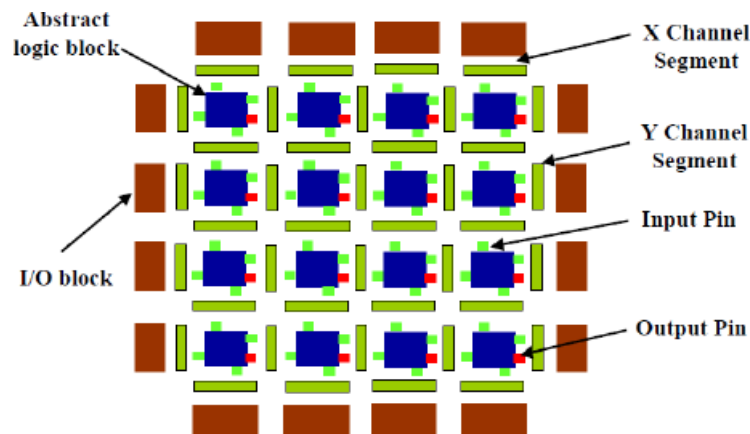


Figura 3.24: Vista interna de un chip FPGA.

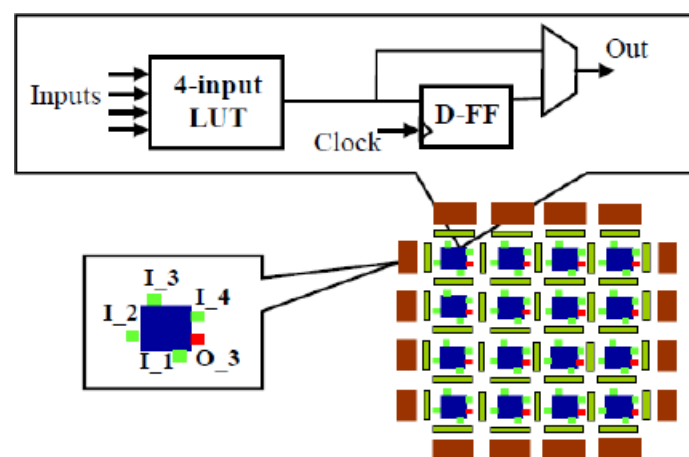


Figura 3.25: Diagrama de un bloque CLB.

La matriz de bloques CLB está rodeada por una serie de registros que permiten las comunicaciones de las metodologías implementadas en la FPGA con el exterior. Los bloques CLB inmediatos a dichos registros son bloques especiales (I/O Block, IOB) que se encargan de la transferencia de datos de entrada y salida a los bloques CLB internos.

Cada bloque CLB está interconectado con sus 4 vecinos mediante un *switch* programable que

permite la transferencia de datos entre los *pins* de entrada y salida correspondiente. De esta forma, durante la compilación de una metodología se establece una conversión del funcionamiento del mismo a modo de varios niveles de tablas LUT representados en los bloques CLB, así como una programación de los *switch* para una correcta transferencia de datos entre los mismos. Estos *switches* pueden actuar a un nivel horizontal, o bien vertical, creando una red de interconexión compleja (Figura 3.26). Entre los bloques CLB existen bloques especiales que también participan en el enrutamiento de datos (Figura 3.26(b)), como son bloques de memorias RAM, o los muy utilizados bloques Digital Signal Processing (DSP) [Pha13]. Concretamente, estos bloques DSP (Figura 3.27) implementan metodologías como circuitos integrados, entre los que destacan operaciones trigonométricas, aritméticas, en coma flotante, etc., y son muy importantes a la hora de configurar el diseño hardware de la metodología, ya que suponen una minimización del número de ciclos de reloj respecto a sus versiones convertidas a bloques CLB. La capacidad del chip FPGA siempre está limitado por el número de bloques que la conforman (CLB, IOB, DSP, etc.), por lo que es siempre necesario controlar los recursos consumidos por la metodología implementada para un óptimo funcionamiento.

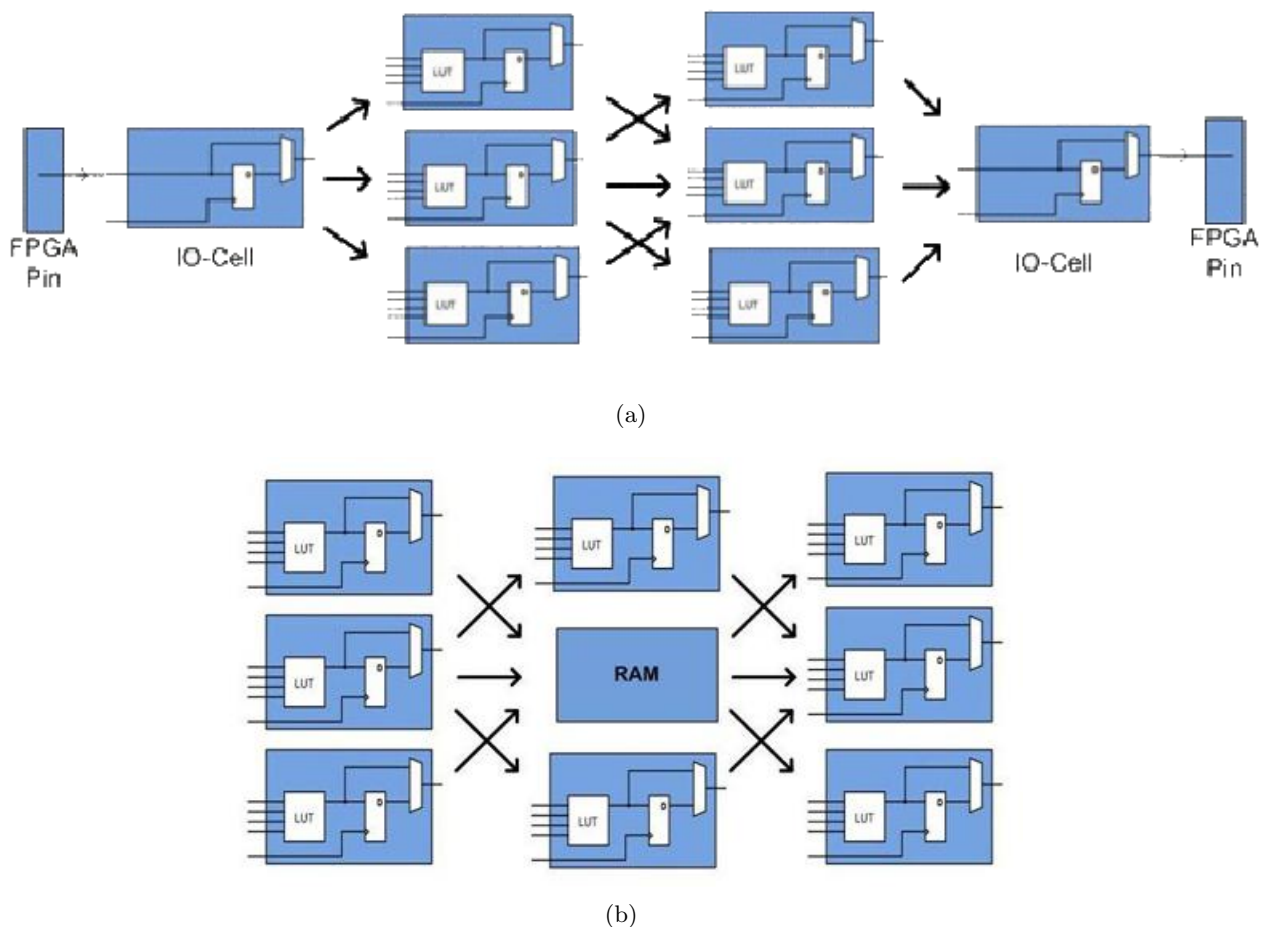


Figura 3.26: Comunicaciones entre los bloques CLB dentro de un chip FPGA.

(a) Establecimiento de comunicación entre extremos de un chip FPGA. (b) Comunicaciones entre bloques CLB y bloques especiales de propósito específico.

Una tendencia común es combinar los bloques lógicos presentes en las FPGA con microprocesadores para formar un *Sistema en un Chip Programable* (Programmable System-on-Chip, SOPC) [Mea01, Kha12], así como con periféricos de comunicaciones con el exterior utilizando conectores y protocolos convencionales, componiendo en su conjunto una tarjeta de desarrollo. Ejemplo de tales tecnologías híbridas pueden ser encontradas en los dispositivos Virtex-II PRO [Xil05] y Virtex-4 de Xilinx [Xil09a], los cuales incluyen uno o más procesadores PowerPC [Del11] embebidos junto a la lógica FPGA (Figura 3.28). El FPSLIC de Atmel [Atm01a] es otro dispositivo similar, el cual usa un procesador AVR en combinación con la arquitectura lógica programable de Atmel [Atm01b].

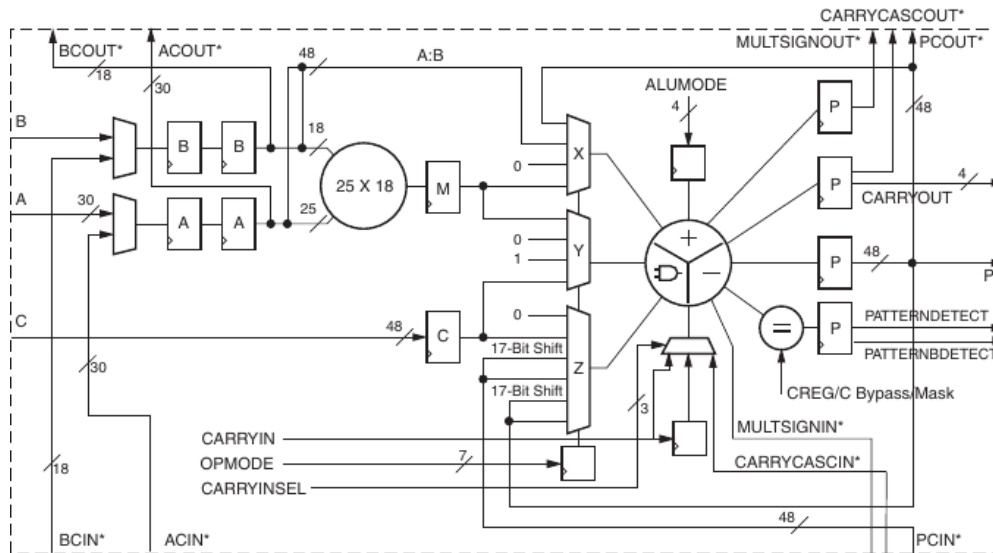


Figura 3.27: Vista interna de un bloque DSP48E, un tipo de bloque DSP desarrollado por Xilinx.

Otra alternativa es hacer uso de núcleos de procesadores implementados mediante implementaciones de la lógica del FPGA, los cuales son conocidos como *soft-processor*, como alternativa a procesadores hardware físicos *hard-processor*. Esos núcleos incluyen los procesadores MicroBlaze [Xil12a] y PicoBlaze [Xil11b] de Xilinx, Nios [Alt00] y Nios II [Alt08] de Altera, y los procesadores de código abierto LatticeMico8 [Lat11] y LatticeMico32 [Lat12].



Figura 3.28: Sistema FPGA híbrido Xilinx Virtex-II PRO.

Actualmente, muchas tarjetas FPGA modernas soportan la reconfiguración parcial del sistema, permitiendo que una parte del diseño sea reprogramada, mientras las demás partes permanecen intactas [Lie09]. Estos dispositivos se acercan todavía más al principio de la idea de la *computación reconfigurable*, o *sistemas reconfigurables*.

3.3.2. Sistemas FPGA a bordo de satélites.

Los sistemas basados en FPGA son comúnmente utilizados en dispositivos de propósito general, pero últimamente están siendo empleados en sectores en las que son sometidas a ambientes radiactivos (por ejemplo satélites, aviones, etc...) [Nor09, Pin08], ya que son muy atractivas al proporcionar dos ventajas principales. La primera de ellas es que cumple con los requisitos de alta densidad (la metodología está representada por un elevado número de puertas lógicas y una electrónica avanzada),

de tamaño y peso (la circuitería se encuentra en un espacio físico pequeño y ligero), de consumo (la energía necesaria para que un dispositivo FPGA funciones es mínima) y de coste (el coste de fabricación de los dispositivos FPGA es relativamente bajo, por lo que es posible obtenerlas en el mercado a precios asumibles) [Pan98], lo que las hace aptas como sistemas de computación en plataformas aéreas y espaciales. La segunda es la facilidad de replicación de los algoritmos, potenciando técnicas de detección de errores basados en la duplicación o triplicación de la circuitería [Car99], como Triple Modular Redundancy (TMR) [Goo13], haciéndolas más robustas y tolerantes a fallos. Generalmente, todas las prácticas de diseño digital estándar que se enseñan en las universidades y que son utilizados en productos comerciales pueden ser convertidos a sus equivalentes en productos basados en FPGA [Sch10]. El problema principal surge a la hora de estudiar el entorno en el que será ejecutada la metodología implementada en la tarjeta FPGA. Un caso muy común surge a la hora de desarrollar un sistema satélite, ya que el Espacio es un ambiente radiactivo. Esta radiación consiste en enormes masas eyectadas de materia y partículas de alta energía procedentes del Sol y el resto de estrellas, la cual bombardea la circuitería de los dispositivos, constantemente expuestos de manera directa [Bek11]. La atmósfera de la Tierra se convierte en una barrera fuerte y protectora que absorbe un gran porcentaje de esta radiación a lo largo del espectro electromagnético. Por ello, los sistemas en Tierra no se presentan tan vulnerables, y son precisamente los sistemas satélite (localizados fuera de esta capa protectora) los que deben prestar especial atención en defenderse del efecto radiactivo [Zei12], ya que la electrónica de su circuitería puede ser alterada por las partículas cargadas energéticamente modificando los resultados e introduciendo errores, y llegando a derivar, en casos extremos, en *fallo catastrófico*, en el cual el satélite queda totalmente inutilizable [Ber04].

La manera en la que esta radiación espacial puede afectar a los sistemas basados en FPGA puede dividirse en dos categorías [Tho10]:

1. Efectos de primer orden (directos): Perturbaciones provocadas por la radiación y que afectan directamente a los componentes hardware de la FPGA, dañándolos o haciendo que genere resultados incorrectos.
2. Efectos de segundo orden (indirectos): Perturbaciones provocadas por la radiación y que afectan a los componentes externos a la FPGA (en otros dispositivos) y que hacen que la FPGA funcione de manera incorrecta.

Por tanto, cuando una FPGA es diseñada como producto destinado a convivir en un ambiente radiactivo, han de considerarse multitud de factores para un óptimo desarrollo. Principalmente, los objetivos que deben cubrirse son los siguientes, aunque pueden variar dependiendo de los requisitos de la misión espacial [Le04]:

- Procesamiento en tiempo real de los algoritmos diseñados e implementados, con la precisión y el rendimiento estimado y esperado durante el diseño.
- Diseño y programación inteligente y razonable, que optimicen las facilidades de reconfigurabilidad desde Tierra, de tal manera que las actualizaciones de los algoritmos a bordo se realicen sin complicaciones una vez el satélite está ya en órbita.
- Escalabilidad adecuada que permita realizar de manera holgada las posibles modificaciones en los sistemas de computación y de comunicaciones. Generalmente, se suele dejar el mismo espacio libre que ocupado por la circuitería (2x).
- Demostración de una batería de pruebas lo más similar posible al entorno radiactivo en el que se encontrará el satélite.
- Cumplimiento de las actualizaciones planificadas.

- Correcta interacción con el sistema electrónico del satélite y el resto de componentes y dispositivos.
- Cumplimiento de los requisitos físicos de la misión espacial (tamaño, peso y consumo).
- Duración de vida de misión mínimo de unos tres años, procurando una degradación mínima.
- Tolerancia al ambiente radiactivo del espacio en cualquier altitud orbital, que maximicen la fiabilidad y disponibilidad de los resultados del satélite.
- Viable con riesgos mínimos (en términos de rendimiento, planificación, coste y mantenimiento en órbita).
- Arquitectura abierta y ampliable.

Evitar fallos catastróficos en satélites es el principal móvil en el desarrollo de los mismos y en su reconfiguración remota. Constelaciones enteras de satélites han sido un total fracaso debido a pequeños fallos en su circuitería y en sus componentes lógicos [DeM05]. Los sistemas FPGA, por tanto, aportan un componente de reconfiguración que permite a los satélites ser programados sin la necesidad de ser enviados de vuelta a Tierra, ofreciendo la posibilidad de ser reparados de manera remota. Además, esta tecnología la convierte en potencial a la hora de reutilizar satélites cuyas misiones espaciales ya han finalizado, y reconfigurándolos y adaptándolos a las necesidades de las nuevas.

A pesar de que existen varios diseños e implementaciones, un ejemplo bastante avanzado en desarrollo y ya testado en el Espacio es el dispositivo SpaceCube (Figura 3.34), una familia de sistemas reconfigurables de alto rendimiento, basada en la tecnología FPGA y desarrollada por los ingenieros del centro Goddard Space Flight Center de la NASA, con el objetivo de ejecutar aplicaciones espaciales que requieran procesamiento a bordo o gran consumo de energía [Sea08, Fla11]. La primera versión del SpaceCube (SpaceCube 1.0) estaba basado en las FPGAs comerciales Virtex-4 de Xilinx, aunque actualmente se encuentra en la versión SpaceCube 2.0, con una financiación de 1 millón de dólares. El sistema SpaceCube 2.0 está basado en la FPGA Virtex-5 de Xilinx, la cual ha sido optimizada para minimizar los riesgos de radiación en el espacio.

El sistema SpaceCube supone una gran ventaja frente a los procesadores a bordo anti-radiación tradicionales, siendo las funcionalidades más atractivas la posibilidad de reconfigurar el SpaceCube de manera remota, durante el transcurso de la misión, y la posibilidad de implementar algoritmos que permitan a la nave detectar y reaccionar a eventos o generar productos de datos en tiempo real, reduciendo los requerimientos de almacenamiento de datos a bordo [NAS09b]. Además, gracias a una implementación a bajo nivel de las metodologías, la utilización del sistema SpaceCube ofrece un incremento en la velocidad de procesamiento de hasta 100 veces más rápido que un procesador convencional aerospacial, como por ejemplo Rad750 [Cad02, NAS13].

A pesar de las ventajas que presenta, este sistema también posee varios inconvenientes. La tecnología actual utiliza algunos componentes comerciales de propósito general, lo que supone un abaratamiento del coste, pero que no han sido diseñado para soportar los altos niveles de radiación del espacio. A pesar de añadir un chasis anti-radiación, a la hora de funcionar en el espacio, los efectos de la radiación son inevitables contra estos componentes, por lo que se hace necesario el diseño y desarrollo de algoritmos adicionales que permitan computacionalmente detectar estos errores (bits cambiados de 1 a 0, y viceversa) y corregirlos si es posible [Ber07].

3.3.2.1. SpaceCube como sistema de procesamiento modular.

SpaceCube es una arquitectura hardware flexible, independiente de procesador, y basada en módulos intercambiables [Bol96]. Esto la convierte en la alternativa ideal para sistemas que usen módulos multichip apilados (Stacked Multichip Modules, MCMs) [Sad97], ya que ha sido diseñado

de tal manera que el acoplamiento de nuevos módulos se realiza de manera sencilla y eficaz. De esta forma, es posible desarrollar arquitecturas más complejas que serían bastante difíciles o imposibles de construir como un único dispositivo o máquina. Además, dichos módulos pueden ser construidos en forma de diferentes polígonos (cuadrados, hexagonales, etc), acorde a las necesidades de los usuarios. Estos módulos, una vez apilados, son fácilmente configurables, resultando un sistema fácil de mantener y altamente expansible.

La primera versión de SpaceCube (SpaceCube 1.0) [NAS06a] (Figura 3.34(a)) estaba basado en las tarjetas FPGA Xilinx Virtex-4 (FX60) [Xil10a] y fue testado en el Espacio como parte del experimento Relative Navigation Sensors (RNS) [Naa09] durante el Hubble Servicing Mission 4 [NAS06b, NAS09a], en Mayo de 2009. El sistema SpaceCube sirvió de interfaz de tres cámaras y ejecutó algoritmos de encuentro y acoplamiento (*rendezvous and docking*) para simular un servicio robótico de captura y liberación de componentes en el Espacio. Estuvo operativo durante 60 horas, cumpliendo todos los objetivos del experimento RNS.

A la hora de desarrollar la primera versión de SpaceCube, se decidió seguir la política MCM pero con dispositivos reconfigurables basados en tarjetas FPGA, siguiendo esta filosofía en las sucesivas versiones que siguieron a este primer prototipo. De esta manera, el sistema está conformado por una secuencia de módulos con diferentes funcionalidades. Estos módulos funcionales reciben el nombre de *slice*, que en su conjunto componen el dispositivo modular o *stack* [God06] (Figura 3.29(a)). Uno de los tipos de módulo más destacables es el módulo de procesamiento, o *processing slice*, que se encarga de albergar las metodologías y donde están situadas las placas FPGA. Cada módulo de procesamiento estaba compuesto, en la primera versión de SpaceCube, por dos tarjetas FPGA Xilinx Virtex-4. Esto potencia enormemente dos funcionalidades. Por un lado, es posible programar cada una de las tarjetas con metodologías diferentes, las cuales estarían en ejecución al mismo tiempo minimizando al máximo el espacio empleado por la electrónica. Igualmente, existe la posibilidad de adaptar y dividir la metodología correspondiente para realizar un procesamiento paralelo, optimizando el rendimiento del sistema. Por otro lado, esta configuración también posibilita la elaboración de diseños tolerantes a fallos. La misma metodología podría ser implementada en cada una de las placas, y mediante la utilización de un dispositivo externo (en este caso un microprocesador RISC) que actuaría como *voter*, se analizarían las salidas proporcionadas por ambas tarjetas FPGA. Si ambas salidas son diferentes, entonces es que se ha producido un fallo (producido probablemente por los altos niveles de radiación) y la operación sería repetida de nuevo. Ésta es una opción muy interesante a la hora de utilizar el sistema SpaceCube para procesamiento en el Espacio, donde los niveles de radiación son el principal problema de la electrónica presente en las plataformas espaciales.

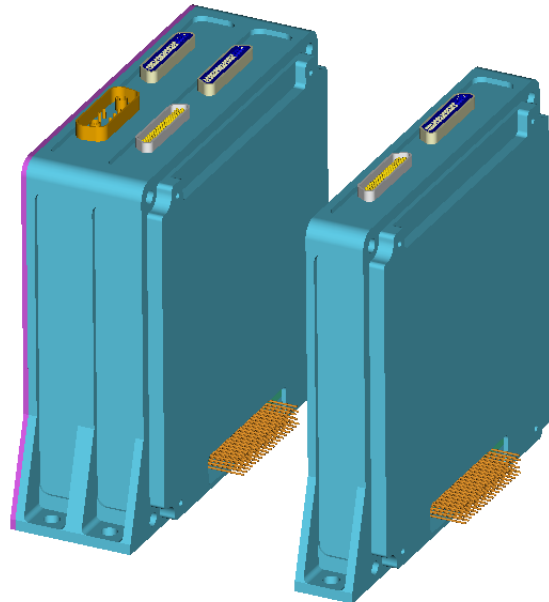
Al tratarse de un sistema modular, sería posible utilizar tantos módulos de procesamiento como requiera la misión, ya que éstos irían acoplados de manera secuencial formando el *stack*. Entre los módulos de procesamiento pueden existir otros tipos de módulos de funcionalidad diversa. De esta manera, es posible encontrar módulos de memoria (*memory slices*), los cuales pueden almacenar los resultados de las metodologías, datos de entrada, información auxiliar, etc.; módulos de enrutamiento (*routing slices*) que posibilitan la correcta transferencia de la información entre los módulos restantes; módulos de entrada/salida (*I/O slices*), que permiten la interacción y transferencia de datos con el exterior del sistema; o los módulos de energía (*power slices*), que suelen estar situados en los dos extremos del *stack* suministrándole energía.

La comunicación entre los diferentes módulos no está limitada a las tarjetas FPGA situadas en un mismo módulo de procesamiento, sino que es posible extrapolarla a los demás módulos presentes en el *stack* gracias al microprocesador RISC, que además de poder actuar como *voter*, también permite el establecimiento de una lógica de comunicaciones compleja (Figura 3.29(b)). De esta manera, la conexión entre módulos puede considerarse como un bus central virtual que lo recorre de extremo a extremo, compuesto por dos tipos de conexiones:

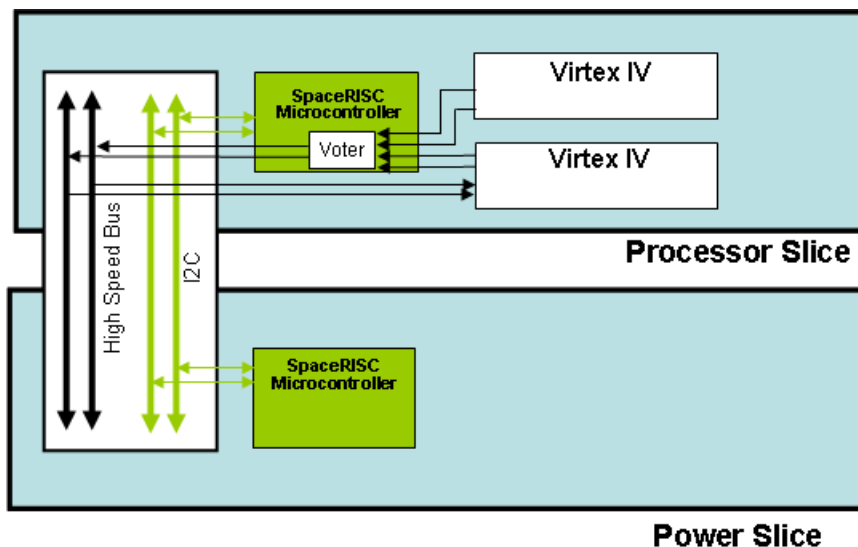
- *Cableado I2C (Conexión de baja velocidad)*: Ofrecen una velocidad de 100 a 400 Kb/s y permiten la comunicación directa entre los microcontroladores de los módulos. Se utilizan para la ejecución

de funciones críticas del sistema.

- *Cableado Ethernet o SpaceWire (Conexión de alta velocidad)*: Ofrecen una velocidad de 125 a 250 Mb/s y permiten la comunicación general de datos entre los módulos.



(a)



(b)

Figura 3.29: Visualización general del sistema *SpaceCube 1.0*.

(a) Diseño del dispositivo *SpaceCube* como sistema modular (*stack*). (b) Comunicación a nivel de módulo y a nivel de *stack*.

Para el caso concreto del *SpaceCube 1.0*, éste estaba formado principalmente por un módulo de procesamiento compuesto por dos tarjetas FPGA Virtex-4 XC4VFX60 (y por tanto, un total de 4 procesadores PowerPC 405) y dos módulos de energía. La Figura 3.30 muestra las características completas para esta versión.

La siguiente versión de *SpaceCube* que voló al Espacio fue la 1.0b (Figura 3.34(b)), como parte del *payload* desarrollado por el Naval Research Lab (NRL) durante la misión Materials on the International Space Station Experiment 7 (MISSE7) [Jen08, Sea09], en Noviembre de 2009.

SPECIFICATIONS

CENTRAL PROCESSORS

450 MHz PowerPC™ 405, 32-bit RISC processors:

- 2 x Xilinx XC4VFx60
- Redundant to handle SEFI
- 32K bytes of secondary (L2) on-die cache

Common Processor Features are:-

- 700+ DMIPS RISC core
- 32-bit Harvard architecture
- 16 KB 2-way set-associative instruction and data caches

Auxiliary Processor Unit (APU) controller

1.2V core voltage

RECONFIGURABLE RESOURCES

2 x 56,880 logic cells

2 x 25,280 slices

2 x 4,176 Kb block RAM

232 18K block RAMs

Example: Helion AES core

- 447 slices, 10 block RAM, 2548Mbps performance

DRAM

3Dplus stacked SDRAM

4 Gbit 75 MHz

Each processor has 1 Gbit dedicated.

SERIAL INTERFACES

32 x LVDS serial pairs:

- Support Ethernet, SpaceWire, or custom interface.
- 18550 compatible UARTs
- Aeroflex LVDS drivers and Receivers
- RS422 can be substituted for LVDS if desired

STACKING CONNECTOR INTERFACE

Airborne Connector:

72 pins

Design uses no backplane or motherboard.

Low speed internal bus – 400Kbps Redundant I2C

High speed bus – TBD Redundant

Power Pins 3.3V, 5V

RAD-HARD SCRUBBER

UT6325 RadHard Eclipse FPGA

320,000 usable system gates

24 dual-port RadHard SRAM modules

RadHard to 300K rad(Si)/sec

OTHER PERIPHERAL INTERFACES

Available through stacking connector by additional card slices

Low Voltage Power Converter card slice

- Same board size as processor
- Provides low voltages from spacecraft bus voltage
- Has 1553 interface, transformers and signal drivers

ETHERNET CAPACITY

8 x Ethernet Media Access Controllers

- IEEE 802.3 compliant
- 10, 100, 1000 Mb/s
- Supports MII, GMII
- Does not use any system gates.

DIGITAL SIGNAL PROCESSING

128 XtremeDSP Slices

18-bit by 18-bit, two's complement multiplier with full precision 36-bit result, sign extended to 48 bits.

FLASH EPROM

256 Mbyte of Flash EPROM

application storage

Flash has separate power switching.

- Allows Flash to be powered off when not in use.

ROM

256 Mbyte of ROM

application storage

backup for Flash

SOFTWARE SUPPORT

Support for Linux, VxWorks

- WindRiver
- MontaVista, BlueCat

GNU GCC Compiler

ELECTRICAL SPECIFICATION

21V to 35V voltage input through optional low voltage power converter card slice

Power Slice can provide:

+5V@ 2 Amps

+3.3V@ 6 Amps

+2.5V@ 4 Amps

all voltages are tolerant to +10% / -10%

SAFETY

SDRAM power is switched separately to handle any potential latchup conditions.

ENVIRONMENTAL SPECIFICATION

-20°C to +55°C (operating Baseplate temperature)

-40°C to +85°C (storage baseplate temperature)

10% to 90% Relative Humidity, non-condensing (storage)

MECHANICAL SPECIFICATION

4 inches x 4 inches (PCB)

Box slice 4.25 inches x 4.25 inches x .75 inches/slice

single board, double sided

I/O connectors:

- 72 pin, Airborne Stacking
- 2 x 37 pin LVDS

Figura 3.30: Especificaciones técnicas del sistema SpaceCube 1.0.

Durante esta misión, el dispositivo SpaceCube fue instalado en la Estación Espacial Internacional (ISS) [NAS07] con el objetivo de testear y validar técnicas software de mitigación de radiación (Radiation Hardened by Software, RHBS [Meh05]). Las técnicas RHBS utilizadas son una combinación de funciones tradicionales de algoritmos de detección y corrección de errores en memorias *Memory Scrubber* [Rev10] y una arquitectura software específica de detección y corrección de errores de ejecución de programas, la cual se ejecuta en segundo plano para encontrar y arreglar alteraciones multi-bit introducidas por la radiación. El objetivo era analizar la tasa de errores producidos por sistemas hardware potentes computacionalmente, pero de bajo coste y de propósito general, como son las tarjetas FPGA ofrecidas por Xilinx. El sistema SpaceCube sigue aún operativo incluso tras la misión Materials on the International Space Station Experiment 8 (MISSE8) [NAS11], realizada en Mayo de 2011. La configuración de la versión 1.0b es exactamente la misma que la de la versión 1.0, pero con la adición de puertos RS-485 [Axe99] para comunicaciones, y una ligera reducción y optimización del consumo energético.

La siguiente versión de SpaceCube fue la 1.5 [Fla10] (Figura 3.34(c)), en Junio de 2011, y fue utilizado como sistema de procesamiento central de un micro-satélite Small Rocket/Spacecraft Technology (SMART) [Esp11, Pet11], una plataforma de propósito general que tenía como objetivo reducir los tiempos de planificación, diseño y desarrollo de las arquitecturas de satélites mediante la utilización de hardware modular reconfigurable. Las metodologías implementadas en el sistema SpaceCube fueron diversas, como por ejemplo la reducción de datos generados a bordo, o el procesamiento de la información y generación de productos en tiempo real (destacando la elaboración de un algoritmo de detección de nubes para la futura misión HypIRI [Gla09]).

Las principales características de esta nueva versión era el paso de las tarjetas FPGA Xilinx Virtex-4 a las Virtex-5 FX100T, y una mayor simplicidad en el diseño. Estaba compuesto por un módulo adicional que hacía las funciones de GPS, un módulo de energía, y un módulo de procesamiento consistente en una única tarjeta FPGA, con sus dos procesadores PowerPC 440 correspondientes (Figura 3.31). Además se añadieron los protocolos SATA II [Del04] y Gigabit Ethernet (GigE) [DAm08] para comunicaciones. Esta versión se ha considerado como intermedia entre la 1.0b y la 2.0.

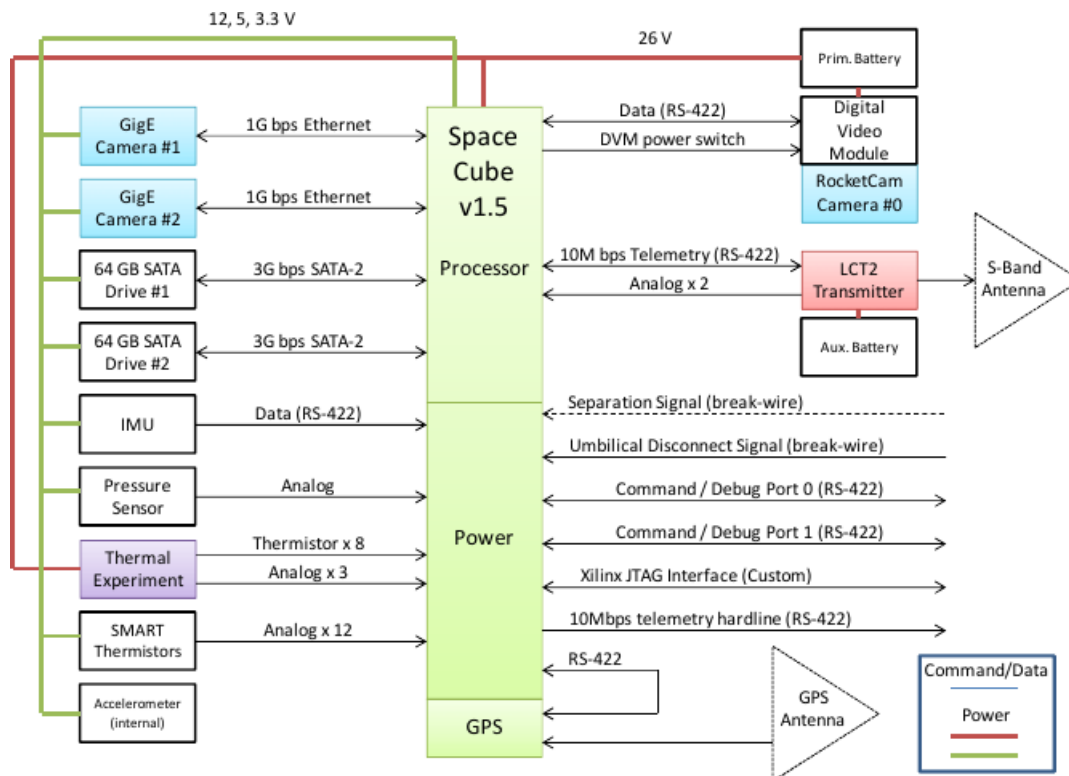


Figura 3.31: Diseño del sistema SpaceCube 1.5.

La versión 2.0 del sistema SpaceCube (Figura 3.34(d)) es la última estable desarrollada hasta la

fecha [Fla12]. Actualmente se está utilizando desde Marzo de 2013 de nuevo en la Estación Espacial Internacional (ISS) como parte de la misión Space Test Program-Houston 4-ISS SpaceCube Experiment 2.0 (STP-H4) [McL12]. Durante esta misión, el dispositivo SpaceCube utilizará diversos instrumentos adjuntos a la plataforma (como un sistema de cámaras, un detector de rayos gamma, un fotómetro, una antena, etc.) con el objetivo de realizar un estudio más detallado sobre las metodologías RHBS, testear algoritmos de detección de eventos sobre la corteza terrestre, y estudiar las eyecciones de rayos gamma procedentes de las tormentas en nuestro planeta.

En esta versión, el sistema SpaceCube está compuesto por un módulo de energía, un módulo de memoria FLASH, un módulo de entrada/salida y un módulo de procesamiento (Figura 3.32(a)). El módulo de procesamiento ha sido diseñado de tal manera que puede albergar o bien dos tarjetas FPGA Xilinx Virtex-5 ML510 V5FX130T (proporcionando un total de 4 procesadores PowerPC 440), o bien una única tarjeta Xilinx Virtex-5QV Single-event Immune Reconfigurable (SIRF) [Xil12b], un modelo especial con alta tolerancia a fallos (Figura 3.32(b)). A pesar de que durante la misión STP-H4 se están utilizando las dos tarjetas FPGA Xilinx Virtex-5 ML510 V5FX130T, el hecho de adaptar el módulo de procesamiento para la adición de otro tipo de tarjetas permite la posibilidad de experimentar nuevas posibilidades en futuras versiones. Respecto a los protocolos utilizados en la versión 2.0 del sistema SpaceCube, se añadieron CompactPCI (cPCI) [PIC99] y SpaceWire [Par99] a los ya presentes en versiones anteriores, tanto para establecer comunicaciones entre los módulos del *stack*, como con el exterior.

Futuras versiones de SpaceCube tienen como máximo exponente a SpaceCube 2.0 Mini (SpaceCube 2M) [NAS12] (Figura 3.34(e)), una versión miniaturizada de dicho dispositivo y que será testada como *payload* del microsatélite Intelligent Payload Experiment (IPEX) CubeSat [Sta12, Chi12], demostrando las capacidades del procesamiento a bordo de metodologías avanzadas para después ser incorporadas en futuras misiones como HypsIRI. La tecnología varía bastante a la versión análoga 2.0, ya que todo el sistema está compuesto por un cubo en el que cada lado implementa un módulo. De esta forma, es posible encontrar un módulo de energía, y dos módulos de procesamiento diferentes. El primero de ellos responde al prototipo de módulo de procesamiento presentado en la versión 2.0 original, pero utilizando la tarjeta Xilinx Virtex-5QV SIRF anteriormente descrita (no incluye procesadores embebidos). El segundo tipo de módulo de procesamiento lo conforma una única tarjeta FPGA Aeroflex UT6325 [Aer12], la cual presenta una alta tolerancia a fallos y soporta altos niveles de radiación (Figura 3.33).

Aunque se ha realizado un detallado resumen de todas las versiones del sistema SpaceCube, el resto de la documentación de tesis se centrará en la versión actual desarrollada y testada, es decir, la versión SpaceCube 2.0 con módulo de procesamiento consistente en dos tarjetas FPGA Virtex-5 interconectadas.

Los siguientes apartados detallan la arquitectura Virtex-5 que compone principalmente el tipo de dispositivo FPGA presente en el sistema SpaceCube 2.0 (Sección 3.3.2.1.1), así como las herramientas que permiten el diseño y desarrollo de metodologías en este tipo de plataforma reconfigurable a nivel de hardware (Sección 3.3.2.1.2).

3.3.2.1.1. La arquitectura Virtex-5.

La familia de FPGAs Virtex-5 ofrece a los diseñadores plataformas múltiples con un optimizado balance de lógica de alto rendimiento, conectividad serie, procesamiento de señales, y recursos de procesamiento embebidos [Xil09b]. Todos los miembros de la familia Virtex-5 son construidos usando la segunda generación de la tecnología Advanced Silicon Modular Block (ASMBL) [Fie10] la cual produce las FPGAs de mayor rendimiento de la industria. Aparte de la adición de un bloque de doble procesador PowerPC 440 embebido, la familia FPGA Virtex-5 también proporciona componentes como un sistema integrado PCI Express, conexión Ethernet de tres velocidades o transeptores serie

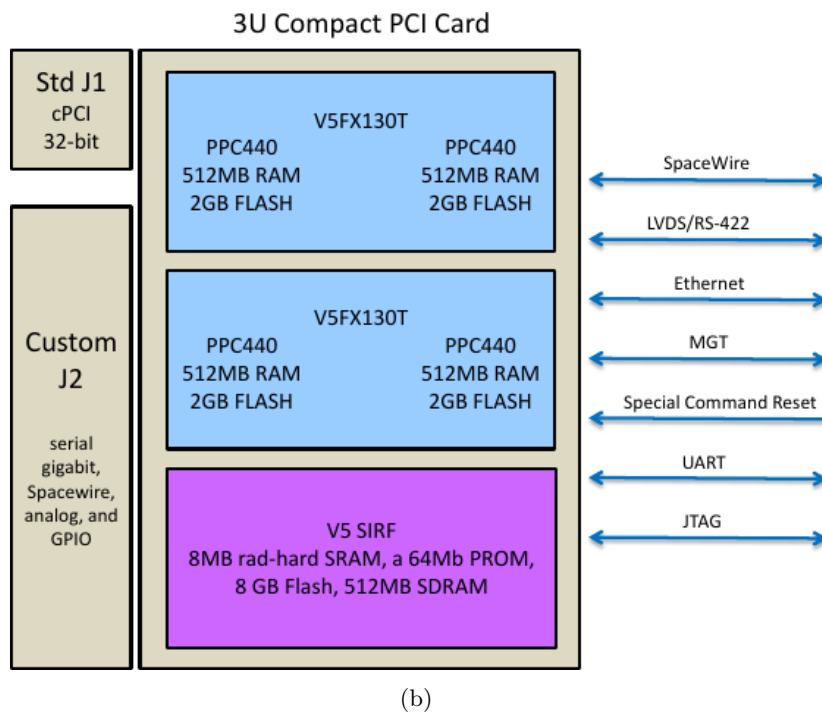
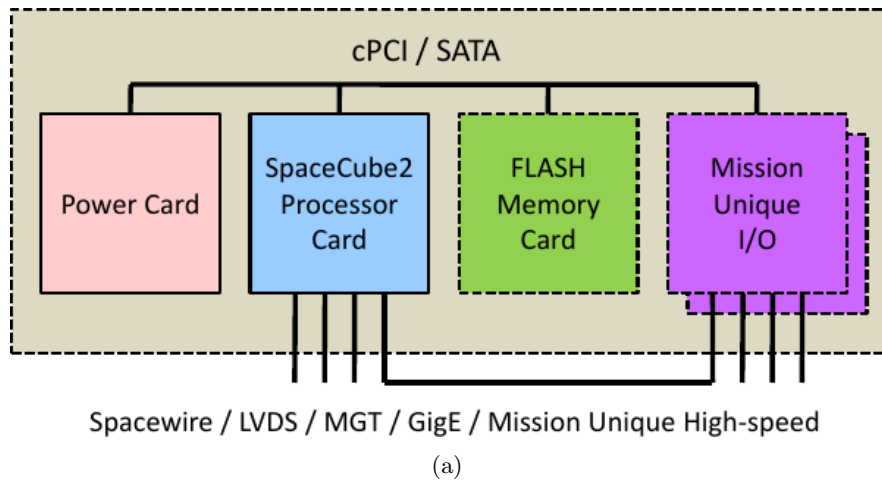


Figura 3.32: Visualización general del sistema SpaceCube 2.0.
 (a) Diseño como sistema modular (stack). (b) Módulo de procesamiento.

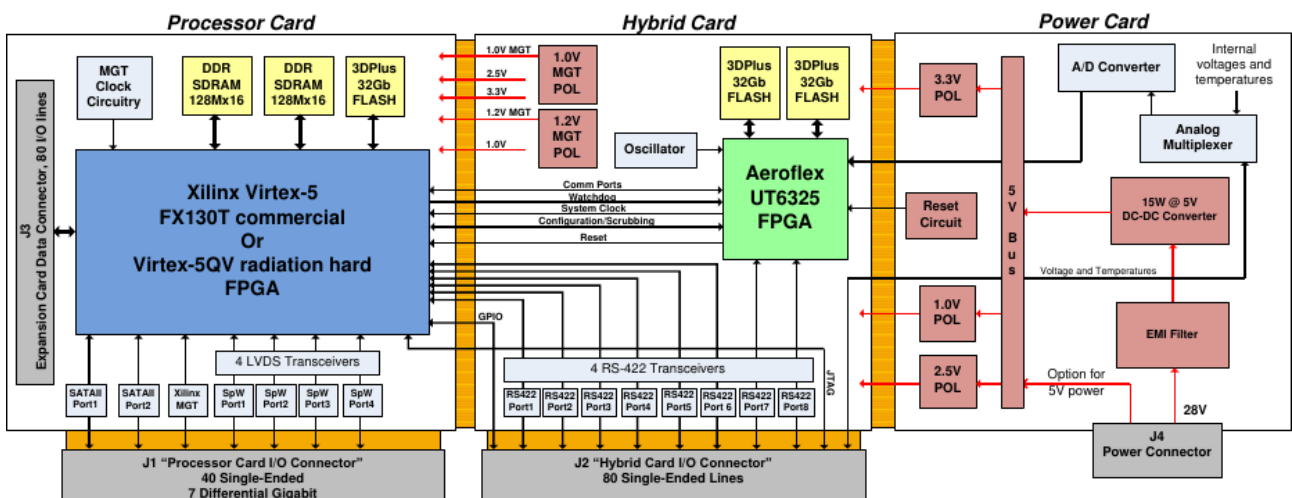
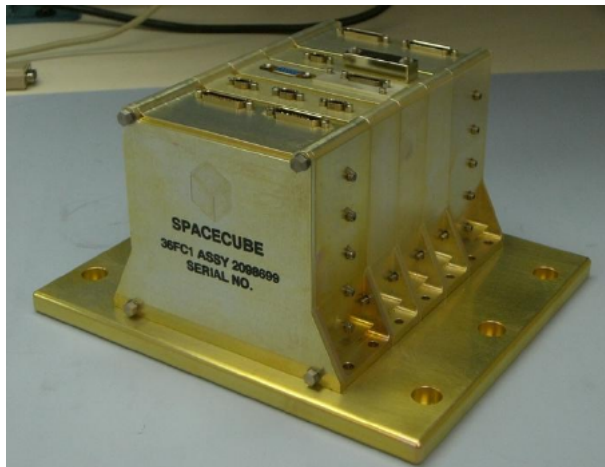
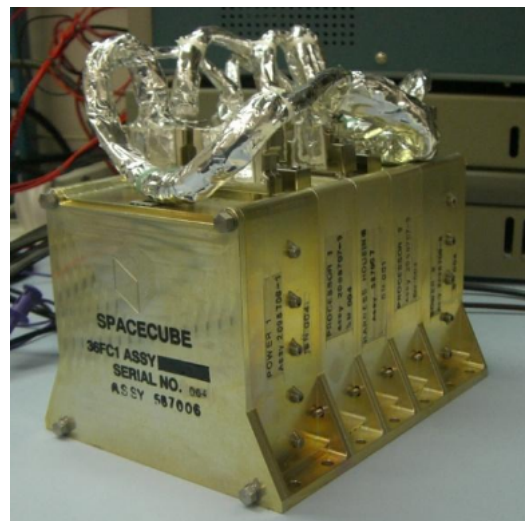


Figura 3.33: Diseño del sistema SpaceCube 2.0 Mini.



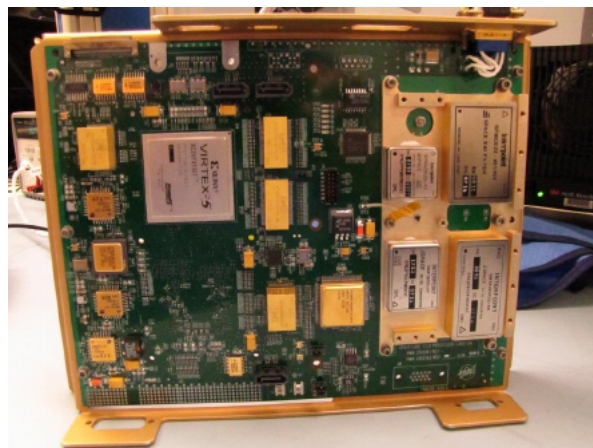
(a)



(b)



(c)



(d)



(e)

Figura 3.34: Versiones del sistema SpaceCube.

(a) SpaceCube 1.0. (b) SpaceCube 1.0b. (c) SpaceCube 1.5. (d) SpaceCube 2.0 (Prototipo). (e) SpaceCube 2.0 Mini (Diseño).

RocketIO GTP y GTX de alta velocidad.

Esta familia proporciona una serie de dispositivos FPGA con similares características, las cuales pueden variar adaptándose a las diferentes necesidades de las metodologías que se desean implementar a un nivel hardware [Xil09c]. De esta manera, es posible distinguir entre modelos de tarjeta FPGA Virtex-5 de gama baja, gama media y gama alta. Las más características, y las que han sido usadas por el grupo SpaceCube, son los modelos Virtex-5 ML507 (XC5VFX70T) [Xil11c] y Virtex-5 ML510 (XC5VFX130T) [Xil11d]. Ésta última es una de las versiones más completas de los modelos proporcionados por Xilinx dentro de la familia Virtex-5, la cual destaca por presentar un chip FPGA de mayor tamaño, dos procesadores PowerPC embebidos, un mayor número de conectores para los protocolos de comunicaciones implementados, y conectores de reconfiguración adaptable más avanzados. Es precisamente este modelo el sistema definitivo utilizado para la composición del dispositivo SpaceCube 2.0, y el cual es adaptado para ser ejecutado en el Espacio. El otro modelo, ML507, es una versión más limitada, con un chip FPGA ligeramente de menor tamaño, un único procesador PowerPC embebido, un único conector para la mayoría de los protocolos de comunicaciones implementados, y conectores de reconfiguración más simples y lentos. Este modelo, debido a su inferior coste y su completa compatibilidad con el modelo ML510, es utilizado como *plataforma de testeo*, en el cual son previamente diseñadas, implementadas y probadas las metodologías que serán ejecutadas a bordo del satélite. Por tanto, en una primera instancia, todos los algoritmos son desarrollados en el modelo ML507 para después ser implementados de manera definitiva en el modelo ML510.

La Figura 3.35 muestra el diagrama de bloques de la plataforma Virtex-5, mientras que la Figura 3.36 ofrece una vista detallada de los componentes más importantes de los modelos de tarjeta FPGA Virtex-5 ML507 y ML510.

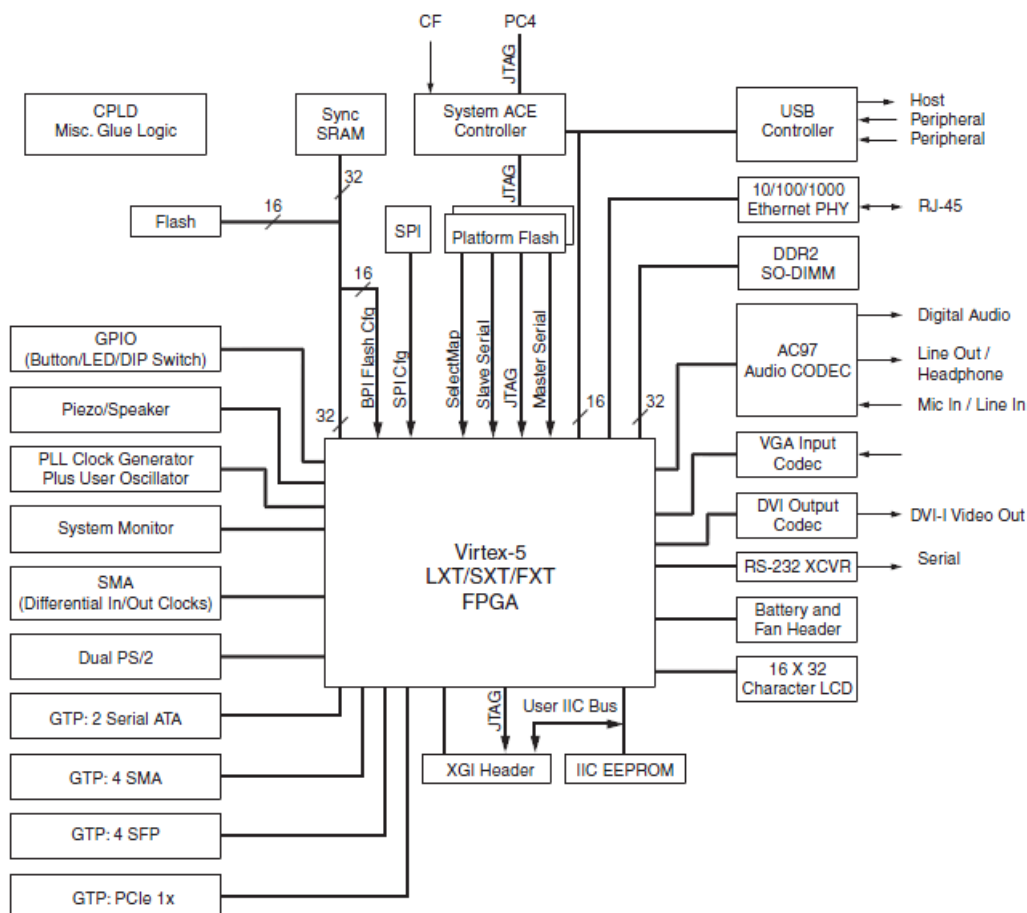
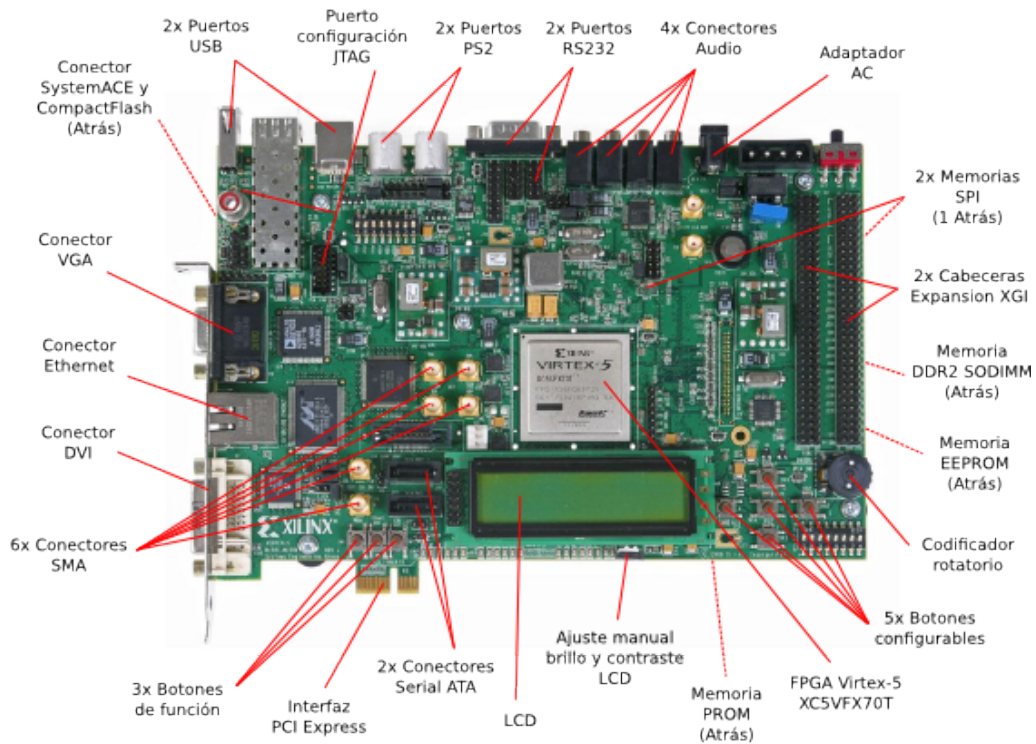


Figura 3.35: Diagrama de bloques de la plataforma Virtex-5.

ML507



ML510

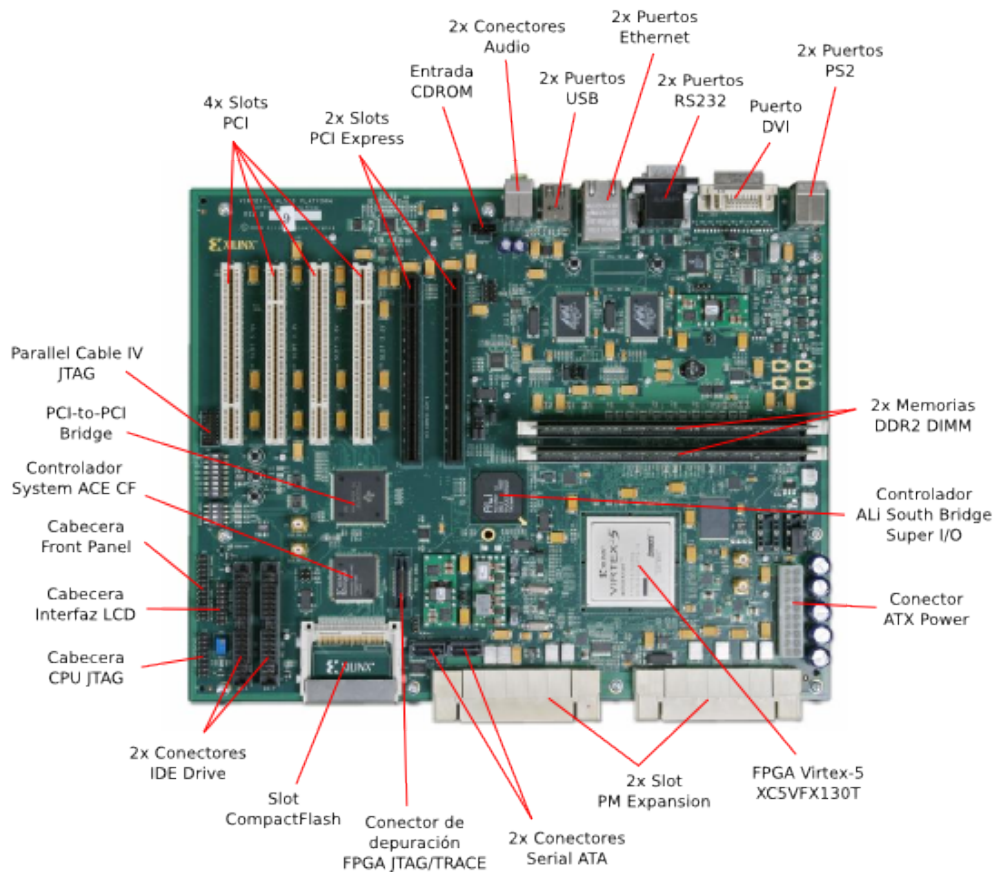


Figura 3.36: Vista detallada de los componentes presentes en las tarjetas FPGA Xilinx Virtex-5 ML507 y ML510.

3.3.2.1.2. El entorno de desarrollo de Xilinx.

Con el objetivo de poder diseñar e implementar los algoritmos hardware/software en la tarjeta de desarrollo, Xilinx ofrece un conjunto de herramientas acordes a las necesidades del usuario, las cuales se encuentran recopiladas en el entorno Xilinx ISE Design Suite. Las más importantes son:

- *Chipscope Pro*: Esta herramienta ofrece núcleos software que actúan como analizadores lógicos, analizadores de sistema y monitores virtuales de entrada/salida. Permite visualizar el estado de cualquier señal o nodo, incluyendo los de los procesadores embebidos. Dichas señales pueden capturarse en tiempo de ejecución y mostrarse en pantalla.
- *Xilinx Platform Studio (EDK)*: Esta herramienta permite desarrollar un sistema completo de procesador embebido en un dispositivo Xilinx FPGA mediante el uso de componentes periféricos de Propiedad Intelectual (Intellectual Property, IP).
- *Project Navigator (ISE)*: Esta herramienta organiza a alto nivel los ficheros fuente y los procesos de ejecución de los diseños de la FPGA y CPLD desarrollados por el usuario, permitiendo de esta manera la adición, creación y modificación de ficheros fuente y ejecución y monitorización de los procesos correspondientes.
- *PlanAhead*: Esta herramienta permite a los desarrolladores implementar sus diseños directamente en la tarjeta a un nivel físico. Esto se consigue mediante la utilización de una metodología de diseño jerárquica para minimizar la congestión de enrutamiento, simplificar la complejidad de interconexión y asignación de reloj, y explorar las opciones de implementación.
- *System Generator*: Esta herramienta integra el entorno de desarrollo The Mathworks Simulink [[Simwww](#)] para el diseño de FPGAs, y añade todas las funcionalidades y pasos necesarios para la transferencia de las implementaciones a la FPGA de manera automática, como la síntesis, posicionamiento y enrutamiento, mediante la generación de un fichero de programación FPGA.

El siguiente capítulo realiza una introducción a alternativas a los formatos de datos convencionales basados en la medición como unidad básica geolocalizada, lo que permite la minimización de los errores espaciales durante el proceso de geocorrección (entre otras funcionalidades añadidas), y la posibilidad de ser fácilmente adaptable a las diferentes requisitos y necesidades de las misiones. Además, detalla el diseño de un sistema de computación reconfigurable a nivel de hardware y la integración de dichos formatos de datos, con el objetivo de optimizar el procesamiento a bordo en tiempo real.

Capítulo 4

Metodologías

Tal y como se ha mostrado en el Capítulo 3, es importante estudiar la manera en la que se almacena y provee la información recogida por los sensores a bordo de aviones y satélites, ya que una mala gestión de los datos puede generar unos errores de posicionamiento de las mediciones adquiridas. Este hecho implica que ciertas metodologías, como la clasificación espectral o el análisis subpíxel, no sean tan precisas como deberían.

En la adquisición de datos de un terreno o escena, existen dos parámetros muy importantes que influyen de manera directa sobre los resultados del estudio científico que se está realizando. El primero de ellos es el valor de las mediciones en sí, es decir, los valores adquiridos por un instrumento y que conformarán los datos sobre los que se realizarán los experimentos y con los que trabajarán las metodologías, como podrían ser firmas espectrales. Para una correcta adquisición, es necesario disponer de una instrumentación fiable y de calidad. El segundo parámetro es la georreferenciación de dichos valores, es decir, la localización, dentro de la escena o terreno, en donde los datos han sido obtenidos. Una mala georreferenciación puede implicar la asignación de los valores de adquisición a elementos o posiciones del terreno irreales, que pueden dar como resultado la imprecisión de las metodologías utilizadas en los experimentos.

Por tanto, es importante el diseño de estructuras de datos que gestionen, de la manera más óptima posible, tanto los valores resultantes de las mediciones realizadas, como la localización de los mismos. Los formatos de datos convencionales prestan especial atención al primer parámetro, mientras que el segundo se abarca de manera superficial y casi siempre orientado a la correcta representación visual de la escena escaneada.

En este apartado, se proponen diferentes estructuras de datos alternativas a las convencionales, considerando de esta manera que tanto los datos de la medición, así como su localización, son igual de importantes y que el correcto almacenamiento y procesamiento de los mismos han de realizarse de manera optimizada.

En un principio, para afrontar este problema de manera sencilla, es posible considerar una campaña de campo en la que un experto va adquiriendo mediciones de diferentes elementos de interés sobre una escena o terreno cualquiera (Figura 4.1(a)). Para una correcta adquisición, deben almacenarse tanto los valores obtenidos con el instrumento de medición (que en el caso de un espectrómetro se trataría de firmas espectrales, aunque puede ser otro tipo de dato, dependiendo del instrumento utilizado), así como la posición en la cual se adquirió cada medición, y que puede ser suministrado por un dispositivo GPS. Además, hay que considerar que podría ser posible tener que almacenar información adicional, como la identificación del elemento medido, la fecha de adquisición, o el nombre del experto que está realizando la medición. Siguiendo este esquema, el experto se iría desplazando por la zona de interés obteniendo mediciones, representados como valores de información, sobre elementos situados en una posición espacial. El resultado es el escaneo manual *in-situ* de

un terreno para la adquisición de datos con una distribución espacial irregular, implicando un almacenamiento y un procesamiento ineficientes por parte de las metodologías desarrolladas en los experimentos, sobre todo si se utilizan estructuras de datos matriciales para la representación de dicha información.

Este mismo problema puede ser extrapolado a la adquisición de mediciones desde aviones o satélites (Figura 4.1(b)). Un instrumento situado en una plataforma obtiene los datos de una escena o terreno de interés, y, conociendo las características de la óptica y la geometría del sensor utilizado, son georreferenciados mediante la sincronización del instrumento con el dispositivo de posicionamiento. En estos casos, el resultado de la adquisición es el mismo, una secuencia de mediciones que se distribuye espacialmente de manera irregular, y que hay que almacenar de manera óptima para su correcto y preciso procesamiento.

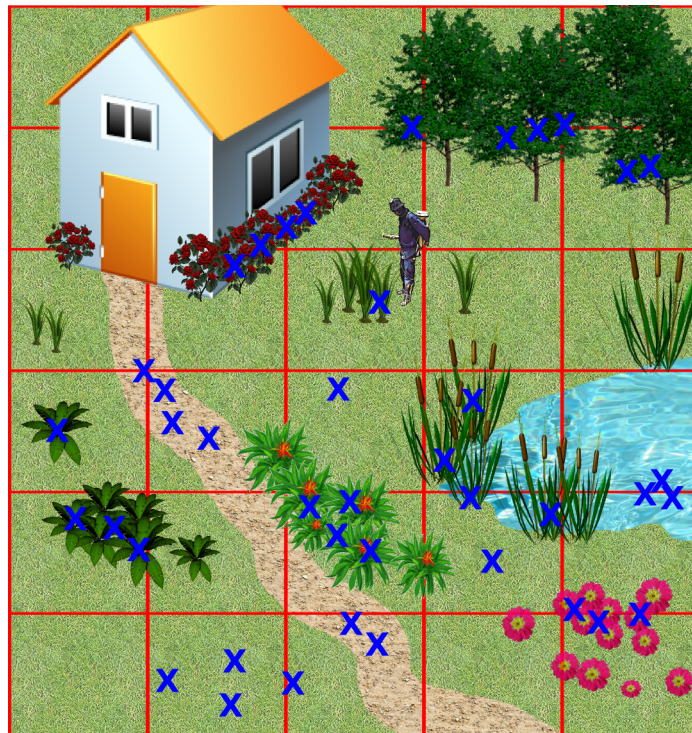
Una vez obtenidos los datos de interés y su posicionamiento dentro de la una escena escaneada bien en campañas de campo (adquiridos por el experto), o bien de manera remota (obtenidos por aviones o satélites), es posible considerar pequeñas zonas o subáreas en las que las densidades de mediciones difieren. Existirán zonas que, por motivos accidentales (turbulencias sufridas por el avión, o geometría de los sensores) o intencionados (dirección del vuelo, trayectoria del experto) contendrán un número superior de adquisiciones respecto a otras áreas, en las que podría no haber ninguna. De esta manera, es difícil realizar una división matricial de la escena, considerando pequeñas áreas cuadradas como píxeles, en las que almacenar un único valor de medición (por ejemplo, un espectro completo), tal y como consideran los formatos de datos convencionales matriciales. En este marco, es complicado establecer un tamaño de píxel para construir la imagen geocorregida correspondiente. Un tamaño de píxel demasiado grande da lugar a una imagen final de baja resolución, en la que un gran porcentaje de las mediciones se pierden, al almacenar únicamente un valor de medición por píxel. Si se quiere conservar el mayor número de mediciones posibles en la imagen geocorregida correspondiente, sería necesario considerar un tamaño de píxel mínimo, lo que resultaría una imagen final de alta resolución, de gran tamaño, y conteniendo demasiada información *vacía* o *nula* de zonas en las que no se han realizado adquisiciones (Figura 4.2).

Esta dificultad a la hora de seleccionar un tamaño de píxel adecuado, hay que sumar las imprecisiones introducidas por los procedimientos de geocorrección convencionales (presentados en la Sección 3.2.5), más orientados a generar una imagen final optimizada a análisis visuales que automatizados.

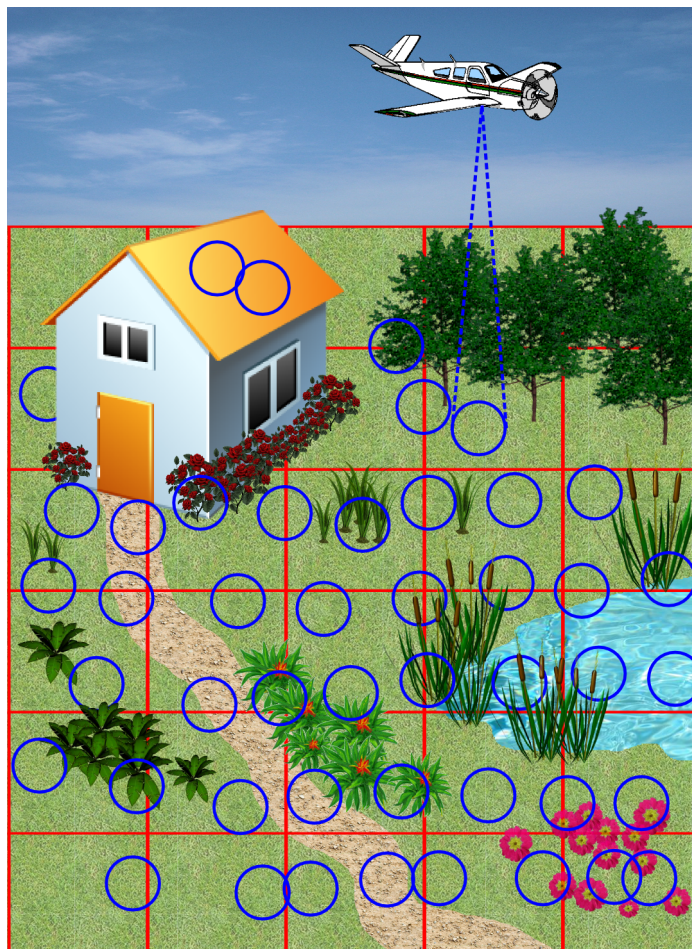
De esta forma, y considerando todos los problemas comentados con anterioridad, a la hora de diseñar una estructura de datos que permita almacenar la adquisición de mediciones con tan específicas características, en un primer planteamiento se establecen una serie de pautas y posibilidades que busquen la óptima gestión de la captura de la información sobre una escena:

- Considerar áreas (píxeles) en las que puede no haber información.
- Considerar áreas (píxeles) en las que puede haber uno o más datos de medición.
- Particionar de manera regular la escena mediante subáreas (píxeles) cuadradas que posibiliten un procesamiento sencillo y rápido, así como un acceso directo a áreas de interés. Sin embargo, podría ser eficiente considerar divisiones irregulares, de acuerdo a la densidad de mediciones (Figura 4.3). De esta manera, no sólo existiría la posibilidad de utilizar un diseño matricial, sino basado en árboles.
- Georreferenciar cada dato de medición por su verdadera localización espacial (proporcionada por el sistema de posicionamiento), en vez de utilizar posiciones relativas a un píxel referencia de la imagen (por ejemplo, las esquinas) y el tamaño de píxel.

En este marco, y teniendo en cuenta estas consideraciones, se propone un formato alternativo a los sistemas de almacenamiento convencionales: *Diffused Matrix Format (DMF)*, una estructura



(a)



(b)

Figura 4.1: Adquisición de mediciones sobre una determinada escena. (a) Utilizando un instrumento manual (campanas de campo). (b) Utilizando un instrumento sobre una plataforma a bordo de un avión / satélite.

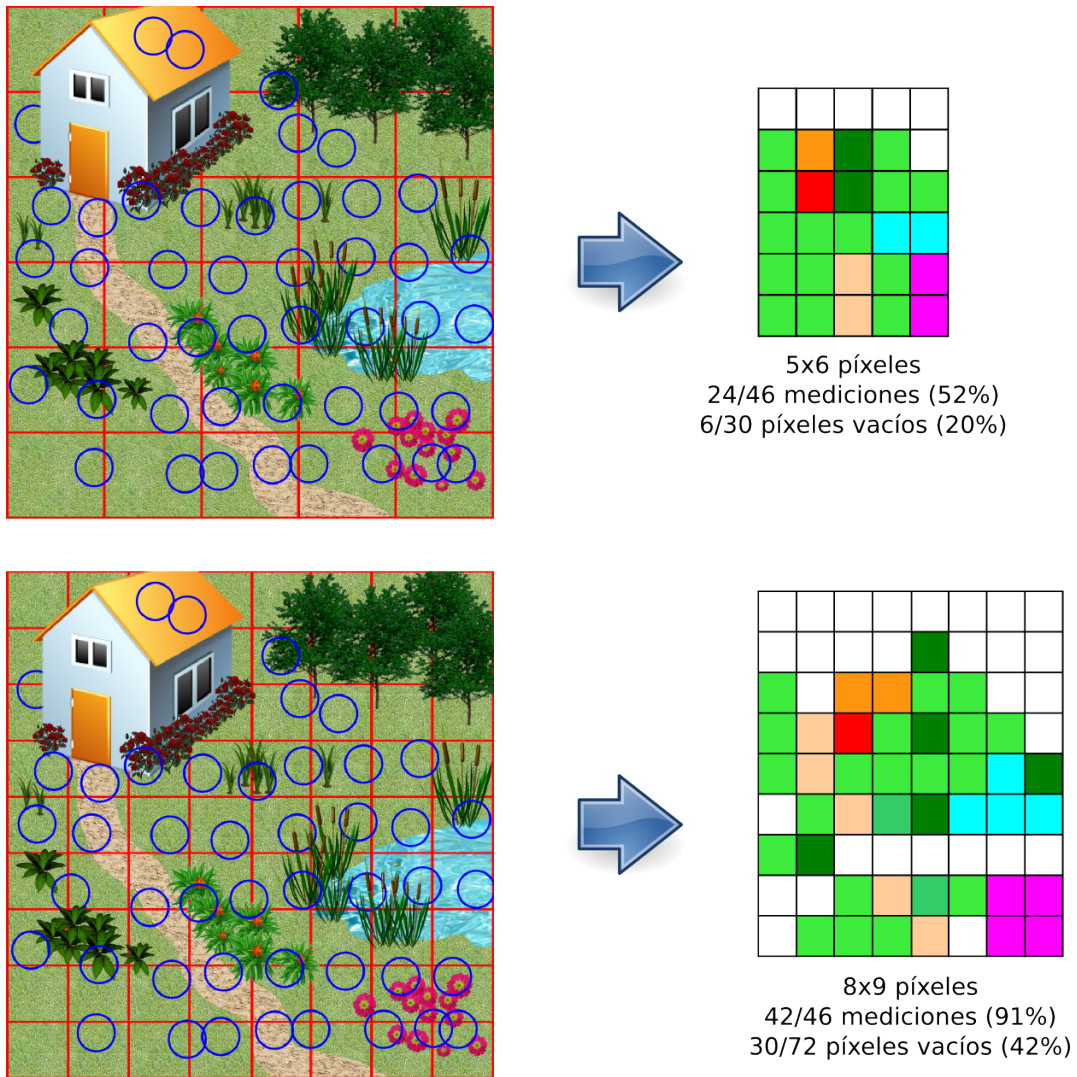


Figura 4.2: Representación de una misma escena utilizando diferentes tamaños de píxel.

pseudo-matricial que permite almacenar en cada celda (lo que correspondería a un píxel en una imagen convencional) ninguna, una o varias mediciones. De esta manera, la unidad en torno a la cual se basa el formato no sería la imagen, como en formatos convencionales, sino la medición, la cual estará compuesta por la información obtenida por el instrumento (por ejemplo, un espectro completo), y las coordenadas espaciales donde fue adquirida (latitud y longitud), así como datos adicionales de acuerdo a las necesidades de la misión. Utilizando este esquema, los datos siempre estarán georreferenciados con una precisión limitada a la del sistema de posicionamiento, y los desplazamientos de las mediciones, así como su repetición o eliminación, desaparecerán independientemente del tamaño de celda seleccionado.

Este formato, aparte de presentarse como una nueva opción a los formatos de almacenamiento y procesamiento de imágenes convencionales, supone una gran alternativa en sistemas de procesamiento a bordo (en las cuales es difícil disponer de los datos geocorregidos) ya que se potencia la medición como unidad elemental. El hecho de presentar un tipo de información en el que se presentan a la vez valores de medición junto con coordenadas espaciales permite la potenciación de metodologías orientadas a la monitorización de la corteza terrestre (detección y monitorización de incendios, detección de nubes, evaluación de catástrofes, etc.). Además, los resultados de dichas metodologías, así como toda la información recogida por los instrumentos y sensores disponibles, se encontrarán directamente georreferenciados (nada más terminar la adquisición) y listos para procesar en formato DMF, pudiendo aligerar en ocasiones el largo tiempo de espera hasta la distribución de los datos a la comunidad científica.

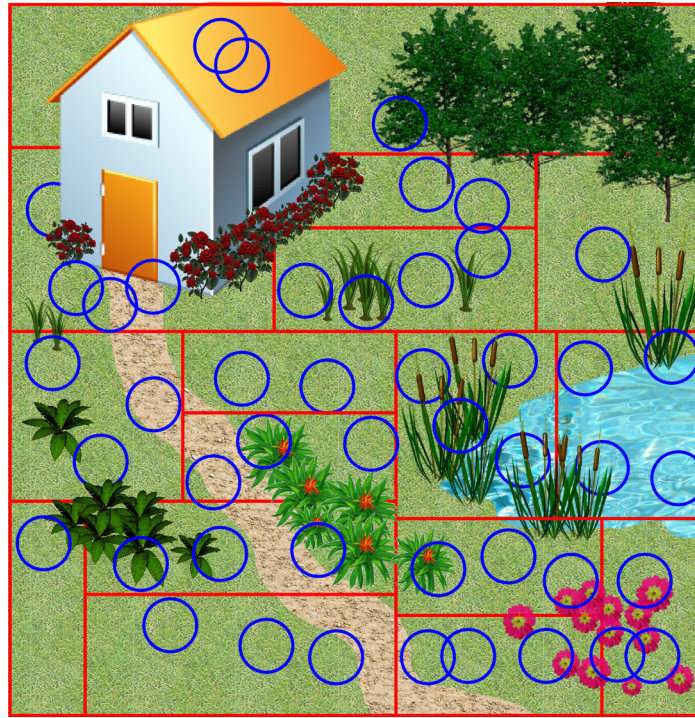


Figura 4.3: *Particionamiento de una determinada escena de manera irregular.*

Sin embargo, es necesario adaptar todos los sistemas hardware y software a bordo de los aviones y satélite para que el formato DMF sea totalmente funcional y compatible. Es por esto por lo que los sistemas re-configurables suponen una gran oportunidad para el desarrollo de este formato.

Uno de los ejemplos que ofrecen más posibilidades es el sistema SpaceCube, diseñado y desarrollado por el centro Goddard Space Flight Center de la NASA (Sección 3.3.2.1). Está basado en la familia de tarjetas comerciales Field Programmable Gate Array (FPGA) Virtex-5, las cuales permiten la posibilidad de re-programar continuamente la circuitería que implementará las metodologías a ser procesadas a bordo del avión o satélite. Aparte del chip FPGA, la tarjeta también proporciona uno o dos procesadores PowerPC embebidos, así como numerosos puertos de entrada / salida para realizar comunicaciones con el exterior utilizando diferentes protocolos. Estas características hacen del sistema SpaceCube una gran alternativa para la implementación hardware de metodologías procesadas a bordo, y su combinación con el formato DMF es potencial para el desarrollo de la tecnología y la ciencia en este ámbito.

El resto del capítulo está organizado de la siguiente manera: En primer lugar, se realiza una presentación detallada del formato DMF (Sección 4.1), el cual abarca las diferentes posibilidades de implementación, sus características (creación, tamaño en RAM y disco, visualización) y su adaptación a operaciones básicas del procesamiento de imágenes (umbralizado y dilatación) y a metodologías más complejas (fusión de datos y detección de cambios). En segundo lugar, se propone la emulación de un sistema SpaceCube real mediante la conexión de dos tarjetas Xilinx Virtex-5 convencionales (Sección 4.2), en las cuales se ha desarrollado una arquitectura hardware a nivel general que permite albergar cualquier tipo de metodología. Esta arquitectura trabajará con el formato DMF para la comunicación entre ambas, así como para la salida del resultado correspondiente.

4.1. Diffused Matrix Format (DMF).

Con el objetivo de solucionar los problemas derivados de la imprecisión espacial y de minimizar los errores cometidos al georreferenciar la información obtenida por los sensores, una estructura de datos que represente una escena debe afrontar principalmente la circunstancia de que las posiciones de

las mediciones no siguen un patrón regular, estando distribuidas de manera aleatoria debido a factores introducidos por agentes externos, como la geometría de los sensores o el movimiento de la plataforma.

Esta distribución irregular de las mediciones dificultan el particionado de la imagen en patrones cuadrados regulares (píxeles) con el propósito de ser almacenados y procesados con un cierto orden. De esta manera, existen problemas cuando dos o más mediciones son asignados a un mismo píxel, o bien a un píxel no se le asigna ninguna medición. Los procesos convencionales de geocorrección abordan el primer problema o bien mediante la asignación de una medición representativa (generalmente la más cercana al centro del píxel), o bien mediante el promediado de los valores de las mediciones involucradas (e introduciendo modificaciones artificiales en la información geocorregida final). Para el segundo problema, los píxeles *vacíos* pueden rellenarse mediante la asignación de una medición por vecino más próximo (introduciendo duplicidad de la información), o, si el píxel cae fuera de la escena de estudio, asignar una medición por defecto (cuyos valores suelen ser 0). Ambas circunstancias se denotan mediante el uso de máscaras adicionales que implican un consumo de memoria adicional, y que indican si un píxel ha quedado sin asignar, o bien ha sido asignado por vecino más próximo. Un ejemplo de este tipo de máscaras es el fichero GLT (Sección 3.2.6.1) que acompaña a la imagen geocorregida en numerosos casos.

El empleo de una máscara adicional puede ser un recurso útil para detectar el origen de cada píxel de la imagen geocorregida, pudiendo diferenciar entre los píxeles a los que les fue asignado una medición de manera directa, de aquéllos a los que les fue asignado por vecino más próximo o bien quedaron sin asignar. Sin embargo, un formato de datos que utilice una máscara para este propósito no supone una mejora a la hora de georeferenciar de manera precisa los datos adquiridos, ni de incrementar la información que una medición puede aportar al considerar la imagen como unidad elemental básica.

Por tanto, el diseño de un formato de datos que sí considere las anteriores circunstancias debe partir de las siguientes premisas:

- *Conservación de la información instrumental y espacial:* La estructura de datos propuesta debe almacenar la información recogida por los sensores tal y como fue adquirida, sin descartar mediciones no asignadas a píxeles, sin realizar cálculos estadísticos y promediados de los valores de mediciones próximas, y evitando el desplazamiento de la medición por asignación al centro del píxel respecto a su posición original.
- *Minimización del tamaño de la imagen:* El almacenamiento de la estructura de datos propuesta no debe suponer un tamaño en disco o en memoria muy superior al correspondiente a la información instrumental y espacial de las mediciones adquiridas. Se debe minimizar el uso de variables auxiliares. Una minimización en el tamaño de la imagen implica un manejo más fluido de los datos, así como un procesamiento más rápido.
- *Eliminación del duplicado de información:* La estructura de datos propuesta debe almacenar la información recogida por los sensores sin duplicarla. Se deben evitar casos como por ejemplo la asignación de una misma medición a diferentes píxeles cercanos al ser seleccionado como vecino más próximo.

De esta manera, y partiendo de las anteriores premisas, ha de considerarse la imagen final como la representación de una escena sobre la que se han adquirido una serie de mediciones de manera irregular. Dicha escena puede ser dividida en varias áreas contenedoras de mediciones, que en un principio, pueden ser regulares y cuadradas (como los píxeles de las imágenes geocorregidas convencionales). Sin embargo, cada una de estas áreas pueden almacenar ninguna, una o varias mediciones.

En este ámbito se propone el formato y estructura de datos pseudomatricial Diffused Matrix Format (DMF) [Cha07b, Mar10, Her11], la cual parte de las estructuras matriciales convencionales, pero añade una serie de elementos y posibilidades que permite evitar o minimizar los errores y

problemas que aquéllas ofrecen. Tal y como se ha propuesto con anterioridad, la unidad elemental es considerada la medición, por lo que está basada en un elemento básico denominado Diffused Matrix Record (DMR), un registro que almacena toda la información correspondiente a una medición obtenida por el sensor (Figura 4.4(a)). Esto es, la posición espacial de la medición, proporcionada por el dispositivo de posicionamiento, junto con los datos ofrecidos por el resto de instrumentos.

Estos registros DMR, son posicionados sobre una cuadrícula denominada matriz DMF (Figura 4.4(b)), la cual representa la escena particionada de manera regular, de acuerdo a sus respectivas localizaciones espaciales. Esta matriz consiste en una rejilla espacial regular indexada en dirección N/S respecto a las coordenadas utilizadas por el dispositivo de posicionamiento, y en la cual el tamaño de celda elemental puede ser variable.

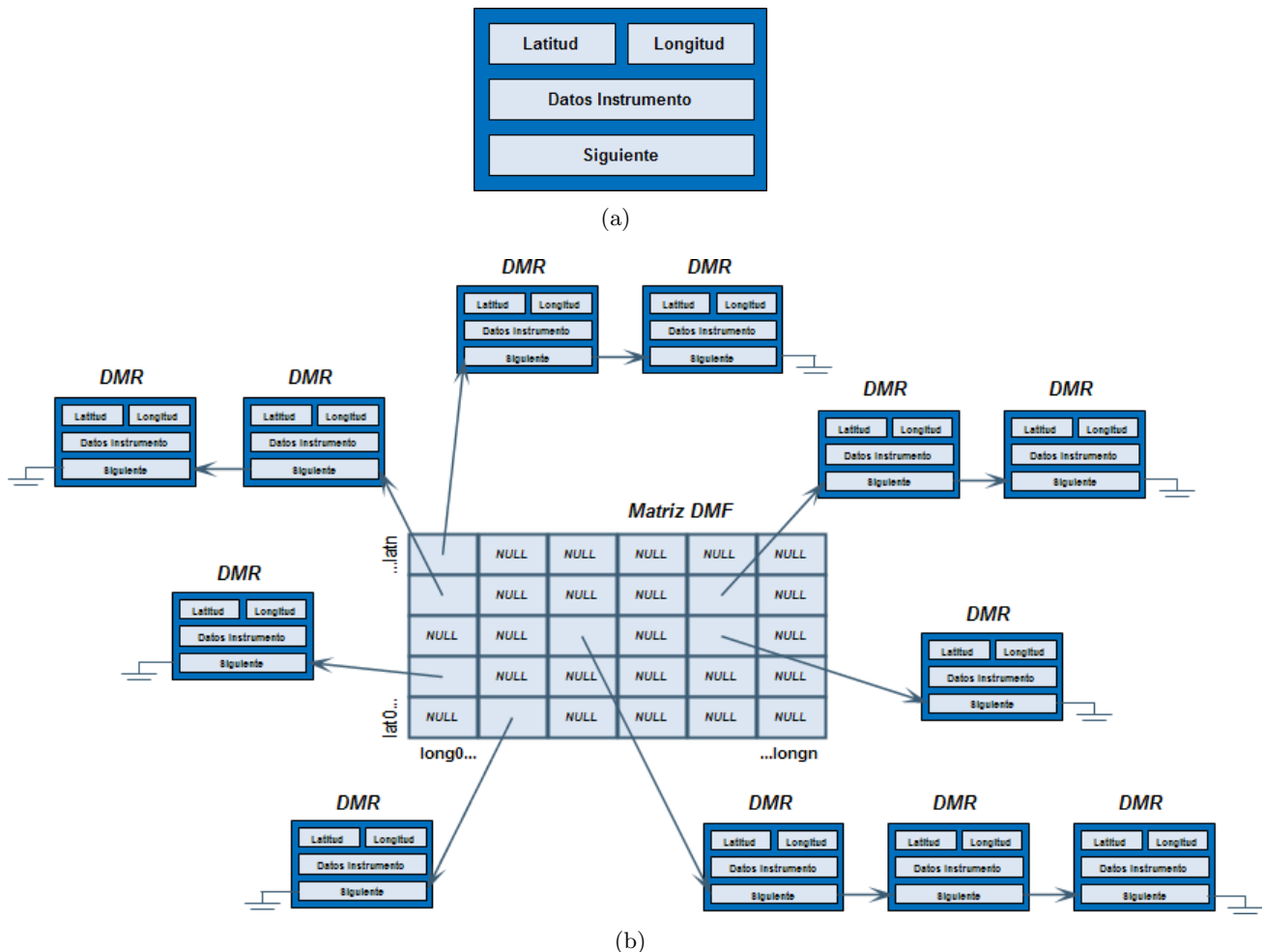


Figura 4.4: Estructuras básicas del formato DMF.

(a) Registro DMR. (b) Matriz DMF.

Mediante esta metodología, a cada celda DMF se le asigna aquéllas mediciones cuyas coordenadas corresponden al área de la escena que dicha celda representa. De esta manera, podemos encontrar tres circunstancias diferentes:

- A una celda DMF se le asigna un único registro DMR: La celda DMF almacenará dicho registro tal cual.
- A una celda DMF no se le asigna ningún registro DMR: Esto ocurre cuando no se ha adquirido ninguna medición justo en el área de la escena que representa la celda DMF. En este caso, dicha celda no almacenará ninguna información y permanecerá a *Null*.
- A una celda DMF se le asigna más de un registro DMF: Esto ocurre cuando el sensor ha adquirido

dos o más mediciones cercanas, todas ellas correspondientes al área de la escena que representa la celda DMF. En este caso no se seleccionará una de ellas ni se realizará promediados de ningún tipo, sino que se almacenarán todas ellas en forma de lista dinámica.

Para que la estructura de datos propuesta pueda considerar las tres circunstancias anteriores, es necesario que las celdas de la matriz DMF puedan contener o bien ningún registro DMR (*Null*), o bien uno o más registros DMR. En una primera aproximación, se ha considerado que una manera fácil y sencilla de almacenar un número variable de mediciones por celda DMF sería mediante una lista dinámica de registros DMR. Para ello, dicho registro poseerá un campo adicional *Siguiente*, que almacenará la dirección de memoria del registro DMR consecutivo en la lista.

El orden de los registros puede variar de acuerdo a las necesidades de la misión: Podrían estar ordenados por orden de adquisición (su creación sería más simple y rápida, pues únicamente habría que insertar el registro DMR recién adquirido al final de la lista almacenada en la celda DMF correspondiente), o bien por distancia a un punto de interés dentro de la celda DMF, como por ejemplo, el centro (Figura 4.5(a)). Sin embargo, hay que considerar que siempre que se desee tener las listas ordenadas según un determinado parámetro, el hecho de tener que re-ordenar cada vez que se adquiere una nueva medición puede ser una operación lenta y costosa que puede perjudicar el rendimiento de metodologías de procesamiento a bordo. Una alternativa recomendable sería almacenar la lista de registros DMR por orden de llegada, y, una vez terminada la adquisición, ordenar las listas de cada celda DMF de la matriz resultante según la política deseada.

El almacenamiento de las mediciones en las celdas DMF mediante listas dinámicas de registros DMR no tiene por qué ser la única opción. Sería interesante considerar estructuras alternativas que permitan además la gestión de las mediciones de manera ordenada espacialmente, por ejemplo, un árbol. Los árboles permiten el almacenamiento de los datos de una manera jerárquica de acuerdo a algún parámetro, como puede ser la distancia al centro, o más útil todavía, las coordenadas espaciales de las mediciones. De esta forma, la información proporcionada por el sistema de posicionamiento permitirán la indexación de los registros DMR mediante dos variables: latitud y longitud. Existen diferentes árboles que permiten realizar esta acción, sin embargo, uno de los más populares en el campo de segmentación de imágenes y el que mejor se atañe a las necesidades surgidas de la georreferenciación de datos distribuidos espacialmente es el Kd-tree [Ben75, Har92].

Mediante el uso de un árbol Kd-tree, las mediciones formarán una jerarquía estructurada dentro de una celda DMF, en vez de una lista dinámica simple (Figura 4.5(b)). De esta manera, el registro DMR funcionará al mismo tiempo como unidad de almacenamiento y de indexación, pudiendo tener cada uno de ellos hasta dos hijos de acuerdo a la variable de división: longitud para niveles impares, y latitud para niveles pares. El resultado es, por tanto, un árbol de varios niveles, en donde cada nodo estaría representado por la información espacial e instrumental de la medición, y un hijo izquierdo y otro derecho. El hijo izquierdo consistirá en un enlace a otro registro DMR (o a *Null* si no existen más mediciones) con un valor de divisor (longitud para niveles impares, latitud para niveles pares) menor o igual al del DMR padre, mientras que el hijo derecho consistirá en un enlace a otro registro DMR (o a *Null* si no existen más mediciones) con un valor de divisor (longitud para niveles impares, latitud para niveles pares) mayor al del DMR padre.

Además, los árboles Kd-tree de cada celda DMF estarán balanceados para asegurar un particionamiento del espacio equitativo y la optimización en la búsqueda de mediciones cercanas a un punto determinado. Esto se puede conseguir mediante la selección, como raíz del árbol Kd-tree correspondiente a la celda DMF, del registro DMR más cercano al centroide de todas las mediciones almacenadas en dicha celda.

Este diseño proporciona unas características adicionales al formato DMF respecto al almacenamiento mediante listas dinámicas de registros DMR:

- Los datos (registros DMR) están mejor organizados dentro de la celda DMF.

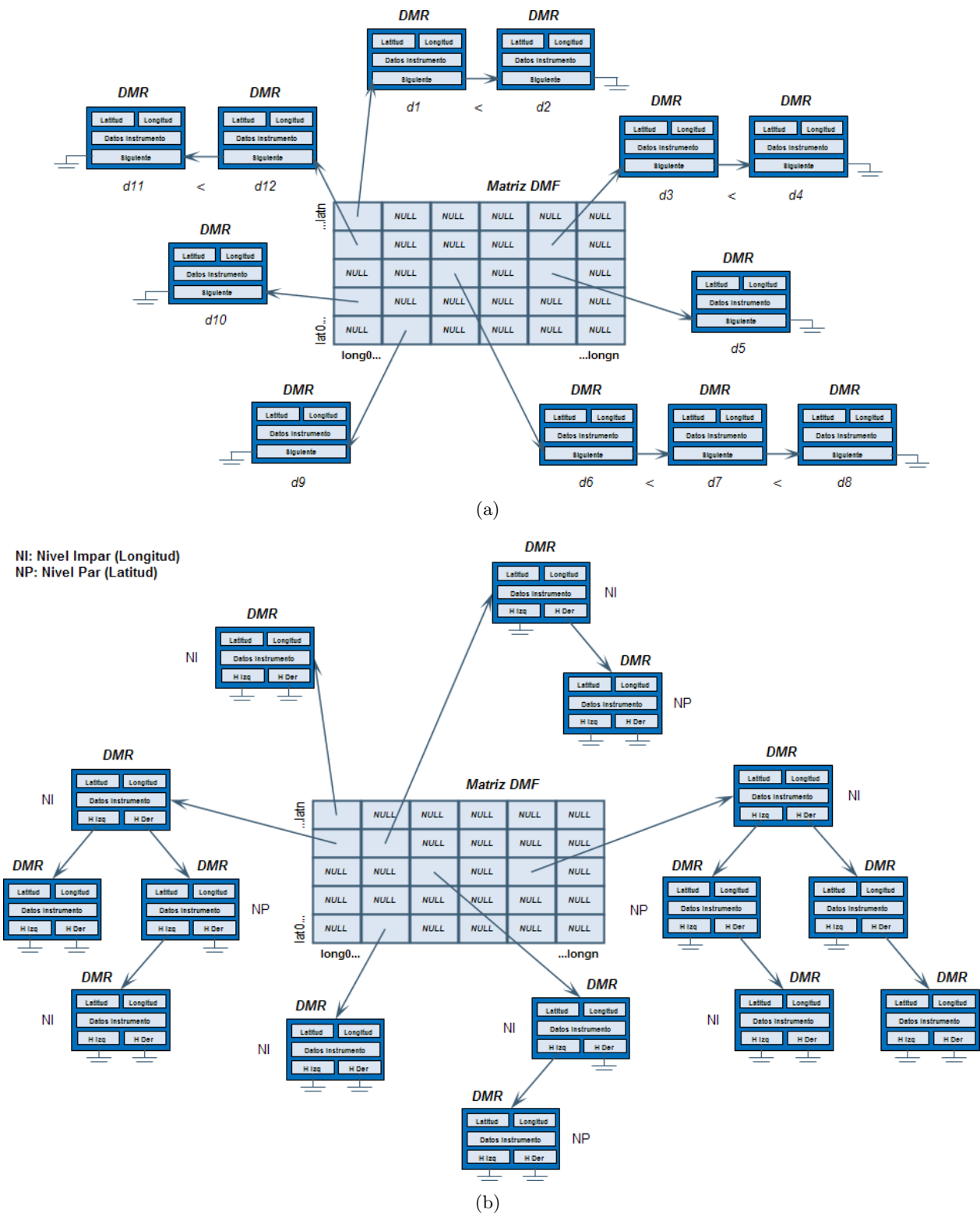


Figura 4.5: Alternativas al modo estándar de almacenamiento de los registros DMR en las respectivas celdas DMF.

(a) Uso de listas ordenadas respecto al centro de la celda DMF. (b) Uso de una estructura arbórea Kd-tree.

- Se decrementan los tiempos de búsqueda de la medición más cercana a un determinado punto, ya que ya no es necesario analizar de manera consecutiva todos los registros DMR almacenados en una celda DMF, sino descender en los niveles del árbol a través del hijo izquierdo o derecho (según el nivel y la variable de división correspondiente).
- Por el mismo motivo, los tiempos de búsqueda de los vecinos más próximos a una medición

también se decrementan. Esto es muy útil para operaciones de procesamiento de imágenes que requieran vecindad (especialmente filtrados espaciales).

- El uso de este diseño requiere un consumo de memoria adicional, ya que es necesario almacenar dos direcciones de memoria por registro DMR (hijo izquierdo e hijo derecho), en vez de sólo una.
- El hecho de tener que balancear el árbol Kd-tree de una determinada celda DMF cada vez que se inserta una medición en la creación de la estructura DMF puede ser una operación lenta y costosa, y para nada beneficiosa si el propósito es utilizar este formato a bordo y para procesamiento en tiempo real. La mejor alternativa es almacenar todos los registros DMR de manera secuencial (por orden de llegada), y, una vez terminada la adquisición de los datos, convertir todas las listas de DMR de cada celda DMF en árboles Kd-tree.

Aunque las alternativas para almacenar los registros DMR en las celdas DMF son múltiples, se puede considerar la lista dinámica ordenada por orden de adquisición (Figura 4.4) como la manera *estándar* de representación de la matriz DMF. De esta forma, en el presente documento, cada vez que se muestre alguna operación, modificación o adaptación de dicha matriz en alguna de las sucesivas figuras, la matriz DMF siempre aparecerá representada con un almacenamiento de registros DMR por orden de adquisición. No obstante, siempre puede considerarse dicho almacenamiento como cualquiera de las otras propuestas presentadas anteriormente (listas ordenadas por distancia a algún determinado parámetro, o árboles Kd-tree), si es necesario.

El formato DMF permite trabajar siempre con los datos reales tal y como fueron adquiridos por los sensores del avión o satélite, y facilita la posibilidad de cambiar la escala de la matriz DMF acorde a las necesidades de la misión, siempre preservando la información original. De esta manera, será posible procesar todos los datos incluso en situaciones extremas. Por ejemplo, considerando una resolución de la matriz DMF máxima, donde cada celda posee como máximo un registro DMR; o bien considerando una resolución mínima, donde la matriz DMF está constituida por una única celda que almacena todos los registros DMR.

La posibilidad de que el tamaño de celda DMF pueda ser variable, es un valor añadido, ya que según las circunstancias, es más interesante poseer una resolución de la matriz DMF mayor o menor a la establecida. Dos situaciones que pueden representar bien estas necesidades pueden ser las siguientes:

1. *Tipos de procesamiento en el que son necesarios analizar mediciones vecinas, o bien es necesario considerar algún parámetro espacial:* En este caso es interesante considerar una matriz DMF con una resolución que garantice lo máximo posible una medición DMR por celda. De esta manera, es posible encontrar las mediciones vecinas en celdas DMF próximas, minimizando el procesamiento de listas DMR por celda, y pudiendo encontrar celdas DMF vacías cuyo procesamiento se reduce a la comprobación de si existe dato o no.
2. *Tipos de procesamiento en el que se analizan todos los datos de manera secuencial:* En este caso es interesante considerar una matriz DMF con una resolución minimizada (incluso compuesta por una única celda), ya que las listas DMR son más largas y se ahorra el tiempo de analizar si una celda tiene contenido o no.

El hecho de que el tamaño de celda DMF sea variable (y por tanto la resolución de la matriz DMF) permite aumentar el rendimiento de la estructura de datos al poderse adaptar a sus necesidades. Sin embargo, este aspecto puede convertirse también en un inconveniente. Si no se está familiarizado con la información con la que se trabaja, o bien se trata de un sistema automático que desconoce las especificaciones técnicas de los instrumentos, entonces un tamaño de celda DMF no apropiado puede derivar en un mal rendimiento del formato. Por ejemplo, la elección de una resolución demasiado pequeña puede generar un tamaño de matriz DMF demasiado grande con demasiadas celdas sin información (*Null*), que puede implicar un mayor tiempo de cómputo. De la misma manera, la elección

de una resolución demasiado grande puede generar un tamaño de matriz DMF demasiado pequeño, donde la densidad de registros DMRs sea alto (listas de DMRs muy largas), implicando también un mayor tiempo de cómputo siempre que se consideren parámetros espaciales.

Por tanto, es importante el desarrollo de metodologías y algoritmos que permitan el cálculo óptimo de tamaño de celda DMF de manera automática, a partir del análisis de las distancias, tanto en latitud, como en longitud, de las mediciones adquiridas.

Aparte de los campos de coordenadas espaciales y valor adquirido por el instrumento, es posible ampliar el número de campos para almacenar información adicional. Un ejemplo de esto puede ser la adaptación del formato DMF para fusión de datos (Sección 4.1.2.1). Introduciendo campos adicionales, como por ejemplo el identificador del sensor que realizó la adquisición, es posible introducir todas las mediciones obtenidas por diferentes sensores a bordo de plataformas también diferentes en la misma matriz DMF. El resultado es la escena representada por registros DMR de diferente procedencia (y tipo), que permiten un procesamiento adecuada a la misión.

De la misma manera, introduciendo campos adicionales en el registro DMR, es posible adaptar el formato a metodologías basadas en detección de cambios (Sección 4.1.2.2). En lugar de un identificador de sensor, el campo adicional consistiría en un *timestamp* que indicaría la hora y fecha de adquisición de la medición. El resultado es la escena representada por registros DMR adquiridos en diferentes intervalos de tiempo. El hecho de que en cada celda DMF convivan registros DMR obtenidos en diferentes fechas hacen más sencilla la detección de cambios, al tener los datos ya georreferenciados desde un principio, sin importar las geometrías de los sensores ni los problemas de posicionamiento derivados de procesos de geocorrección en imágenes convencionales.

A continuación se presenta, de manera detallada, todas las soluciones comentadas con anterioridad. Los siguientes apartados se dividen en dos bloques principales. En el primero de ellos (Sección 4.1.1) se detallan las operaciones que son posibles realizar con el formato DMF, equivalentes a las operaciones típicas de cualquier formato de datos convencional (creación, visualización, almacenamiento y procesamiento). En el segundo bloque, se presentarán adaptaciones más complejas del formato DMF (Sección 4.1.2), donde se han añadido nuevos campos al registro DMR, orientando dicho formato a metodologías como la fusión de datos y la detección de cambios.

4.1.1. Operaciones con el formato DMF.

El formato DMF parte de una estructura matricial convencional pero modificada para poder minimizar los errores y problemas derivados a la hora de posicionar y georreferenciar los datos. Por tanto, es necesario adaptar la estructura de datos para que permita realizar las mismas operaciones básicas que suelen utilizarse sobre los formatos de datos convencionales, como son la creación de la estructura de datos (Sección 4.1.1.1), la visualización (Sección 4.1.1.2), el almacenamiento en disco (Sección 4.1.1.3) y el procesamiento de la información (Sección 4.1.1.4).

4.1.1.1. Creación de la matriz DMF.

Para la creación de la matriz DMF es necesario tanto la información instrumental como la espacial de cada una de las mediciones. La información instrumental es proporcionada por los sensores a bordo del avión o satélite, mientras que la espacial es proporcionada por el dispositivo de posicionamiento. El origen de los datos puede variar. Puede encontrarse en ficheros si la adquisición ya ha sido realizada (por el avión o el satélite), o pueden proceder directamente de los instrumentos, potenciando el procesamiento a bordo.

Para el primer caso, esta información suele proporcionarse a través de dos ficheros diferentes: el primero de ellos (L0) almacenando la información de los instrumentos, y el segundo (IGM) las

coordenadas de cada medición (Sección 3.2.6.1). De esta manera, el primer paso a realizar para la generación de la matriz DMF es obtener el área espacial en el que se van a situar las mediciones, es decir, la cobertura espacial de la escena. Para ello, se obtendrán las longitudes y latitudes máximas y mínimas a partir del fichero IGM. Mediante un tamaño de celda DMF predeterminado, el número de filas y columnas de la matriz DMF vendrá representado por:

$$\begin{aligned} N_c &= \left\lceil \frac{M_L - m_L}{T_L} \right\rceil \\ N_f &= \left\lceil \frac{M_l - m_l}{T_l} \right\rceil \end{aligned} \quad (4.1)$$

Donde M_L , m_L , M_l , m_l , son la longitud máxima, longitud mínima, latitud máxima y latitud mínima encontradas en el fichero IGM, y T_L , T_l el tamaño de celda DMF seleccionado en longitud y latitud, respectivamente.

Una vez creada la matriz DMF con esta información, sólo basta leer uno a uno los valores del fichero L0, junto con sus coordenadas del fichero IGM, para así crear un registro DMR. A continuación dicho registro ha de almacenarse en la celda DMF correspondiente, cuya posición $[x, y]$ es determinada mediante:

$$\begin{aligned} x &= \left\lceil \frac{dmr_L - m_L}{T_L} \right\rceil \\ y &= \left\lceil \frac{dmr_l - m_l}{T_l} \right\rceil \end{aligned} \quad (4.2)$$

Donde dmr_L y dmr_l son las coordenadas de la medición en longitud y latitud, m_L y m_l la longitud y latitud mínimas de la matriz DMF, y T_L , T_l el tamaño de celda DMF seleccionado en longitud y latitud.

Ya localizada la posición de la celda DMF en la cual debe residir el registro DMR, pueden ocurrir dos circunstancias:

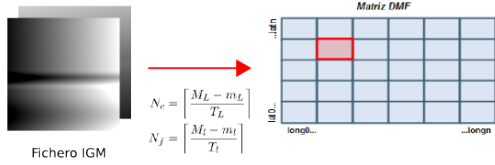
- *La celda DMF no tiene medición (Null)*: El registro DMF será el primer registro almacenado en dicha celda.
- *La celda DMF tiene una o varias mediciones*: Localizar el último registro DMR de la lista y enlazarlo al registro DMR recién creado, siendo ahora éste el último de la lista.

Al terminar la creación de la matriz DMF y la asignación de los registros DMR, éstas estarán almacenadas en sus correspondientes celdas DMF de manera secuencial de acuerdo al orden de adquisición o llegada a la estructura. Tal y como se recomendó en la Sección 4.1, es en este momento en el que puede optimizarse la manera en la que los registros DMR son gestionados. Sería necesario recorrer la matriz celda DMF a celda DMF, re-ordenando las mediciones según el parámetro de interés (por ejemplo, distancia al centro de la celda DMF), o bien construyendo los árboles Kd-tree correspondiente. Por supuesto, este tipo de operaciones de optimización del almacenamiento de las mediciones suponen un incremento de tiempo de construcción adicional, que dependerá directamente del número de mediciones presentes en la celda DMF.

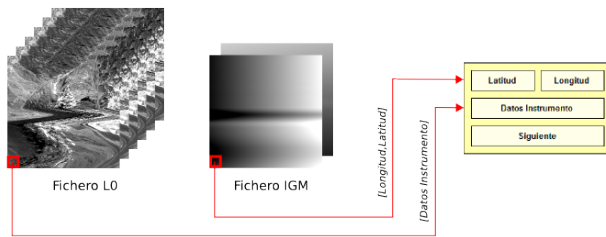
Todo este procedimiento de creación de la matriz DMF se muestra detallado en la Figura 4.6.

Mediante esta metodología es necesario determinar las longitudes y latitudes máximas y mínimas, de las mediciones adquiridas por los sensores, así como el tamaño de celda DMF para poder construir

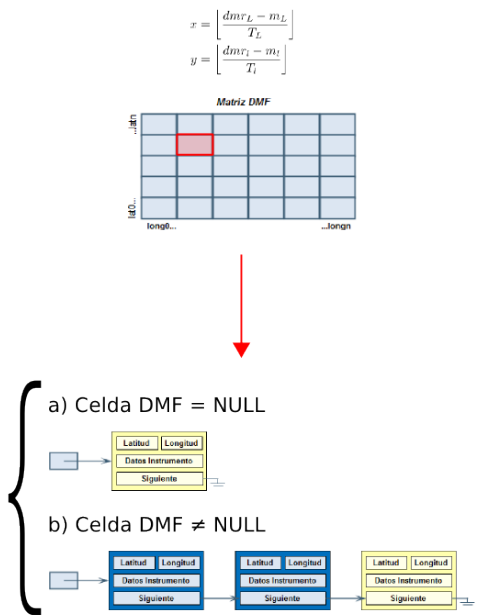
1. Creación de la matriz DMF



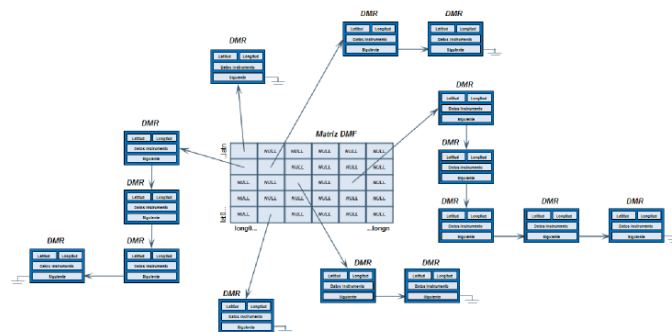
2. Creación de los registros DMR



3. Asignación de registros DMR a celdas DMF

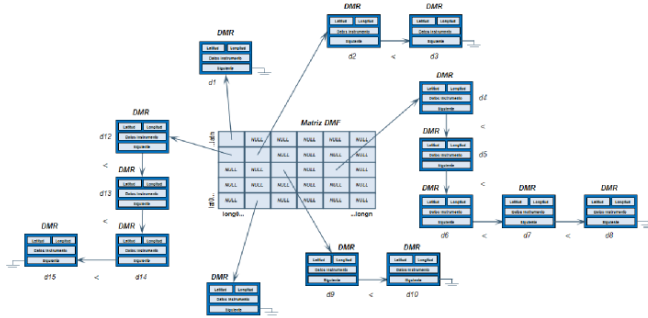


4. Generación de la matriz DMF resultante



5. Optimización de la matriz DMF (opcional)

a) Listas ordenadas



b) Árboles Kd-trees

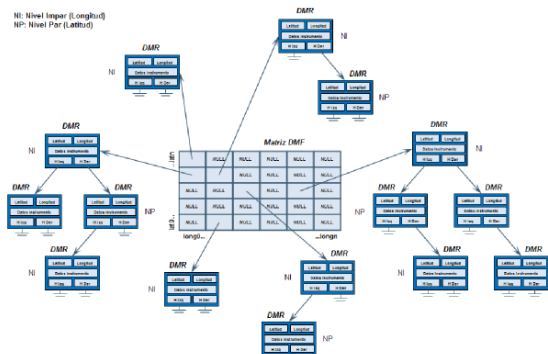


Figura 4.6: Proceso de creación de la matriz DMF.

la matriz DMF, es decir, es necesario conocer los datos previamente antes de poder construirla. Esto puede ser un inconveniente si se necesita utilizar la estructura DMF directamente para procesar la imagen en tiempo real desde un dispositivo que automáticamente reciba la información de los sensores, y en donde la disposición de los datos son desconocidos. Para el caso del tamaño de celda, es fácil estimar el más oportuno antes de empezar la captura de datos, de acuerdo a la altura de la plataforma y la geometría del sensor. Sin embargo, las coordenadas máximas y mínimas de la escena no pueden resultar tan fáciles de estimar de manera automática. En estos casos, se pueden utilizar diferentes políticas:

- *Selección manual de unas coordenadas máximas y mínimas acorde a la escena en la que se van a obtener los datos:* Esto requiere el conocimiento previo de la región sobre la que se va a volar y reconocer, muy característico de la adquisición de imágenes desde aviones. De esta manera, la matriz DMF está construida antes de comenzar la adquisición, y el dispositivo únicamente se limita a posicionar los datos que le llegan, y a su procesamiento (si se requiere). Con esta opción es posible encontrar que:
 - *La matriz DMF es demasiado grande y los datos cubren una escena reducida:* Tras la adquisición de los datos (y procesamiento, si es requerido) pueden recortarse los bordes a *Null* y limitar la matriz DMF al área cubierta por las mediciones.
 - *La matriz DMF es demasiado pequeña y existen datos que son asignados fuera de ella:* Almacenar dichos datos en los bordes (primera o última fila o columna) de la matriz DMF. Tras la adquisición (y procesamiento, si es requerido) puede escogerse un tamaño y dimensionalidad de matriz DMF diferente que permita el correcto posicionamiento de las mediciones.
- *Almacenamiento secuencial de registros DMR, en lugar de utilizar una matriz DMF previamente construida para su posicionamiento:* Ésta es la opción más recomendable. Los registros DMR pueden ser procesados (si se requiere) mientras se determina a su vez las coordenadas máximas y mínimas, y, una vez la captura de datos ha finalizado, puede construirse la matriz DMF de manera automática asignando los registros DMR (procesados o no). Sin embargo, esta política puede limitar el rendimiento y los tiempos de ejecución de la metodología de procesamiento si se necesitan los vecinos próximos a cada medición para obtener un resultado (ya que los datos no están posicionados y es difícil determinar la vecindad de los mismos).

4.1.1.2. Visualización DMF.

La esta estructura de datos DMF está compuesta por una matriz cuyas celdas pueden contener ninguna, una o varias de las mediciones adquiridas por los sensores. Este diseño dificulta en cierta medida la visualización de los datos, ya que no es tan directa y trivial como en los formatos convencionales, en donde a cada píxel de la imagen le corresponde un píxel de la pantalla.

No se desea aumentar la complejidad en el modo de mostrar los datos en pantalla respecto a los formatos convencionales, por lo que existen dos problemas que son necesarios abarcar:

1. *Una celda DMF puede estar compuesta por varios registros DMR:* Si se desea realizar una asignación directa de tal manera que a cada celda DMF le corresponda un píxel de pantalla esto es un problema, ya que es imposible visualizar todas las mediciones residentes en la celda DMF correspondiente. Sería necesario seleccionar solamente una de acuerdo a alguna política o las necesidades de las misiones espaciales. Varias posibilidades serían, por ejemplo, mostrar la media de los valores de los registros DMR almacenados en la misma celda DMF, o el valor del registro DMR más cercano al centro. Se ha considerado esta última política, ya que presentaría visualizaciones más cercanas a la realidad.

2. *Una celda DMF puede estar a Null sin almacenar registro DMR alguno*: Esta circunstancia ha de denotarse al realizar la transformación a una visualización RGB en pantalla. Es decir, es necesario asignar un determinado color a las celdas DMF a *Null* para indicar de que se tratan de sub-áreas sin mediciones. Como este tipo de celda DMF conformarán el fondo de la escena escaneada, es recomendable utilizar un color oscuro para realzar el contraste, por ejemplo, un color negro. Sin embargo, hay que tener en cuenta que los valores de los datos serán representados mediante la utilización de un determinado esquema de color compuesto por la escala entre dos colores extremos, convencionalmente el negro y el blanco. De esta manera, al valor de medición más bajo le corresponderá el color negro, mientras que al más alto le corresponderá el blanco. Pero si se utiliza el color negro para representar las celdas DMF a *Null*, éstas pueden confundirse con valores de medición bajos, por lo que se hace necesario cambiar el esquema de color. Un ejemplo de esquema de color que permite una buena representación de la escena escaneada posee como extremos el azul oscuro y el blanco puro, de tal forma que a los valores de medición bajos les corresponde un color azul oscuro, mientras que a los altos les corresponde un color blanco puro.

En resumen, si se considera una determinada matriz DMF y se desea la visualización de la misma en pantalla, hay que considerar que se realiza una correspondencia de una celda DMF por píxel de pantalla. De acuerdo al número de registros DMR presentes en la celda DMF, pueden darse diferentes circunstancias:

- *La celda DMF tiene un valor Null (vacía, sin mediciones)*: El píxel se mostrará de color negro.
- *Si la celda DMF no está vacía (una o más mediciones)*: Se mostrará la medición más cercana al centro de la celda (la única existente si la celda posee sólo un registro DMR). El valor de medición se representará de manera escalada según dos extremos que corresponderán a un azul oscuro (RGB:0.0.30) para el valor más bajo, y blanco puro (RGB:255.255.255) para el valor más alto.

La Figura 4.7 muestra de manera esquemática el resultado de la visualización de una determinada matriz DMF.

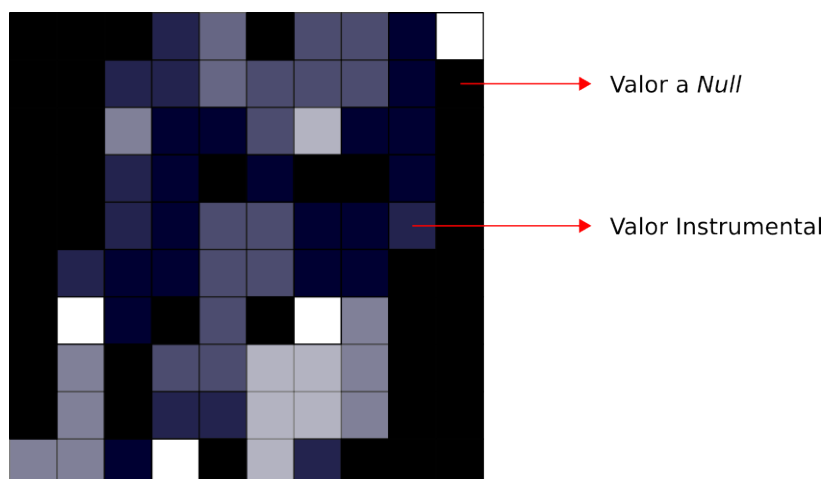


Figura 4.7: Visualización de una determinada banda de la imagen mediante el formato DMF.

Como se puede comprobar, a la hora de visualizar la estructura DMF el proceso es un poco limitado, al mostrar únicamente una medición por píxel y descartando la visualización de las demás en caso de que la celda DMF esté compuesta por más de un registro DMR. No obstante, esto sólo afecta a la visualización, no al procesamiento. Como método de visualización este proceso puede ser suficiente, pero es necesario recordar que si se quiere analizar, procesar o tratar la imagen representada

mediante la estructura DMF, hay que considerar todos y cada uno de los registros DMR almacenados en cada celda DMF, independientemente de cómo los datos se muestren en pantalla.

Si se requiere visualizar en mayor detalle los datos almacenados en una estructura DMF, se pueden considerar diferentes técnicas:

- *Construcción de una matriz DMF análoga a la original pero con una mayor resolución espacial (con un tamaño de celda DMF más pequeño):* Esto permitiría que los datos se mostrasen con mayor definición. La máxima resolución se consigue mediante la consideración de un tamaño de celda DMF igual a la mínima distancia entre dos mediciones. En este caso, todas las celdas DMF tendrán como máximo un registro DMF, y además las dimensiones espaciales de la imagen RGB resultante pueden ser demasiado grandes, dificultando su correcta visualización.
- *Visualización dinámica del contenido de la celda DMF:* Modificación del proceso de visualización para que, de manera interactiva al analista experto, se muestren todas las mediciones correspondientes a una celda DMF. Esto puede realizarse mediante el muestreo de la matriz DMF de la manera propuesta anteriormente, y cuando el analista seleccione un determinado píxel, se muestre al margen de la imagen (en una sub-imagen adicional o en otra sección de la pantalla), la información instrumental de todos los registros DMR almacenados en la celda DMF análoga.

4.1.1.3. Almacenamiento DMF.

Es muy importante definir la manera en la cual el formato DMF será almacenado en disco, ya que será la manera en la que se proveerán los datos a la comunidad científica. Tal y como ocurre en los formatos de almacenamiento convencionales, es primordial que el tamaño del fichero resultante sea mínimo y que la carga de los datos en memoria RAM sea rápida.

A la hora de realizar un almacenamiento de los datos en disco a partir de una matriz DMF previamente construida, es posible tener en cuenta varias consideraciones si se quiere realizar este procedimiento de manera óptima y eficiente.

En un primer lugar, hay que prestar especial atención en el tipo de dato que representan los valores de los instrumentos, y sobre todo, del posicionamiento de los mismos. Para los valores de las mediciones, el tipo de dato y la precisión depende directamente del instrumento utilizado y su calidad. Sin embargo, para el caso de las coordenadas correspondientes a cada medición, proporcionada por el sistema de posicionamiento, el tipo de datos suele ser de gran tamaño proporcionando un gran número de decimales. Pero es muy difícil que el instrumento utilizado como sistema de posicionamiento presente tal precisión, siendo ésta generalmente del orden de décímetros, e incluso metros. Por tanto, no tiene sentido emplear un tipo de datos en coma flotante de gran tamaño (generalmente 64 bits), cuando se puede emplear otro tipo de datos, también coma flotante, pero de menor tamaño (32 bits). De esta forma, teniendo en cuenta siempre las especificaciones técnicas del sistema de posicionamiento (concretamente la precisión espacial), es posible redondear los valores de coordenadas proporcionados a la máxima precisión ofrecida, y reducir así el tamaño de datos referente a la información espacial prácticamente a la mitad.

En segundo lugar, es necesario determinar la manera en la que se va a representar la matriz DMF en disco. Siguiendo las mismas pautas que en los formatos de archivos convencionales, el formato de archivo DMF estará compuesto por un fichero cabecera y un fichero de datos.

Respecto al fichero cabecera, es interesante y útil almacenar en modo texto toda la información posible sobre la matriz DMF a almacenar. Esto permite la extracción correcta de los datos tanto por los algoritmos mediante analizadores sintácticos, así como el análisis visual por parte de expertos. Dicha información puede incluir los siguientes campos:

- *Num Rows*: Número de filas de la matriz DMF.
- *Num Columns*: Número de columnas de la matriz DMF.
- *Num Bands*: Número de bandas que componen el valor instrumental.
- *Cell Size X*: Tamaño de celda DMF en X.
- *Cell Size Y*: Tamaño de celda DMF en Y.
- *Num DMRs*: Número de mediciones (registros DMRs) almacenados en la matriz DMF.
- *Bytes Instrument*: Número de bytes que ocupa un valor instrumental.
- *Bytes Coordinates*: Número de bytes que ocupa un valor de coordenada.
- *Min Longitude*: La mínima longitud de todas las mediciones.
- *Max Longitude*: La máxima longitud de todas las mediciones.
- *Min Latitude*: La mínima latitud de todas las mediciones.
- *Max Latitude*: La máxima latitud de todas las mediciones.
- *DMR Storage Type*: Tipo de almacenamiento de los registros DMR en las celdas DMF.

Los campos *Num DMRs*, *Bytes Instrument* y *Bytes Coordinates* son útiles para obtener los datos del fichero, ya que nos indican el número de registros a leer y su tamaño. Los campos *Num Rows*, *Num Columns*, *Num Bands*, *Cell Size X* y *Cell Size Y* permiten construir la matriz DMF original en memoria RAM. Los campos *Min Longitude*, *Max Longitude*, *Min Latitude*, *Max Latitude* ayudan a construir la matriz DMF original en memoria RAM, ya que nos indica de manera directa el área que abarca la imagen sin tener que analizar todos los registros DMR, y por tanto el posicionamiento de los mismos es más rápido. El campo *DMR Storage Type* indica el modo de almacenamiento de los registros DMR en las celdas DMF, y puede tomar valores predefinidos de acuerdo a si ésta se ha realizado mediante una lista ordenada por llegada de los datos (*Ninguna*), mediante una lista ordenada por distancia al centro de la celda DMF (*Distancia_Centro*), o mediante una estructura arbórea (*Kd-Tree*).

Respecto al fichero de datos, éste almacenará toda la información residente en la matriz DMF, es decir, los registros DMR presentes en sus celdas DMF (valor instrumental y coordenadas). Se puede optar entre las siguientes alternativas: Una primera opción en la que se salvase en disco el estado de la matriz DMF tal cual; una segunda opción que también apostara por esta política, pero con la ayuda de máscaras adicionales para una posterior carga de la información de manera más rápida; y una tercera opción en la que se salvaran los registros DMR de manera secuencial.

La primera alternativa apuesta por almacenar la estructura matricial, tal cual, en disco. Al no tratarse de una matriz regular, sino de un tipo especial en la que es posible encontrar ninguno, uno o varios registros DMR en sus celdas DMF, es necesario completar la información para un correcto almacenamiento, y posterior carga, de datos. Esta información adicional estará compuesta por pequeños campos que indican el número de registros DMR en la presente celda. Es decir, el fichero de datos DMF estará compuesto por una matriz de las mismas dimensiones en donde cada celda DMF quedará representada por un contador que indicará el número de registros DMR que la forman, y la secuencia de dichos registros DMR (Figura 4.8). Es importante seleccionar un tipo de datos correcto para este contador, ya que un tamaño excesivo puede implicar un incremento considerable del tamaño de fichero final. Un tipo de datos entero sin signo de 8 bits (*char*) permitirá identificar hasta un total de 255 registros DMR por celda DMF, lo que debería ser suficiente. En caso de necesitar almacenar un número mayor sería posible considerar un tipo de datos entero sin signo de 16 bits (*short int*), que limitaría hasta un máximo de 65535 registros DMR, pero que supondría un incremento del tamaño de la

información espacial hasta el doble. De esta manera, la carga de una matriz DMF de disco implicará la lectura, una a una, de todas las celdas DMF que la componen (información proporcionada por el fichero cabecera), y considerando que la lectura de cada celda DMF conlleva la lectura del número de registros DMR que la forman primeramente (contador), y después la información de dichos registros DMR de manera secuencial.

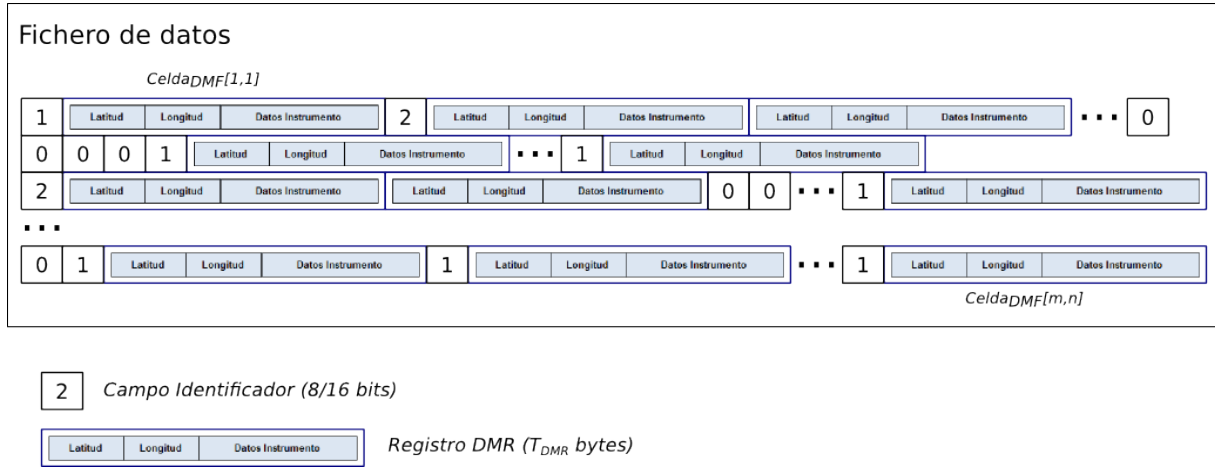


Figura 4.8: Almacenamiento de la matriz DMF en fichero mediante la utilización de campos contadores.

El tamaño de un único registro DMR está definido por la Ecuación 4.3, en donde T_{Inst} representa el tamaño en bytes de los valores de medición, y T_{Coord} el tamaño en bytes de un valor de coordenada (como se emplean dos valores, latitud y longitud, este parámetro es multiplicado por 2), que podrán ser 4 bytes (32 bits) ó 8 bytes (64 bits) dependiendo del tipo de datos seleccionado para la representación. Siguiendo el procedimiento de almacenamiento DMF presentado, el tamaño en bytes del fichero estará determinado por:

$$T_{DMR} = T_{Inst} + 2 \times T_{Coord} \quad (4.3)$$

$$T_{DMF} = N_f \times N_c + N_{DMRs} \times T_{DMR} \quad (4.4)$$

Donde N_f y N_c representan el número de filas y columnas de la matriz DMF (1 Byte por celda DMF para el contador); y $N_{DMRs} \times T_{DMR}$ el tamaño total de todos los datos representados como registros DMR, con N_{DMRs} representando el número total de éstos.

El principal inconveniente que presenta esta propuesta de almacenamiento proviene de la manera en la que se distribuyen los datos sobre la escena. Como resultado de una adaptación de la imagen a una matriz DMF se generará un porcentaje de celdas DMF con un valor *Null*, lo que supone un alto número de campos de contador con un valor de 0. Es decir, un gran porcentaje del tamaño final del fichero DMF estará dedicado a indicar que la celda DMF no contiene dato. La segunda alternativa supone una posible mejora del método de almacenamiento para minimizar el tamaño debido a esta circunstancia podría ser la utilización de una máscara cabecera binaria matricial que indicase, para cada celda DMF, si contiene dato o no. Tras esta máscara binaria, se encontrarían almacenadas únicamente las celdas DMF que contuvieran algún registro DMR según el procedimiento ya propuesto: inicialmente un contador de 1 Byte para indicar el número de registros DMR en la celda DMF correspondiente (contador), y a continuación una lista de dichos registros DMR (Figura 4.9).

Por tanto, la lectura de los datos del fichero DMF se realizaría mediante la utilización de dos descriptors, uno realizando la lectura de la máscara matricial binaria y el segundo los datos correspondientes a las celdas DMF. Cuando el primer descriptor encontrara un valor binario en la máscara matricial con valor 1, el segundo descriptor realizaría la lectura de la celda DMF (contador

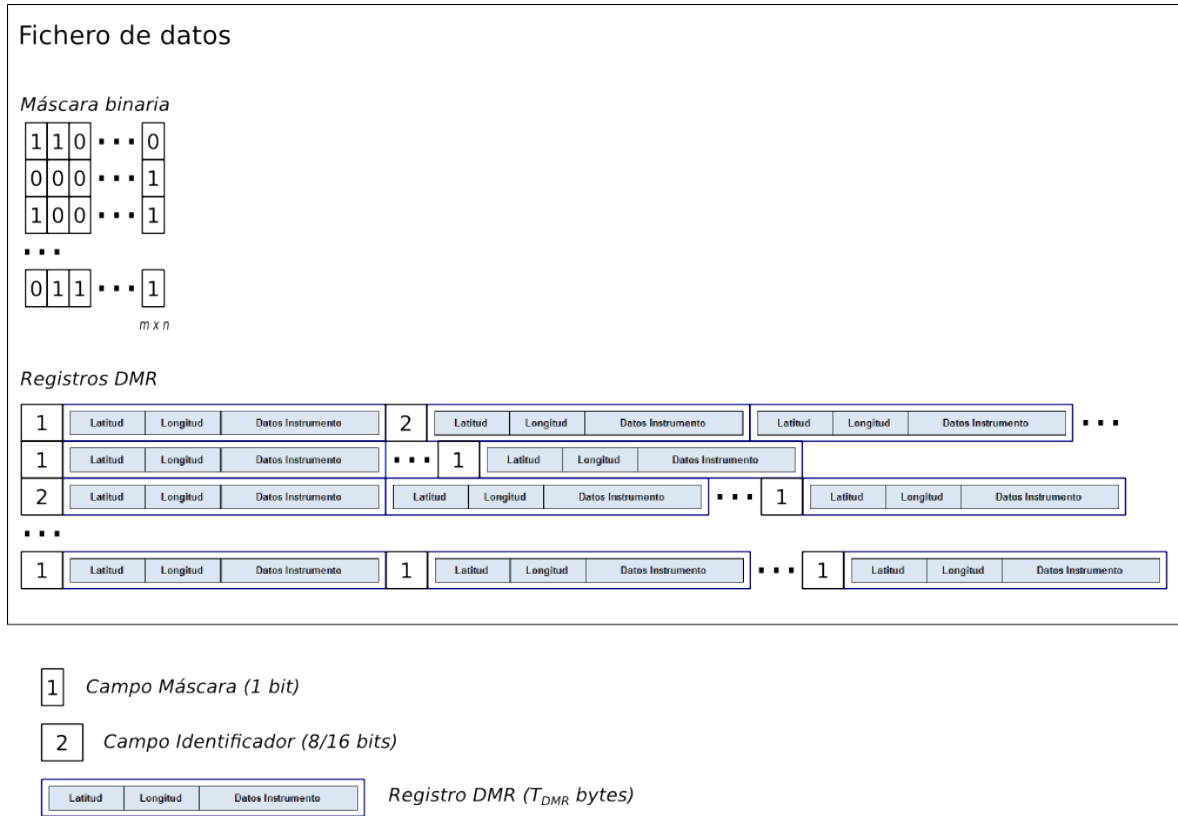


Figura 4.9: Almacenamiento de la matriz DMF en fichero mediante la utilización de máscara y campos contadores.

y secuencia de registros DMR), para ser insertados en la posición de la matriz DMF correspondiente a la posición en la máscara matricial del primer descriptor.

Mediante esta propuesta de almacenamiento mejorada, el tamaño en bytes del fichero estará determinado por:

$$T_{DMF} = \left\lceil \frac{N_f \times N_c}{8} \right\rceil + N_{C_{DMF \neq Null}} + N_{DMRs} \times T_{DMR} \quad (4.5)$$

Donde $\lceil (N_f \times N_c)/8 \rceil$ representa el tamaño de la máscara binaria matricial (al ser binaria, la representación de 8 celdas DMF implicarán el tamaño de 1 Byte), con N_f y N_c representando el número de filas y columnas de la matriz DMF; $N_{C_{DMF \neq Null}}$ el número de celdas DMF que contienen uno o varios registros DMR (1 Byte por cada una de ellas para el contador); y $N_{DMRs} \times T_{DMR}$ el tamaño total de todos los datos representados como registros DMR, con N_{DMRs} representando el número total de éstos.

Sin embargo, el tamaño de fichero mínimo se obtendría almacenando únicamente la información útil adquirida por los sensores a bordo del avión o del satélite y por el sistema de posicionamiento, sin campos adicionales para la gestión de una correcta lectura posterior. Esto es posible almacenando uno tras otro, todos los registros DMR presentes en la matriz DMF (Figura 4.10), tal y como propone la tercera alternativa. De esta manera, en el fichero de datos no se salvaría ninguna información sobre la posición de las mediciones en dicha matriz, únicamente los valores instrumentales y espaciales. A partir de las coordenadas que abarca la matriz DMF (proporcionadas por el fichero cabecera mediante los campos *Min Longitud*, *Max Longitud*, *Min Latitud* y *Max Latitud*), y las coordenadas espaciales de cada registro DMR, sería posible determinar la posición del mismo dentro de la matriz DMF. Por tanto, la lectura de los datos del fichero DMF se realizaría de manera secuencial, registro DMR por registro DMR, obteniendo para cada uno de ellos la información instrumental y sus coordenadas.

A partir de dichas coordenadas, y utilizando la Ecuación 4.2, es posible determinar la celda DMF correspondiente. Este procedimiento, por tanto, supone un incremento en los tiempos de carga de la matriz DMF de fichero, al tener que calcular la posición en dicha matriz, de cada uno de los registros DMR a partir de su información espacial. Sin embargo, proporciona dos ventajas que la convierten en la mejor alternativa como formato de almacenamiento (de las tres propuestas). La primera ventaja es que el tamaño en disco se reduce al máximo (ya que únicamente se almacena información útil). La segunda ventaja es que este formato se adapta completamente a la manera en la que se adquieren los datos. A medida que se van obteniendo las mediciones, es posible almacenarlas de manera secuencial como registros DMR en un fichero de manera directa. Una vez terminada la adquisición, sólo bastaría con calcular los campos del fichero cabecera y construirlo, permitiendo que la representación de la escena esté disponible en formato DMF en el acto, sin necesitar procesos de geocorrección previos.

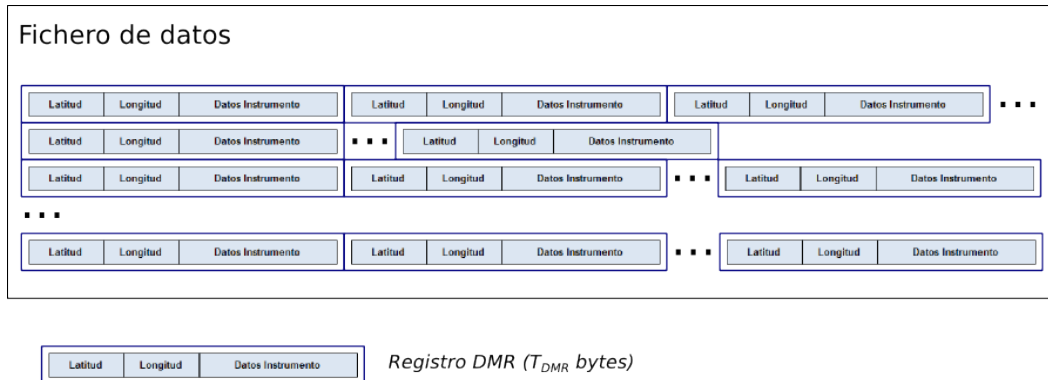


Figura 4.10: Almacenamiento de la matriz DMF en fichero considerando únicamente la secuencia de registros DMR.

Con esta propuesta de almacenamiento mejorada, el tamaño en bytes del fichero quedará reducido tal que:

$$T_{DMF} = N_{DMRs} \times T_{DMR} \quad (4.6)$$

Donde $N_{DMRs} \times T_{DMR}$ representa el tamaño total de toda la información útil adquirida por los sensores a bordo y por el sistema de posicionamiento, con N_{DMRs} representando el número total de registros DMR.

En resumen, los tres procedimientos de almacenamiento del formato DMF pueden implicar una serie de ventajas, pero también de inconvenientes.

La ventaja principal que presenta este procedimiento es el hecho de considerar únicamente los datos reales adquiridos por los sensores a bordo del avión o satélite, por lo que el tamaño de la información a almacenar se reduce respecto a los formatos de datos convencionales. Éstos suelen almacenar cierta información redundante o inválida, como pueden ser espectros a 0 que componen los bordes indeterminados de la imagen, o bien espectros duplicados al ser asignados, durante el proceso de geocorrección, a diferentes píxeles por vecino más próximo. Además, si se considera un almacenamiento secuencial de los registros DMR, el formato de almacenamiento DMF supone una gran alternativa, ofreciendo una representación de la escena geocorregida en el mismo instante en el que finaliza la adquisición de mediciones.

Sin embargo, el principal inconveniente es que, a pesar de que únicamente se almacena la información útil adquirida de los sensores a bordo y del sistema de posicionamiento, el almacenamiento de la estructura DMF sí que requiere un consumo de almacenamiento extra causado por la adición de las coordenadas de todas las mediciones. Los formatos de datos convencionales suelen utilizar las coordenadas del píxel de la esquina superior izquierda como referencia, y a partir del tamaño de celda, calculan las posiciones del resto de los píxeles. De esta manera, no necesitan almacenar cada una

de las coordenadas de los píxeles, requisito que sí es indispensable en el formato DMF si se quiere conservar la minimización de los errores de posicionamiento de los datos. Además, este problema puede agravarse si se utilizan las dos primeras propuestas, que requieren campos adicionales para el correcto posicionamiento de los registros DMR almacenados sobre las celdas DMF correspondientes.

4.1.1.4. Procesamiento DMF.

Para que el formato DMF suponga una alternativa a los formatos de datos convencionales, es necesario definir la manera en la que se van a procesar los datos dentro de la matriz DMF. En los formatos convencionales, la estructura que representa la imagen escaneada está constituida por una matriz (generalmente tridimensional) en la cual es posible aplicar de manera fácil y simple operaciones matemáticas propias de la teoría de matrices.

Sin embargo, a pesar de que el formato DMF está también compuesto principalmente por una matriz, el hecho de poder almacenar ninguna, una o varias mediciones en cada una de sus celdas implica tener que redefinir la manera en la que funcionan estas operaciones básicas. En el campo del procesamiento de imágenes, es posible distinguir entre dos tipos de operaciones, dependiendo de si requieren o no vecindad. Para el caso de imágenes convencionales, los métodos que no requieren vecindad consideran el píxel como unidad básica de procesamiento, y en la que para obtener un resultado no se necesitan los píxeles de la periferia. En los métodos que sí requieren vecindad, la unidad básica de procesamiento la conforma un ventana o *kernel*, una agrupación de píxeles cercanos necesarios para conseguir un determinado resultado. Para el caso del formato DMF, el principio es el mismo. Las operaciones que no requieren vecindad considerarán como unidad básica de procesamiento el registro DMR. De esta manera, para conseguir un resultado se centrará en el registro DMR actual sin considerar los demás registros DMR insertados en la matriz DMF. Las operaciones que sí requieren vecindad tendrán como unidad básica de procesamiento un pequeño área de la escena formada por una serie de registros DMR. Todos ellos serán utilizados para obtener un determinado resultado [Cri12a].

La Sección 4.1.1.4.1 detallará la manera en la que se realizan las operaciones que no requieren vecindad para el formato DMF, mientras que la Sección 4.1.1.4.2 la detallará para las operaciones que sí la requieren.

4.1.1.4.1. Operaciones que no requieren vecindad.

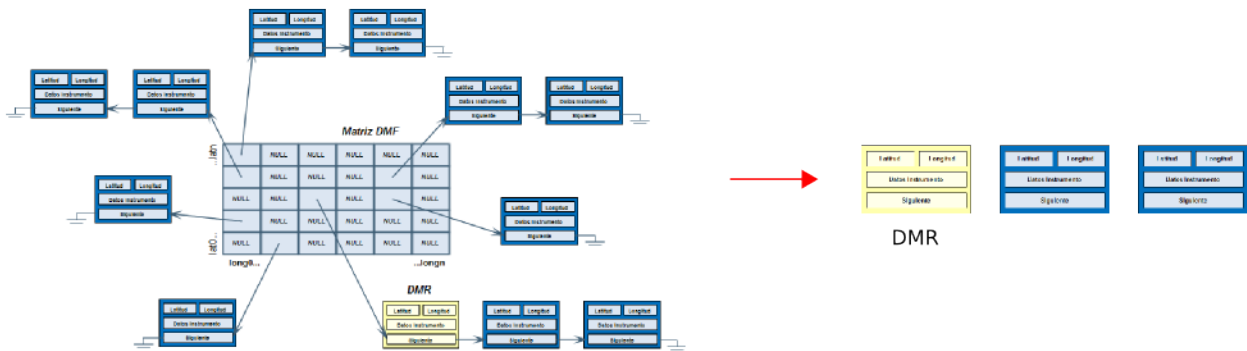
Las operaciones que no requieren vecindad son metodologías que procesan cada medición sin considerar las mediciones adyacentes. Este tipo de procesamiento puede involucrar varias técnicas de tratamiento de imágenes: umbralizados, cálculos estadísticos (media, desviación estándar, ángulo espectral, etc.), clasificaciones y desmezclado espectral, etc.

En una imagen representada mediante una estructura matricial convencional este tipo de operación es sencillo. Únicamente es necesario recorrer cada píxel de la imagen y aplicar a cada medición la operación correspondiente. En una imagen representada mediante el formato DMF, el procedimiento es el mismo, pero hay que tener en cuenta que cada celda DMF puede contener ninguna, una o varias mediciones, por lo que hay que procesar todos los registros DMR insertados en la celda DMF correspondiente.

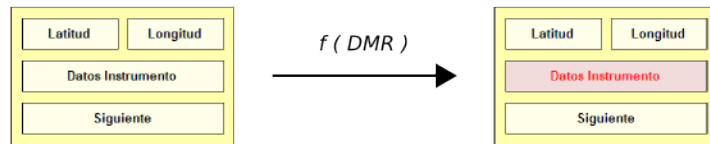
Este tipo de procesamiento, mostrado esquemáticamente en la Figura 4.11, posee una ventaja respecto al tipo de almacenamiento convencional, y es la posibilidad de decrementar el tiempo de procesamiento global al no emplear tiempo en analizar y tratar celdas DMF que no tienen medición. Aunque por otro lado, el hecho de comprobar para cada celda DMF si ésta está a *Null* o bien contiene medición requiere un pequeño tiempo de cómputo adicional.

Una posible mejora para evitar el tiempo empleado en analizar si una celda DMF está a *Null* o

1. Para cada celda DMF considerar todos los registros DMR en su interior



1.1. Aplicar procesamiento al registro DMR actual



2. Generación de la matriz DMF resultante

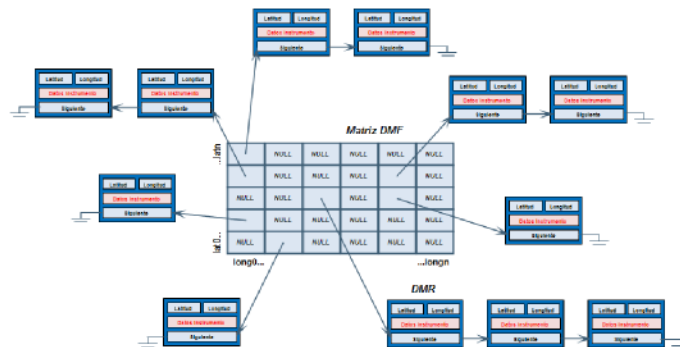


Figura 4.11: *Procesamiento de la matriz DMF mediante operaciones que no requieren vecindad.*

no, es construir una matriz DMF compuesta por una sola celda DMF (de tamaño 1×1) para tener todas las mediciones en una única lista, y así procesar una a una de manera secuencial. Sin embargo, el ámbito espacial se perdería de esta manera, por lo que sólo sería recomendable construir este tipo de matriz DMF cuando fuera a ser procesada con una metodología que no requiera vecindad. Una alternativa a esta solución, que requeriría una ligera modificación de la matriz DMF sería añadir una lista secuencial de punteros a todas los registros DMR almacenados en la matriz (*DMR List*, DMRL), tal y como se muestra en la Figura 4.12. De esta forma, se haría uso de la matriz DMF de manera convencional, y cuando se requiera procesar todas las mediciones mediante una metodología que no necesite vecindad, utilizar la lista adicional para recorrer de manera secuencial todas los registros DMR insertados en la matriz (tal y como de accesos directos se trataran). Sin embargo, esta optimización supondría un coste adicional al tener que almacenar las direcciones de memoria de cada uno de los registros DMR insertados en la matriz DMF.

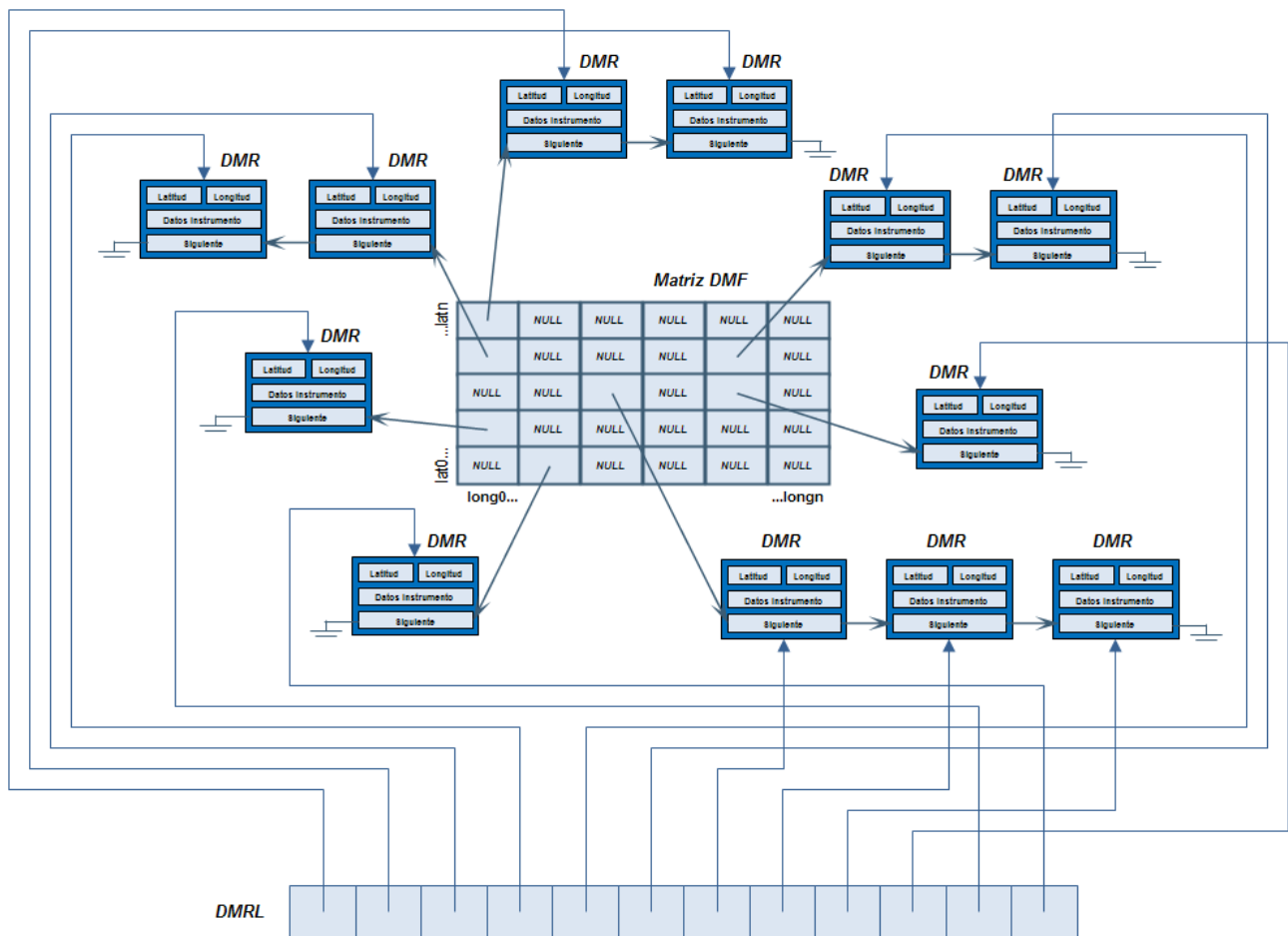


Figura 4.12: Optimización de la matriz DMF mediante la adición de una lista de accesos directos a todos los registros DMR almacenados.

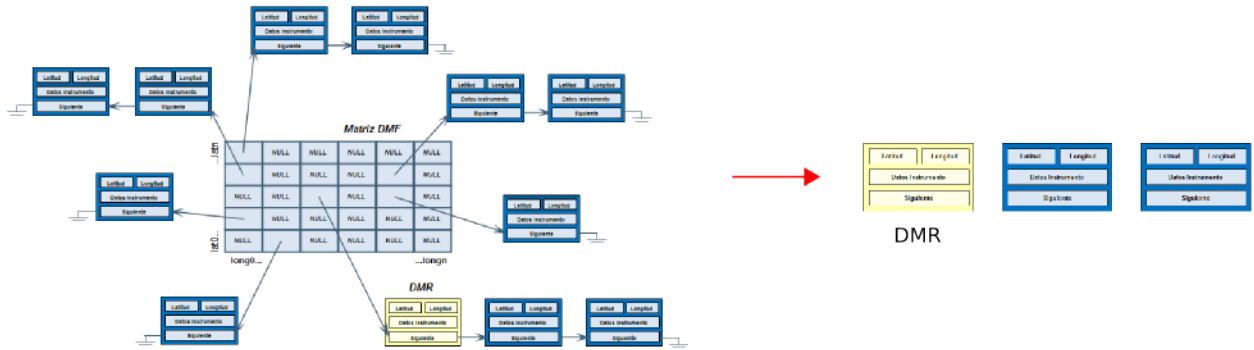
4.1.1.4.2. Operaciones que requieren vecindad.

Las operaciones que requieren vecindad son metodologías que procesan cada medición considerando las mediciones adyacentes. Este tipo de procesamiento puede involucrar varias técnicas de tratamiento de imágenes: filtros de mediana, máximo, mínimo y media, suavizados, detección de bordes, etc.

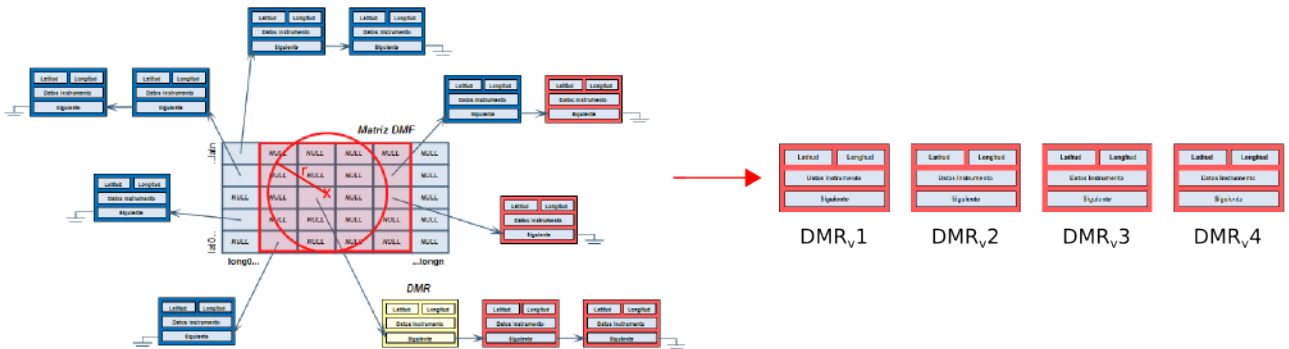
La manera de realizar este tipo de operaciones mediante una estructura matricial convencional es utilizando una ventana (también denominada *kernel*) de tamaño $n \times n$ píxeles que recorre toda la imagen, asignando al píxel central un valor procedente de aplicar una función a todos los píxeles que quedan dentro de la ventana. Para el caso de que la imagen esté representada mediante el formato DMF, el procesamiento es similar, tal y como se muestra en la Figura 4.13, pero en vez de utilizar una ventana, se hará uso de una distancia d . De esta manera, cada registro DMR almacenado será procesado y modificado acorde a la información de los demás registros DMR que se encuentren situados a una distancia menor a r del registro DMR actual.

La adaptación de este tipo de operaciones a formato DMF requiere un tiempo de cómputo mayor que su análogo en formato matricial convencional. Aunque por un lado existe un ligero ahorro al no procesar celdas DMF a *Null*, sí que existe un consumo de tiempo adicional al tener que comprobar si una celda DMF está a *Null* o no (incluyendo a la hora de buscar la vecindad de cada registro DMR), así como el tener que obtener todas las mediciones dentro de un determinado radio r (mediante el cálculo de la distancia euclídea) respecto al registro DMR referencia. El hecho de poseer una ventana de procesamiento convencional (como ocurre en los formatos matriciales convencionales) que permita el acceso directo de los píxeles adyacentes implica un ahorro en el tiempo de procesamiento global que

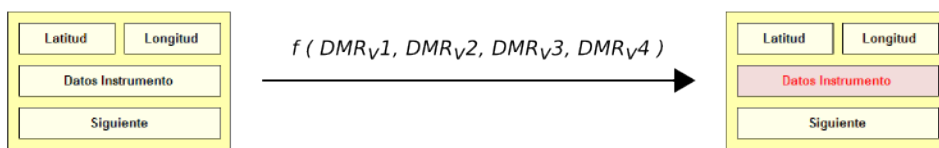
1. Para cada celda DMF considerar todos los registros DMR en su interior



1.1. Para cada registro DMR, considerar la vecindad de acuerdo a su localización espacial (x) y el radio de búsqueda (r)



1.2. Aplicar procesamiento al registro DMR actual



2. Generación de la matriz DMF resultante

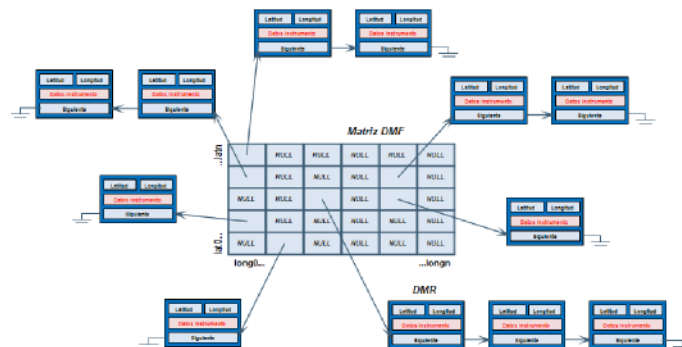


Figura 4.13: Procesamiento de la matriz DMF mediante operaciones que sí requieren vecindad.

no puede ser superado por el formato DMF. Sin embargo, la manera en la que los registros DMR están almacenados en las celdas DMF influirá directamente en el incremento de los tiempos de cómputo, por lo que la utilización de estructuras más complejas, como por ejemplos árboles (Kd-tree, Sección 4.1) pueden servir de ayuda a la hora de localizar, de manera más rápida, los registros DMR situados dentro del radio r , decrementando de esta manera los tiempos de ejecución globales.

4.1.2. *Adaptaciones y modificaciones del formato DMF.*

Las metodologías de los diferentes campos científicos basan su algoritmia en una representación de la imagen como una estructura matricial, la cual, en la mayoría de las ocasiones, está compuesta por tres dimensiones. Estas metodologías hacen uso de las operaciones matemáticas básicas comentadas con anterioridad (Sección 4.1.1.4), para las cuales dicha estructura matricial permite un procesamiento rápido y eficiente. Sin embargo, dichas metodologías suelen ser más complejas e implicar técnicas adicionales avanzadas, por lo que no siempre una estructura matricial es el tipo de datos más adecuado. Además, al tratarse de estructuras estáticas, la capacidad de adaptación a nuevas necesidades es bastante limitada.

El formato DMF, al estar basado en una unidad básica que representa la medición (registro DMR), la cual puede estar formada por diversos campos que aportan diferentes tipos de información, permite una completa adaptación a cualquier tipo de necesidad o cambio. Su adaptación es una tarea relativamente sencilla, pues todas las modificaciones se limitarán a la adición de campos al registro DMR, los cuales permitirán añadir nueva información acorde a los problemas existentes o las necesidades de las misiones.

Las siguientes secciones muestran adaptaciones y modificaciones del formato DMF para su aplicación en diferentes metodologías científicas, muy utilizadas e importantes hoy en día, como son la fusión de datos (Sección 4.1.2.1) y la detección de cambios (Sección 4.1.2.2).

4.1.2.1. *Adaptación del formato DMF para fusión de datos.*

Existen situaciones en las que la información adquirida por los sensores no se reducen a un espectro, sino que asociado a cada medición pueden existir datos adicionales, como pueden ser diferentes tipos de máscaras, señales de control, precisión del instrumento, monitorización de fenómenos, etc. Generalmente este tipo de información es proporcionada por un mismo instrumento, o por el sistema de procesamiento tras realizar las operaciones pertinentes a la medición. Siempre que esta información adicional esté asociada a una misma medición, la adaptación del formato DMF estaría reducida a la adición de nuevos campos que contemplen los datos correspondientes al registro DMR, tal y como se muestra en la Figura 4.14(a).

Por otro lado, existen casos en los cuales la información no proviene de un único sensor o sistema de procesamiento, sino de varios. Se tratarían de mediciones totalmente independientes, adquiridas en diferentes localizaciones espaciales y que almacenarían informaciones de diferente ámbito y tipo. La adaptación del formato DMF para considerar esta circunstancia, se realizaría mediante la adición de un nuevo campo en el registro DMR (*ID.Sensor*) que permita identificar el sensor que adquirió la medición, o bien la procedencia del mismo (Figura 4.14(b)). Por supuesto, este nuevo campo puede ir acompañado de otros que proporcionen información adicional a la medición, tal y como se comentó en el caso anterior, facilitando la mezcla de diferentes datos proporcionados por un mismo y por diferentes sensores.

Todas las mediciones adquiridas por los diferentes orígenes, en el formato DMR adaptado, pueden estar almacenadas en una misma matriz DMF (Figura 4.14(c)) [Pér10, Cri11, Cri12b]. De esta manera, conociendo las características de cada sensor o de cada origen, cada medición puede ser perfectamente identificada y procesada acorde a dichas características o a las diferentes necesidades.

El único problema existente a la hora de realizar esto, es que las geometrías de los sensores y las alturas de los vuelos impliquen trabajar con diferentes tamaños de píxeles. Al integrar todos los datos en la misma estructura, es necesario seleccionar un determinado tamaño para las celdas DMF. Respecto a esto, se pueden considerar diferentes políticas:

- Seleccionar como tamaño de celda DMF el tamaño de píxel mayor de las imágenes adquiridas por los diferentes sensores.
- Seleccionar como tamaño de celda DMF el tamaño de píxel menor de las imágenes adquiridas por los diferentes sensores.
- Seleccionar como tamaño de celda DMF el tamaño de píxel medio (u otro procedimiento estadístico) de las imágenes adquiridas por los diferentes sensores.
- Seleccionar como tamaño de celda DMF el tamaño de píxel de la imagen con más mediciones.
- Seleccionar un tamaño de celda DMF acorde a ciertos requisitos de utilización.

Esta adaptación proporciona la ventaja de disponer de todas las mediciones de interés almacenadas en la misma estructura de datos, en vez de en diferentes, haciendo mucho más cómodo el procesamiento, sobre todo si se requieren metodologías que impliquen análisis espaciales.

4.1.2.2. Adaptación del formato DMF para detección de cambios.

Una segunda posibilidad de adaptación del formato DMF sería para el estudio temporal de mediciones sobre una misma escena. En los sistemas de detección de cambios convencionales, se suelen utilizar diferentes estructuras basadas en imágenes por intervalo temporal [Bru02]. Como los datos provienen de diferentes vuelos, existen parámetros que complican la identificación de mediciones análogas y el análisis imagen-imagen de manera directa, como son la altura a la que fue adquirida la información (implicando imágenes de diferentes tamaños), o la dirección del vuelo (resultando imágenes con diferente rotación y la existencia de zonas fuera de solapamiento). A esto hay que sumarle los errores espaciales cometidos durante el proceso de geocorrección de las imágenes implicadas.

El hecho de poder almacenar todas las mediciones de intervalos diferentes de tiempo en una misma estructura de datos potencia esta comparación de la información de una manera directa, pues es fácil encontrar las mediciones análogas, así como detectar las zonas en las que no existe solapamiento entre las escenas capturadas.

De esta manera, la detección de cambios puede considerarse un caso especial de fusión de datos (Sección 4.1.2.1), en las cuales las mediciones son adquiridas por un único sensor (o varios) en diferentes instantes, y siendo el *tiempo* quien define el factor *origen*.

La adaptación del formato DMF, por tanto, se realizará, como en los casos anteriores, añadiendo un nuevo campo *Timestamp* al registro DMR (Figura 4.15(a)), que indicará el instante en que la medición correspondiente ha sido obtenida, pudiendo ser esta información proporcionada por el dispositivo GPS/INS o por el sistema de procesamiento. La manera en que se indica este carácter temporal puede llevarse a cabo de dos maneras diferentes:

- El campo *Timestamp* puede hacer referencia a un dato tipo *date* y *hour*, en el que para cada medición se almacena tanto la fecha (día, mes y año) como la hora (hora, minuto y segundo) en el que fue adquirida (por ejemplo: *Timestamp = 08/04/2012,15:59:23*). De esta manera, es prácticamente imposible encontrar dos mediciones con un mismo *timestamp*, sobre todo si se utiliza un único sensor. Con este formato, es posible realizar un procesamiento de la imagen por intervalos de tiempo, al estar el instante de adquisición de cada medición completamente identificado.

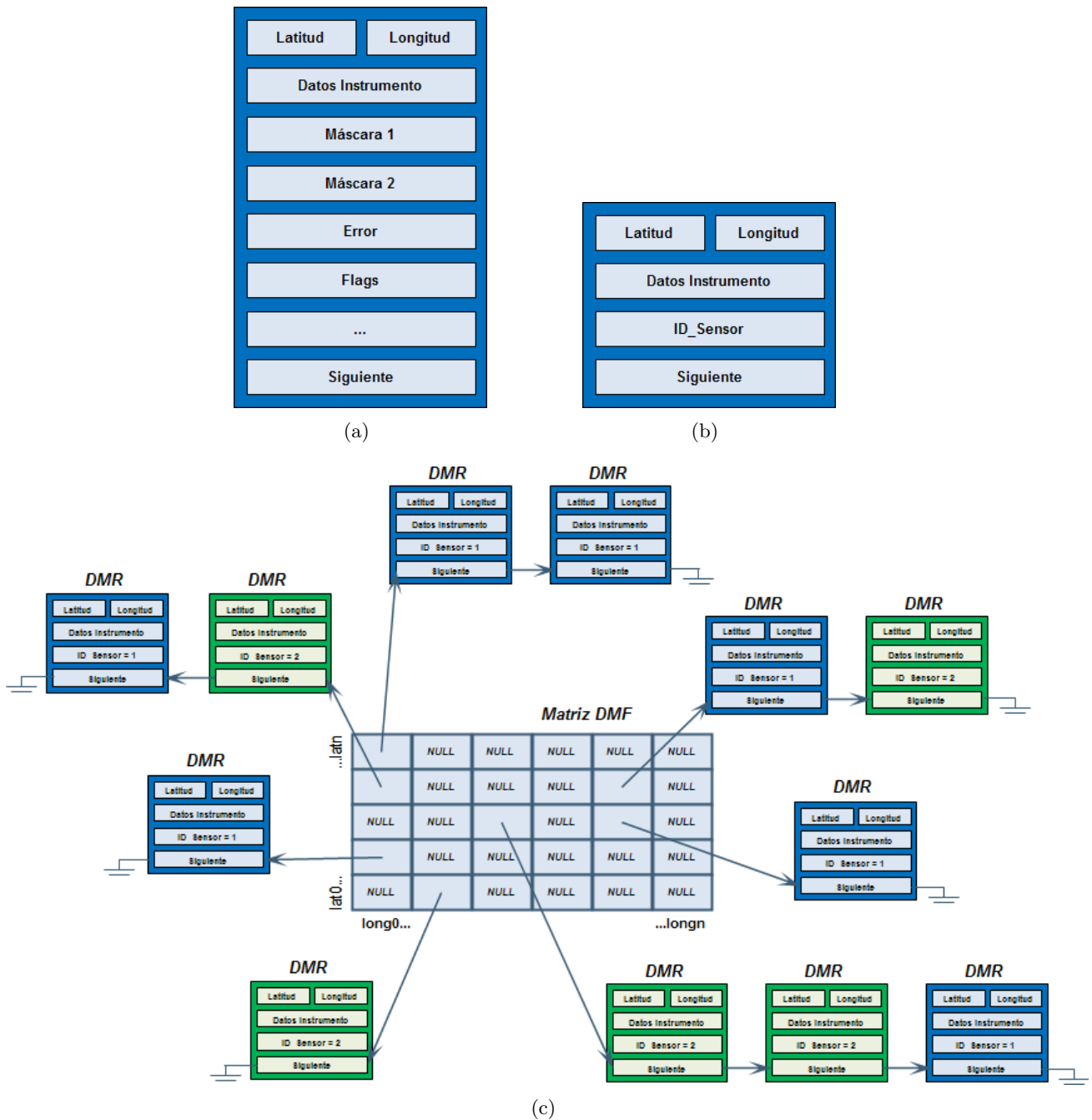


Figura 4.14: Adaptación del formato DMF para fusión de datos.

(a) Modificación del registro DMR para fusión de datos de mismo origen. (b) Modificación del registro DMR para fusión de datos de diferente origen. (c) Almacenamiento de registros DMR de diferentes orígenes en la misma matriz DMF.

- El campo *Timestamp* puede hacer referencia a valores predefinidos, en el que cada medición es identificada por una etiqueta de carácter temporal (por ejemplo: *Timestamp = Día1*). De esta manera, todas las imágenes adquiridas en un intervalo de tiempo definido (un vuelo, una hora, un mes, etc...), tendrán un mismo identificador. Este formato dificulta la posibilidad de realizar un procesamiento de la imagen por intervalos de tiempo, ya que no se sabe con exactitud en qué momento fue adquirida cada medición, pero resulta muy útil si se trabaja con series de imágenes bien diferenciadas en el tiempo, donde el momento de adquisición no requiere gran exactitud ni precisión.

Igual que ocurría con la adaptación para fusión de datos, todas los registros DMR modificados

serán insertados en la misma matriz DMF (Figura 4.15(b)), por lo que todas las mediciones serán almacenadas en la misma estructura de datos, siendo cada una de ellas identificada temporalmente por el campo *Timestamp* [Cri12c].

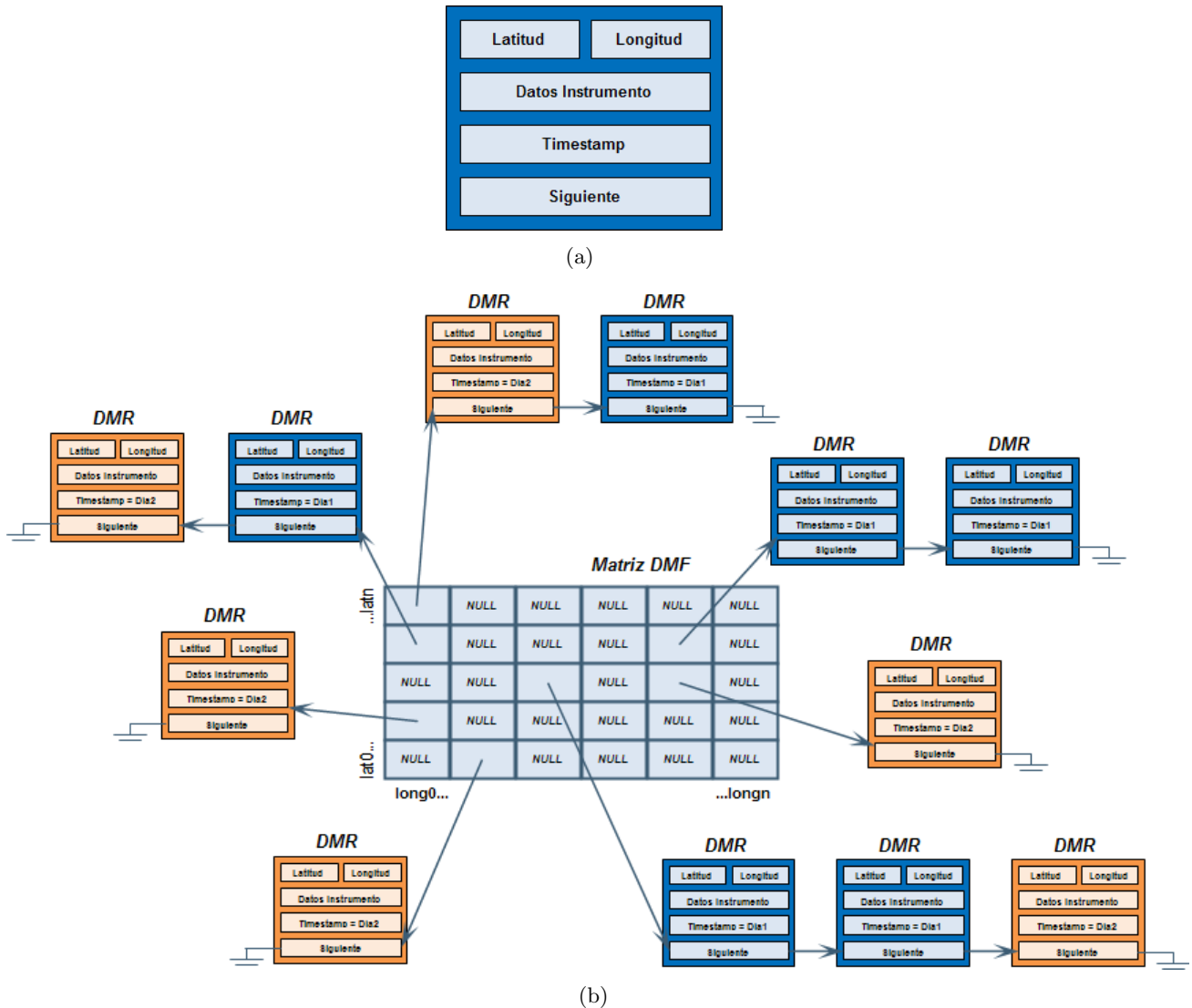


Figura 4.15: Adaptación del formato DMF para detección de cambios.

(a) Modificación del registro DMR para detección de cambios. (b) Almacenamiento de registros DMR de diferentes intervalos temporales en la misma matriz DMF.

4.2. Simulación de un sistema SpaceCube.

El formato DMF presenta varias ventajas y características que la convierten en una gran alternativa a los formatos de representación de información convencionales. Sin embargo, sus funcionalidades se potencian cuando es utilizada a bordo de la nave o satélite, proporcionando un cómodo, fácil y eficaz procesamiento de los datos a medida que se van obteniendo. Además, su modo de almacenamiento (Sección 4.1.1.3) permite una rápida distribución de los resultados obtenidos a bordo, estando disponibles (y georreferenciados) nada más aterrizar el avión, o bien recibida completamente la transmisión en el caso de satélites. Sin embargo, es posible que los resultados obtenidos a bordo necesiten ser refinados, o incluso completados con productos existentes en Tierra, por lo que un procesamiento previo a bordo puede implicar varias ventajas:

- Debido al enorme tiempo dedicado a generar unos resultados definitivos disponibles a la

comunidad científica (del orden de horas e incluso días) causado por la gran cantidad de datos adquiridos y por la complejidad de la cadena de procesamiento, puede suponer un ahorro considerable en el tiempo de cómputo global y una agilización del proceso total.

- Los resultados obtenidos pueden ser interpretados como un avance o *preview* de los resultados definitivos, de tal manera que el usuario puede conocer a priori si los resultados definitivos le serán útiles o no.
- Los resultados obtenidos pueden ser considerados como un tipo de producto a bajo nivel, que los usuarios pueden obtener a sabiendas de que trabajarán con un determinado margen de error.

En esta sección se propone la simulación de un sistema de procesamiento a bordo que utiliza el formato DMF como estructura de datos básica como resultado final. Dicho sistema de procesamiento está basado en la arquitectura hardware SpaceCube, presentada en la Sección 3.3.2.1, ya que una arquitectura reconfigurable de tales características puede ser muy útil si se utiliza como unidad de procesamiento general, o, como es el caso, como sistema auxiliar. De esta manera, el sistema SpaceCube puede ayudar al computador a bordo de diferentes maneras, como puede ser realizando a un nivel de hardware operaciones básicas muy costosas (y liberando la carga del computador a bordo), o bien generando diferentes resultados de manera paralela.

El contexto en el que se situaría la propuesta de sistema de procesamiento a bordo se muestra en la Figura 4.16. Un sistema de adquisición de datos, el cual puede ser un avión, o bien un satélite, está formado por infinidad de componentes y módulos que permiten su correcto funcionamiento, sin embargo, en dicha figura únicamente se muestran aquéllos que son imprescindibles para el correcto funcionamiento del sistema propuesto. Por tanto, algunos elementos, como los subsistemas mecánicos, o los componentes de gestión y distribución de la energía [Spe07], imprescindibles en un sistema de adquisición convencional, no han sido añadidos a la figura.

Todos los elementos que intervienen tanto en un avión como en un satélite han sido representados con líneas continuas, mientras que los representados con líneas punteadas se tratan de elementos opcionales que pueden utilizarse o no, o bien son propios de un único tipo de sistema de adquisición. Todos los dispositivos interactúan con el sistema de procesamiento global de la nave, enviando y recibiendo datos e información de control. Este hecho se representa con una flecha grande azul bidireccional. Las flechas pequeñas unidireccionales indican flujos de datos entre dispositivos. Si la línea de la flecha es continua, esta comunicación es directa (la conexión entre los dispositivos es directa). Si es discontinua, esta comunicación puede ser directa, o bien indirecta (utilizando al sistema de procesamiento global como intermediario).

Los elementos que intervienen en el sistema de procesamiento se detallan a continuación:

- *Sistema de posicionamiento*: Sistema que realiza una monitorización de la posición y trayectoria de la nave en tiempo real. Para ello, hace uso de dos dispositivos:
 - *Sistema de Navegación Inercial (INS)*: Dispositivo compuesto por la Unidad de Medición Inercial (IMU), un pequeño computador que, junto con acelerómetros (sensores de movimiento) y giroscopios (sensores de rotación), calcula de manera continua la posición, orientación y velocidad de la nave.
 - *Sistema de Posicionamiento Global (GPS)*: Dispositivo que permite establecer la posición de la nave mediante la utilización de redes de satélites dedicados a dicho propósito. Suele utilizarse en plataformas a bordo de aviones, para poder refinar, y hacer más precisa, la posición calculada por el dispositivo INS.
- *Instrumentos*: Dispositivos transportados por la nave, encargados de adquirir las mediciones. Pueden tratarse de telescopios, espectrógrafos, sensores, cámaras, sistemas radar, etc. Los datos

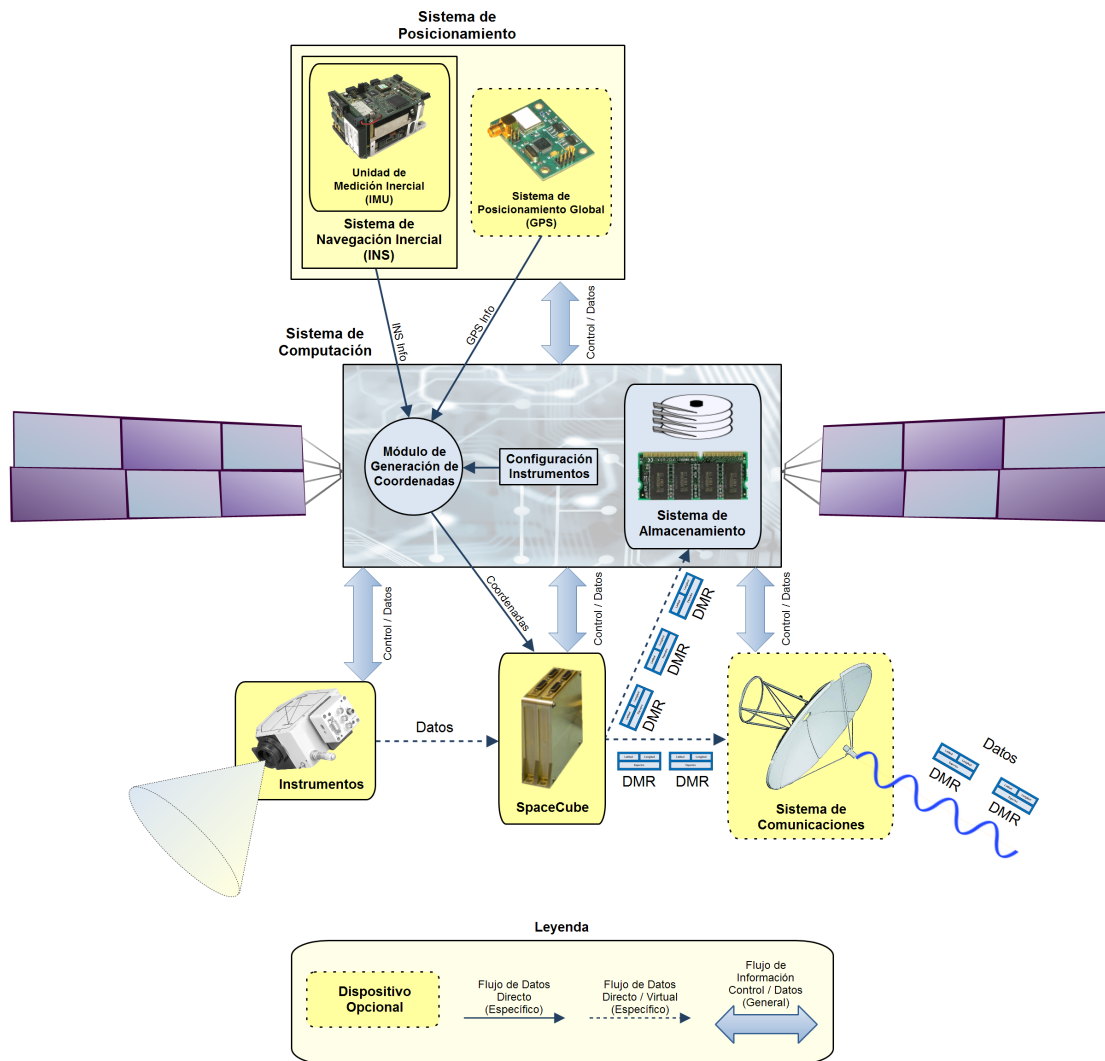


Figura 4.16: Contexto del sistema de procesamiento a bordo propuesto.

recogidos por los instrumentos son enviados directamente al Sistema de Computación para su procesamiento, almacenamiento o transmisión.

- *Sistema de Comunicaciones*: Son más frecuentes en los satélites, y establecen las comunicaciones con Tierra, enviando y recibiendo señales de control de manera continua, y transmitiendo información procedente del Sistema de Computación, incluyendo las mediciones adquiridas por los instrumentos (tras ser procesados, si se requiere).
- *Sistema de Computación*: Se trata del *cerebro* del sistema, y está basado en uno o más computadores que permiten realizar las operaciones para las que fue diseñado. Su complejidad y densidad modular depende de los requisitos de la misión. Como puede estar compuesto por cientos de sistemas y subsistemas, en la figura se han representado, de manera sencilla y básica, los necesarios para al correcta ejecución del sistema propuesto:
 - *Sistema de Almacenamiento*: Sistema de memorias que permiten almacenar datos de configuración de la nave, variables de las aplicaciones ejecutadas, o datos de diferentes procedencias, y pueden ser tanto internas en el Sistema de Procesamiento, como externas. Entre otras funciones, pueden utilizarse para almacenar de manera parcial las mediciones adquiridas por los sensores (pudiendo incluir resultados de diferentes procesados), o bien completamente, por lo que se requerirían memorias de mayor capacidad. Éste último caso es más representativo de los sistemas de adquisición a bordo de aviones, en los cuales los datos capturados se extraen de la memoria al finalizar el vuelo.

- *Configuración de los Instrumentos*: Pequeño bloque de memoria (en Memoria Principal o independiente) que almacena las características de los instrumentos a bordo, como tipo de escáner, geometría, campo de visión (FOV), cobertura y *sampleado* espectrales, distancia focal, etc. Estos datos son útiles para poder georreferenciar cada medición adquirida utilizando los datos del Sistema de Posicionamiento.
 - *Módulo de Generación de Coordenadas*: A partir de la información proporcionada por el Sistema de Posicionamiento y de la configuración de los instrumentos a bordo, determina la localización de cada una de las mediciones adquiridas. De esta manera, en tiempo real, y a bordo de la nave, se determina la posición de cada dato, siendo posible realizar operaciones que requieran dimensionalidad espacial.
- *SpaceCube*: Propuesta de dispositivo hardware reconfigurable. Como puede observarse, este dispositivo forma parte del *payload* del satélite y funcionaría como un sistema de computación independiente (aunque también podría conformar el sistema computación global del satélite). De esta manera, es posible considerarlo como un módulo hardware auxiliar que permite ayudar al sistema de computación global en la ejecución de operaciones costosas, aligerándolo de carga, o bien obtener diferentes resultados de manera independiente. Para ello, recibiría los datos de los instrumentos, bien directamente, o a través del Sistema de Computación, junto con sus coordenadas, previamente calculadas por el Módulo de Generación de Coordenadas. Tanto los datos instrumentales, como las coordenadas, se obtienen por canales de información diferentes, por lo que es posible que la recepción de los mismos no se realice de manera ordenada (dato instrumental - coordenada), sino que es posible que se produzca a través de ráfagas irregulares de datos instrumentales y coordenadas. Por tanto, ha de existir un módulo interno que permita controlar esta circunstancia. Los resultados obtenidos por el sistema SpaceCube, que pueden incluir un procesamiento de los datos originales, son remitidos al Sistema de Computación para un refinamiento de los mismos, un almacenamiento, o una transmisión a Tierra. La estructura de datos básica para realizar esta transferencia sería el registro DMR (Figura 4.17(a)), el cual estaría compuesto por diferentes campos como las coordenadas de la medición, la información original obtenida por los sensores, los resultados del procesamiento, y otros campos derivados de la adaptación de técnicas como la fusión de datos (Sección 4.1.2.1) o la detección de cambios (Sección 4.1.2.2).

El dispositivo hardware reconfigurable propuesto fue elaborado utilizando dos tarjetas FPGA Xilinx Virtex-5 ML507 y considerando como base principal el sistema SpaceCube original diseñado por la NASA. Dicho sistema está compuesto originalmente por tarjetas FPGA Xilinx Virtex-5 ML510, una versión posterior a la Virtex-5 ML507, aunque ésta última suele ser utilizada para testeo de las metodologías debido a su inferior coste. Aunque las diferencias entre ambas versiones son múltiples [Esp09], para el caso del dispositivo hardware reconfigurable propuesto es necesario considerar y tener en cuenta las presentadas en la Tabla 4.1, principalmente.

		Virtex-5 ML507	Virtex-5 ML510
	<i>Slices</i>	11200	20480
Recursos Lógicos	<i>Celdas Lógicas</i>	71680	131072
	<i>Flip-Flops CLB</i>	44800	81920
	<i>Slices DSP48E</i>	128	320
	Recursos Embebidos	<i>Procesadores PowerPC</i>	1
Conectores	<i>DE-9</i>	2	2
	<i>RJ-45</i>	1	2
	<i>XGI</i>	2	0
	<i>PM</i>	0	2

Tabla 4.1: Diferencias principales entre las tarjetas Xilinx Virtex-5 ML507 y Virtex-5 ML510, considerando el sistema hardware reconfigurable propuesto.

Ambas tarjetas FPGA estarán interconectadas (Figura 4.17(b)), emulando de esta manera un *slice* de procesamiento del sistema SpaceCube original. Estas tarjetas pueden trabajar en paralelo para, de esta manera, obtener diferentes resultados de un mismo conjunto de datos, o bien colaborar entre ellas para conseguir un objetivo común.

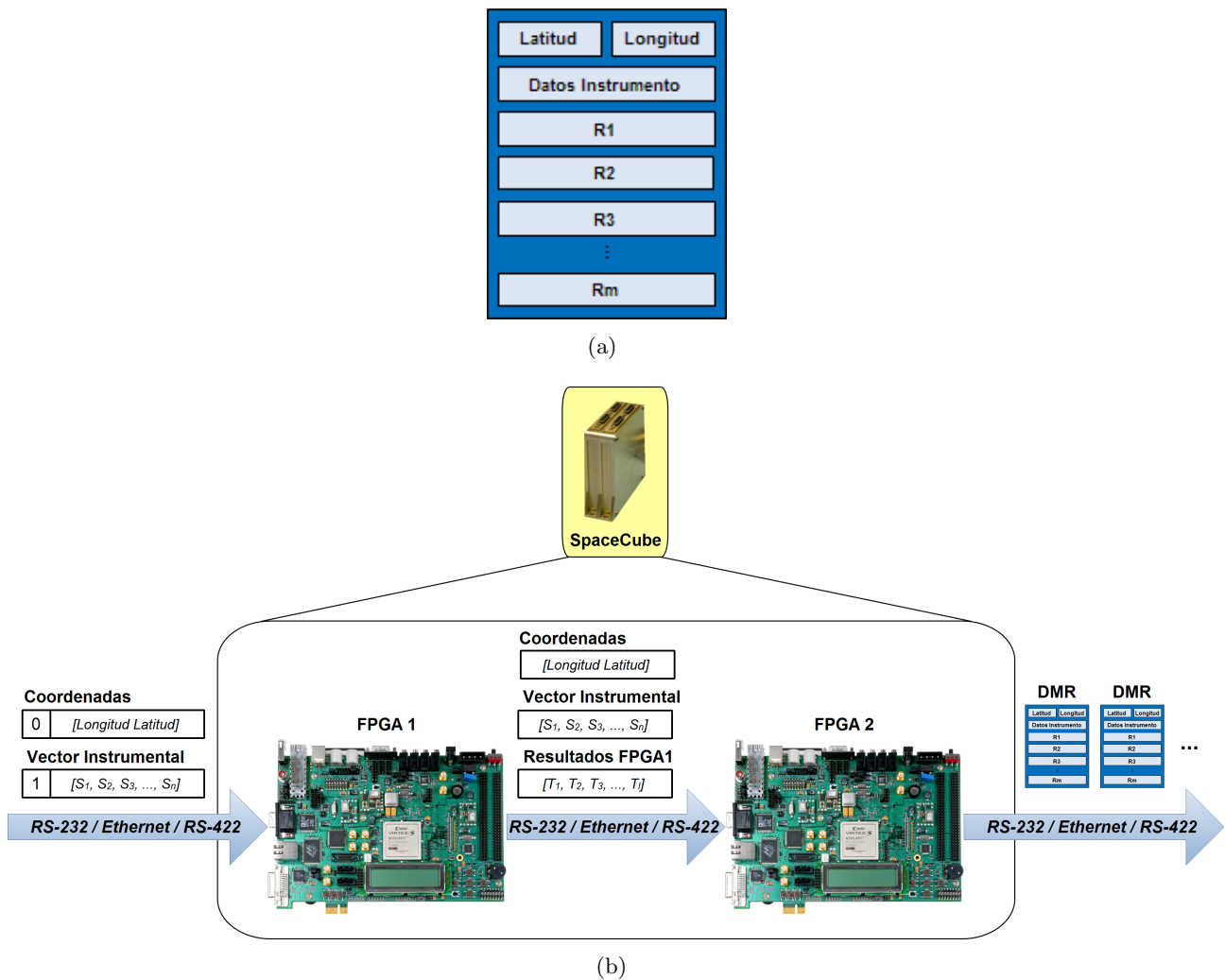


Figura 4.17: Emulación de un *slice* de procesamiento SpaceCube.

(a) Registro DMR resultante del procesamiento de la segunda tarjeta FPGA. (b) Conexión de las dos tarjetas FPGA.

Xilinx ofrece en sus tarjetas FPGA diferentes tipos de conexiones con el exterior, con diferentes protocolos y velocidades de transmisión, que pueden variar según la versión de la tarjeta FPGA utilizada. Para el caso de Virtex-5, y concretamente para el desarrollo del sistema propuesto, se consideraron las siguientes conexiones:

- *Conexiones de baja velocidad:* Permiten la comunicación con el exterior mediante los conectores DE-9, utilizando el protocolo RS-232 [EIA69, TIA97], a través de un interfaz *Universal Asynchronous Receiver / Transmitter* (UART) [Nor05]. El fabricante ofrece el módulo UartLite [Xil09d], que implementa este interfaz con diferentes velocidades y configuraciones. La unidad básica de transmisión del protocolo RS-232 consiste en un paquete de formato adaptable (según la sincronización y el nivel de control de errores deseados) y que tanto emisor como destinatario deben conocer. Para la elaboración de los experimentos, se consideró el tipo de paquete estándar conocido como 8N1, mostrado en la Figura 4.18, y en donde cada uno de los campos es detallado a continuación:
 - *START:* Bit con valor a 1 que indica el comienzo de paquete.

- *Datos*: Datos reales a transmitir.
- *STOP*: Bit con valor a 0 que indica el fin de paquete.

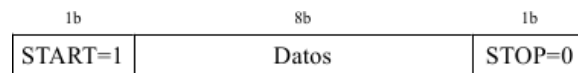


Figura 4.18: Configuración del paquete RS-232 8N1.

Las velocidades de transmisión proporcionadas mediante este tipo de conexión (Baud Rate), incluyendo la referente a los datos reales en Bytes por segundo (B/s) (de acuerdo a la configuración utilizada en los experimentos) se detallan a continuación:

- *110 Baud*: 11 B/s (datos reales).
 - *300 Baud*: 30 B/s (datos reales).
 - *600 Baud*: 60 B/s (datos reales).
 - *1200 Baud*: 120 B/s (datos reales).
 - *2400 Baud*: 240 B/s (datos reales).
 - *4800 Baud*: 480 B/s (datos reales).
 - *9600 Baud*: 960 B/s (datos reales).
 - *14400 Baud*: 1440 B/s = 1.406 KB/s (datos reales).
 - *19200 Baud*: 1920 B/s = 1.875 KB/s (datos reales).
 - *38400 Baud*: 3840 B/s = 3.75 KB/s (datos reales).
 - *57600 Baud*: 5760 B/s = 5.625 KB/s (datos reales).
 - *115200 Baud*: 11520 B/s = 11.25 KB/s (datos reales).
 - *128000 Baud*: 12800 B/s = 12.5 KB/s (datos reales).
 - *230400 Baud*: 23040 B/s = 22.5 KB/s (datos reales).
 - *460800 Baud*: 46080 B/s = 45 KB/s (datos reales).
 - *921600 Baud*: 92160 B/s = 90 KB/s (datos reales).
- *Conexiones de alta velocidad*: Permiten la comunicación con el exterior mediante conectores y protocolos de alta velocidad. Xilinx ofrece dos tipos diferentes:
- *Ethernet*: Permiten la comunicación con el exterior mediante los conectores RJ-45, utilizando el protocolo Ethernet [Sim08]. Xilinx ofrece los módulos Ethernet Lite MAC [Xil10b], que proporciona una velocidad de transmisión máxima de hasta 100 Mbps; y Tri Mode Ethernet MAC [Xil11e] (requiere licencia adicional), que proporciona una velocidad de transmisión máxima de hasta 1000 Mbps. El paquete utilizado para realizar las transferencias de datos se muestra en la Figura 4.19, en donde cada uno de los campos es detallado a continuación:
 - *Preámbulo*: Cabecera que permite la sincronización de los paquetes. Consta de un total de 7 bytes, en donde cada byte contiene el patrón “10101010”.
 - *Start of Frame Delimiter (SFD)*: Campo que indica el comienzo del paquete. Consta de un único byte que contiene el patrón “10101011”.
 - *Destino*: Dirección física Ethernet MAC del sistema de computación destino.
 - *Origen*: Dirección física Ethernet MAC del sistema de computación origen (tarjeta FPGA).
 - *Longitud*: Campo que indica la longitud en bytes del campo Datos (puesto que el tamaño de éste es variable).
 - *Datos*: Datos reales a transmitir.

- *Frame Check Sequence (FCS)*: Campo calculado sobre los datos almacenados en el paquete para control de errores. Utiliza un algoritmo Cyclic Redundancy Check (CRC) de 32 bits (CRC-32) [Sob03], cuyo polinomio generador (G) es definido mediante la Ecuación 4.7.

$$G(x) = x_{32} + x_{26} + x_{23} + x_{22} + x_{16} + x_{12} + x_{11} + x_{10} + x_8 + x_7 + x_5 + x_4 + x_2 + x_1 + x_0 \quad (4.7)$$

7B	1B	6B	6B	2B	0B - 1500B	4B
Preámbulo	SFD	Destino	Origen	Longitud	Datos	FCS

Figura 4.19: Configuración del paquete Ethernet.

Las velocidades de transmisión proporcionadas mediante este tipo de conexión (Mbps) se detallan a continuación. Como con este protocolo el tamaño de paquete es variable, pues depende del tamaño del campo de datos, la velocidad de transmisión de datos reales es variable también. Considerando un tamaño de campo de datos máximo (1500 Bytes), las velocidades de transmisión proporcionadas mediante este tipo de conexión, incluyendo la mínima velocidad referente a los datos reales en Bytes por segundo (B/s), se detallan a continuación ¹:

- *10 Mbps*: 1228702.49 B/s = 1.172 MB/s (datos reales).
- *100 Mbps*: 12287024.9 B/s = 11.718 MB/s (datos reales).
- *1000 Mbps*: 122870249 B/s = 117.178 MB/s (datos reales).

Por supuesto, cuanto más pequeño sea el campo de datos del paquete Ethernet, mayor será la velocidad de transmisión de datos reales proporcionada.

- *RS-422*: Permiten la comunicación con el exterior mediante los conectores DB-25, utilizando el protocolo RS-422 [TIA94]. Su utilización puede llevarse a cabo, igual que con la comunicación RS-232, a través del módulo Ethernet Lite MAC, proporcionando una velocidad máxima de transmisión de hasta 10 Mbps. El conector DB-25 no está disponible de manera física en las tarjetas Virtex-5 ML507 y ML510, por lo que es necesario utilizar una tarjeta adicional con este tipo de conectores. Esta tarjeta hará las funciones de conexión puente entre los datos de entrada (proporcionadas por una computadora o instrumento) y la tarjeta FPGA. La conexión entre la tarjeta puente y la tarjeta FPGA se realiza a través de los conectores Xilinx General Interface (XGI), en la versión ML507, y los dos conectores Personality Modules (PM) en la versión ML510. Estos conectores ofrecen pines LVDS para transferencia de datos con una velocidad de hasta 3 Gbps. Sin embargo, como en el envío y en la recepción de los datos se utilizan también conectores DB-25 mediante el protocolo RS-422, la velocidad de transmisión de los datos se limitará a la de este último tipo de conexión: 10 Mbps.

El formato de paquete para este protocolo, al usar interfaces UART, es el mismo que para el protocolo RS-232 (Figura 4.18). En este formato, también es posible utilizar campos adicionales como el segundo bit de parada o el bit de paridad, sin embargo, para el análisis realizado en los experimentos, sólo se consideraron un bit de inicio y un único bit de parada, con 8 bits de datos.

Por lo tanto, la velocidad de transmisión proporcionada mediante este tipo de conexión (Mbps), incluyendo la referente a los datos reales en Bytes por segundo (B/s) (de acuerdo a la configuración utilizada en los experimentos) se detallan a continuación:

- *10 Mbps*: 1000000 B/s = 976.562 KB/s (datos reales).

Teniendo en cuenta el contexto en el que se situará el dispositivo SpaceCube propuesto (Figura 4.16), éste recibirá, principalmente, dos tipos de datos diferentes: las mediciones adquiridas por los

¹Unidades caudales de transmisión 1 Kbps = 1000 bps; 1 Mbps = 1000000 bps.

instrumentos (datos instrumentales), y sus respectivas coordenadas. Estas dos comunicaciones podrían realizarse o bien mediante un único puerto, o bien mediante el uso de dos puertos o conectores diferentes, cada uno dedicado a cada tipo de caso. En ambos casos, es necesario identificar cada uno de los dos tipos de datos para un correcto funcionamiento del sistema. Como en el diseño propuesto únicamente existe un puerto de entrada, ambos tipos de datos utilizarán el mismo conector. Por tanto, cada vez que recibamos un dato, deberemos identificar si se trata de un dato instrumental, o de coordenadas. Para ello, se ha optado por preceder cada tipo de dato de un identificador, siendo el 0 el correspondiente a la información de coordenada, y el 1 a la información instrumental. Este identificador puede ser introducido por el Sistema de Computación antes de enviar la información al dispositivo SpaceCube.

Además, hay que tener en cuenta el orden en el que va a llegar la información al puerto de la primera tarjeta FPGA. Generalmente, en un sistema de adquisición, cada vez que una determinada medición es adquirida, el instrumento genera un evento que es recogido por el Sistema de Computación, junto con los datos correspondientes [NAS09c, Ngu08, Xu08]. Por tanto, éste puede asignar un determinado *timestamp* por cada medición recibida, así como la determinación de su localización, por parte del Módulo de Generación de Coordenadas, a partir de la posición del satélite o avión (proporcionada por el Sistema de Posicionamiento) y la configuración de los instrumentos. Esto permite una mejor sincronización de la información recogida por los instrumentos garantizando ráfagas dato instrumental - coordenadas de manera ordenada. Pero es posible que por motivos diversos esto no sea así, y que se reciban varios datos instrumentales juntos, y después todas sus coordenadas (o viceversa). Para el desarrollo del sistema propuesto se ha estimado que esto puede ocurrir, pero siempre considerando que la obtención de una medición y sus coordenadas se realizan de manera sincronizada antes de ser enviada al dispositivo SpaceCube. Es decir, si se obtiene una ráfaga de n mediciones $[m_1, m_2, \dots, m_n]$, entonces la siguiente ráfaga de coordenadas corresponderá a esas mediciones en el mismo orden $[c_1, c_2, \dots, c_n]$, siempre siguiendo el orden de adquisición. De esta forma, nunca ocurrirá un caso en el que, si una medición ha llegado antes que otra, sus coordenadas lleguen en el orden inverso. No obstante, si esto ocurriera, bastaría con que el dispositivo SpaceCube buscara cada medición con su correspondiente coordenada de acuerdo al *timestamp* y sin importar el orden de llegada.

La manera en la que se controlan las continuas ráfagas de datos instrumentales y coordenadas (con las consideraciones previamente detalladas) sería mediante la adición de FIFOs separadas para los dos tipos de datos (valor instrumental y de coordenada). De esta manera, la información siempre quedará almacenada de forma ordenada sin importar las posibles combinaciones de ráfagas.

Por otro lado, en cuanto a la salida de esta primera tarjeta FPGA, ésta estará compuesta por los resultados de procesamiento (pueden incluir los datos originales), junto con las coordenadas correspondientes.

Los datos en la segunda tarjeta FPGA entrarán siempre en orden ya que saldrán de esta manera de la primera tarjeta. Pero en caso de que no sea así por cualquier circunstancia, la recepción de la información se almacenará también mediante FIFOs. La salida se hará proporcionando el resultado de ambas tarjetas (pueden incluir datos originales) junto con sus coordenadas, por lo que puede ser considerada como un tipo de dato DMR con múltiples campos. Como en este caso es el sistema propuesto quien realiza los envíos de la primera tarjeta FPGA a la segunda, se controlarán que la transferencia de datos se realice de manera organizada, por lo que en este caso no se utilizan identificadores de tipo antes de cada dato.

A ambas tarjetas FPGA les han sido programadas un sistema hardware que permite el procesamiento de la información, tal y como se muestra en las miniaturas de la Figura 4.17(b), y cuyo diseño, para un caso general, se presenta de manera detalla en la Figura 4.20.

El sistema hardware está compuesto por los siguientes elementos:

- *Procesador PowerPC a 400 MHz*: Un procesador que ejecuta directamente las aplicaciones

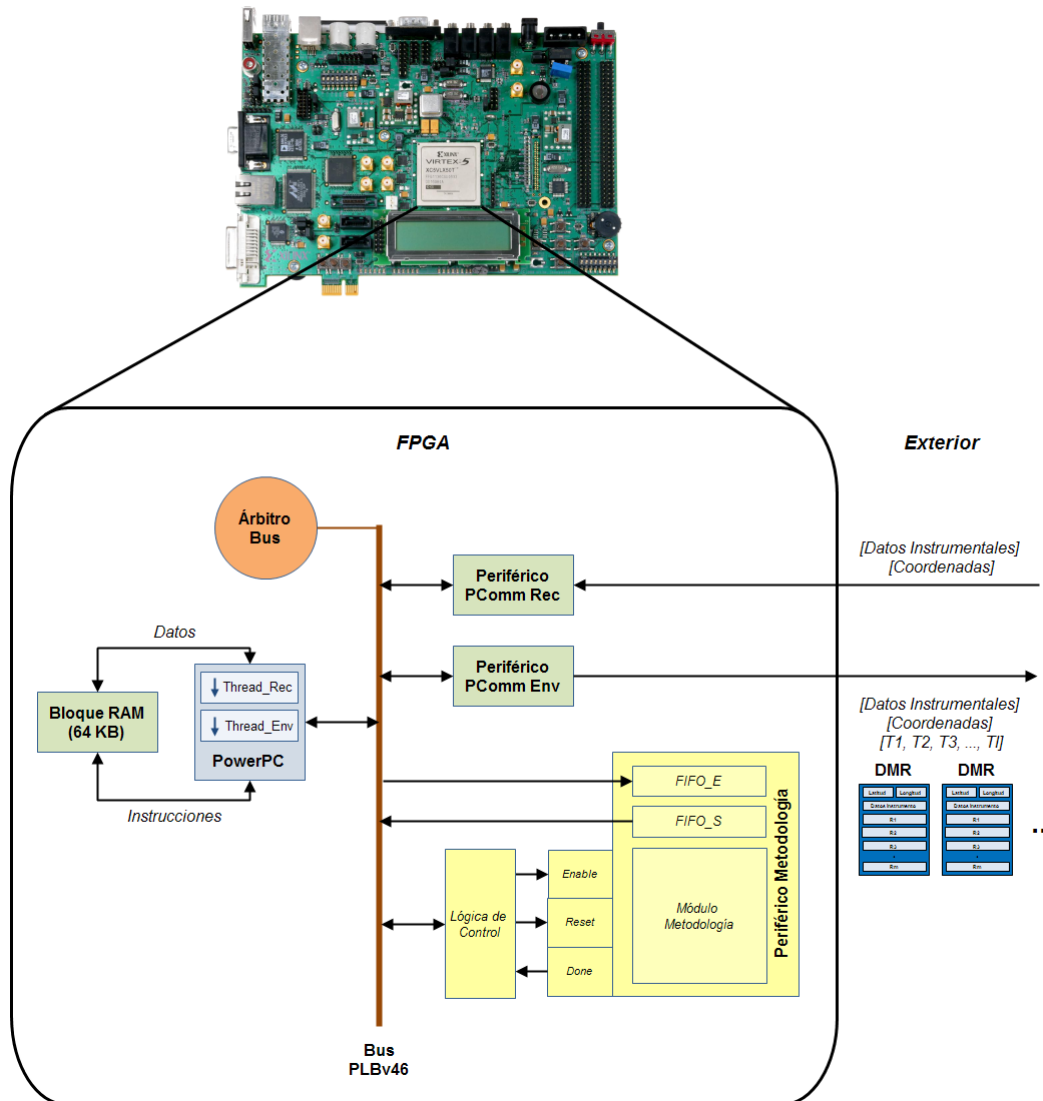


Figura 4.20: Diagrama genérico del sistema hardware implementado en las tarjetas FPGA.

software. Éste puede realizar aplicaciones más o menos complejas de acuerdo a las necesidades de la misión, pero para la propuesta el procesador sólo se encargará de configurar e inicializar los circuitos, así como de gestionar las comunicaciones entre los periféricos. Como sistema operativo ejecuta XilKernel [Xil06], un sistema operativo (SO) ligero orientado a sistemas embebidos y que proporciona diferentes funcionalidades, como por ejemplo la gestión de hilos. De esta manera, sobre este procesador se ejecutarán dos hilos: El primero de ellos, *Thread_Rec* permitirá gestionar los datos procedentes del exterior de la FPGA (comunicaciones entrantes), mientras que el segundo hilo, *Thread_Env*, permitirá gestionar los datos a enviar al exterior de la FPGA (comunicaciones salientes). En ambos casos, el procesador ha de controlar, mediante el identificador detallado con anterioridad, si el tipo de datos es información de medición o coordenada.

- *Bloque RAM de 64 KB*: Bloque de memoria reservado al procesador, y en donde se encuentran almacenadas las instrucciones del SO, y las aplicaciones del usuario junto con las variables y constantes definidas a nivel de software.
- *Bus central*: Permite la comunicación entre los diferentes periféricos, aunque éstas siempre se realizan a través del procesador.
- *Árbitro del bus*: Regula las comunicaciones del bus, estableciendo un orden de prioridades en caso de que dos o más periféricos decidan utilizar el bus en el mismo instante.

- *Periféricos*: Componentes hardware que realizan una determinada función. Para el sistema hardware propuesto, se utilizarán dos tipos de componentes diferentes:
 - *Periférico de Comunicaciones (PComm)*: Componente hardware que controla las conexiones a través de los diferentes conectores. Se utilizarán siempre dos, uno que se encarga de las comunicaciones entrantes (*PComm Rec*), recibíendolas del exterior y enviándoselas al procesador; y otro que se encarga de las comunicaciones salientes (*PComm Env*), recibíendolas del procesador y enviándolas al exterior. Los datos entrantes y salientes corresponderán, o bien a información de medición, o a coordenadas. Sin embargo, para el caso de la segunda tarjeta FPGA, los datos salientes serán considerados directamente como registros DMR. Como se comentó con anterioridad, Xilinx ya ofrece interfaces para los tipos de conexión que se deseen utilizar, y que serán añadidas al sistema como periféricos de comunicaciones:
 - *RS-232*: Interfaz Uart Lite.
 - *Ethernet*: Interfaces Ethernet Lite MAC y Tri Mode Ethernet MAC (esta última interfaz para velocidades superiores a 100 Mbps).
 - *RS-422*: Interfaz Uart Lite.
 - *Metodología*: Este periférico implementa el módulo de la metodología a ejecutar a bordo, y pueden existir tantos como se necesiten. Está formada por una Lógica de Control y una serie de registros que representan la interfaz mediante la cual se establecen las comunicaciones con el procesador. En su interior se encuentran definidos los módulos *FIFO_E* y *FIFO_S*, que permiten almacenar en memorias FIFO los datos entrantes y los resultados del procesamiento, respectivamente.

De esta manera, el procedimiento general mediante el cual se realiza el procesamiento a bordo mediante el sistema propuesto, se detalla a continuación:

1. Un dato llega al puerto y es recogido por el periférico *PComm Rec*. Éste inmediatamente se lo envía al procesador, concretamente al hilo *Thread_Rec*.
2. El hilo *Thread_Rec* recibe el dato de *PComm Rec* y lo identifica como información de medición o como coordenada. A partir de esta información, se lo envía al módulo Metodología para su procesamiento, almacenándolo en memorias FIFO diferentes de acuerdo al tipo de dato.
3. Cuando el módulo Metodología haya recibido correctamente al menos una medición con su coordenada, entonces realizará la lectura del dato de las memorias FIFO de entrada (*FIFO_E*) y comenzará su procesamiento. El número de ciclos de reloj que conlleva el procesamiento está directamente relacionado con la metodología y la manera en la que ésta ha sido implementada. Una vez el procesamiento ha sido finalizado, el resultado se almacena en las memorias FIFO de salida (*FIFO_S*).
4. El procesador, a través del hilo de envío de datos (*Thread_Env*) que está analizando en todo momento el estado de las memorias FIFO de salida, detecta que ya no están vacías. En este momento, obtiene el resultado almacenado, y de acuerdo a la memoria FIFO de la que proceden, los identifica como información de medición o como coordenada, y los envía al periférico *PComm Env*.
5. El periférico *PComm Env* recibe el dato del procesador, concretamente del hilo *Thread_Env*, y los envía al exterior. En el caso de la primera tarjeta FPGA estos resultados pasarán a la segunda tarjeta FPGA. Para el caso de la segunda tarjeta FPGA, estos datos son enviados al exterior del sistema SpaceCube.

Este procedimiento puede optimizarse de diferentes maneras. Una posibilidad podría ser conectar los periféricos *PComm Env* y *PComm Rec* directamente con el periférico Metodología. De esta manera

se aligeraría la carga del procesador, pudiendo dedicarse a otras tareas, o podría quedarse al margen del sistema. Si se considera esta conexión directa, el periférico Metodología debería poseer un submódulo - interfaz que permitiera interpretar las señales de control de los periféricos de comunicaciones y que recogiera e identificara correctamente los tipos de datos recibidos.

La Figura 4.21 muestra un diagrama aún más detallado del contenido del periférico Metodología. En ella, pueden visualizarse los módulos que la componen, y que ya fueron introducidos con anterioridad: *FIFO_E*, *FIFO_S* y Metodología.

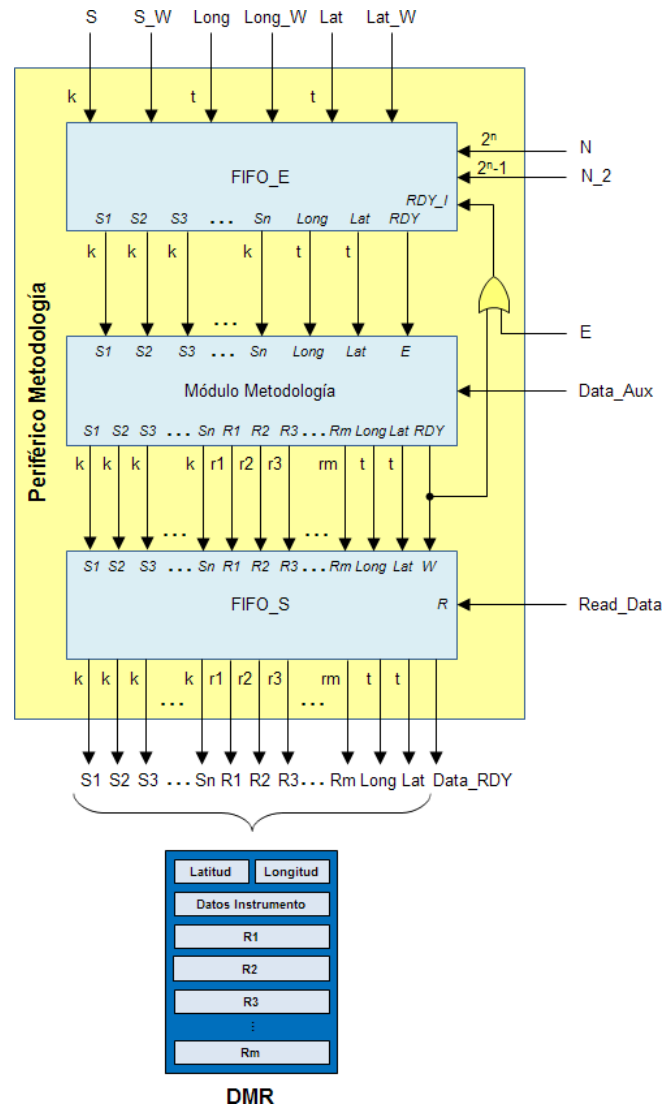


Figura 4.21: Diagrama detallado del periférico Metodología.

De manera general, este periférico realiza el procesamiento de una medición considerando la información instrumental como un vector (vector instrumental, $[S_1, S_2, S_3, \dots, S_n]$) compuesto por n valores (y que podría representar, por ejemplo, a un espectro), adjunto a unas coordenadas en latitud (Lat) y longitud ($Long$):

- El componente *FIFO_E* puede o bien recibir un valor de vector instrumental (S) de tamaño k bits, o bien una coordenada, representada por la latitud (Lat) y por la longitud ($Long$), ambas de t bits. Las señales S_W , S_Lat y S_Long se activarán cuando se quiera insertar en las memorias FIFO un nuevo valor de vector instrumental, de latitud o de longitud, respectivamente. La señal N indica el número de componentes que conforma el vector instrumental (n), mientras que la señal N_2 indica el cielo de la mitad de este mismo número: $\lceil n/2 \rceil$. Ambas señales son necesarias para el correcto funcionamiento del componente. Finalmente, la señal RDY_I indica que el módulo

Metodología está disponible para recibir una nueva medición (datos instrumentales junto con sus coordenadas) para su procesamiento (si es que hay disponibles). Esto ocurrirá tanto en la primera ejecución del circuito, indicada mediante la señal E o bien cuando el módulo Metodología haya terminado de procesar la medición actual (RDY). De ahí surge la necesidad de utilizar la puerta OR del centro del diagrama. Cuando el dispositivo está listo para procesar una nueva medición, y hay alguna disponible, $FIFO_E$ le proporciona al módulo Metodología dicha información, y le indica que puede comenzar el procesamiento mediante la señal RDY .

- El módulo Metodología recibe del componente $FIFO_E$ el vector instrumental ($S_1, S_2, S_3, \dots, S_n$), junto con sus coordenadas Lat y $Long$, y, a partir de la información adicional requerida por la metodología implementada (a través de las señales $Data_Aux$) genera unos resultados $R_1, R_2, R_3, \dots, R_m$, de tamaños $r_1, r_2, r_3, \dots, r_m$ bits, respectivamente. Estos resultados, junto con el vector instrumental y las coordenadas, son transferidos al componente $FIFO_S$ para su almacenamiento, indicándole una orden de escritura mediante la activación de la señal RDY .
- El componente $FIFO_S$ recibe del módulo Metodología el vector instrumental ($S_1, S_2, S_3, \dots, S_n$), junto con sus coordenadas Lat y $Long$, y los resultados obtenidos ($R_1, R_2, R_3, \dots, R_m$), almacenándolos en las memorias FIFOs correspondientes cuando se activa la señal RDY del módulo Metodología. Una vez almacenados, la señal $Data_RDY$ permanece activa indicando al exterior que ya existen resultados procesados. El exterior realiza una petición y recibe dichos datos activando la señal $Read_Data$, enviando de esta manera $FIFO_S$ toda la información almacenada relativa a una medición. La información instrumental, junto con las coordenadas en latitud y longitud pueden ser consideradas como registros DMR para el caso de la segunda tarjeta FPGA.

4.2.1. El módulo $FIFO_E$.

Tal y como se comentó anteriormente, este módulo se encarga de almacenar los datos procedentes del exterior, los cuales pueden ser información de medición (vector instrumental) o bien coordenadas (latitud y longitud). En caso de que la información no llegue en ráfagas ordenadas de dato instrumental - coordenadas, cada tipo de dato es almacenado en una memoria FIFO independiente. De tal manera que es posible encontrar que en un instante t se han recibido más componentes de vector instrumental que coordenadas, o viceversa. Sin embargo, y como se comentó con anterioridad, aunque ambos tipos de datos pueden llegar en ráfagas aleatorias, se ha considerado que nunca se recibirá un tipo de dato adquirido en el instante $t+1$ antes que uno adquirido en el instante t .

Por tanto, este módulo estará conformado principalmente, y entre otros componentes, por tres memorias FIFO (Figura 4.22): una para la información instrumental, otra para las longitudes y otra para las longitudes.

Para su funcionamiento, hay que tener las siguientes consideraciones:

- Las tres memorias FIFO almacenarán cada tipo de dato cuando el exterior lo requiera (en el sistema propuesto esta solicitud la realizará el procesador a través del hilo $Thread_Rec$) a través de las señales de escritura S_W , $Long_W$ y Lat_W .
- La lectura de una medición (información instrumental y coordenadas) de las memorias FIFO se realizará cuando se cumplan dos condiciones: Que exista almacenada una medición completa (que ninguna de las memorias FIFO correspondientes a las coordenadas esté vacía, y que el número de componentes de vector instrumental sea el correspondiente), y además el módulo de procesamiento (en el exterior) esté listo (señal RDY_I).
- Cuando la condición anterior se cumpla, la lectura se realiza mediante la utilización de un contador (Counter) que empezará a funcionar cuando una medición completa esté almacenado

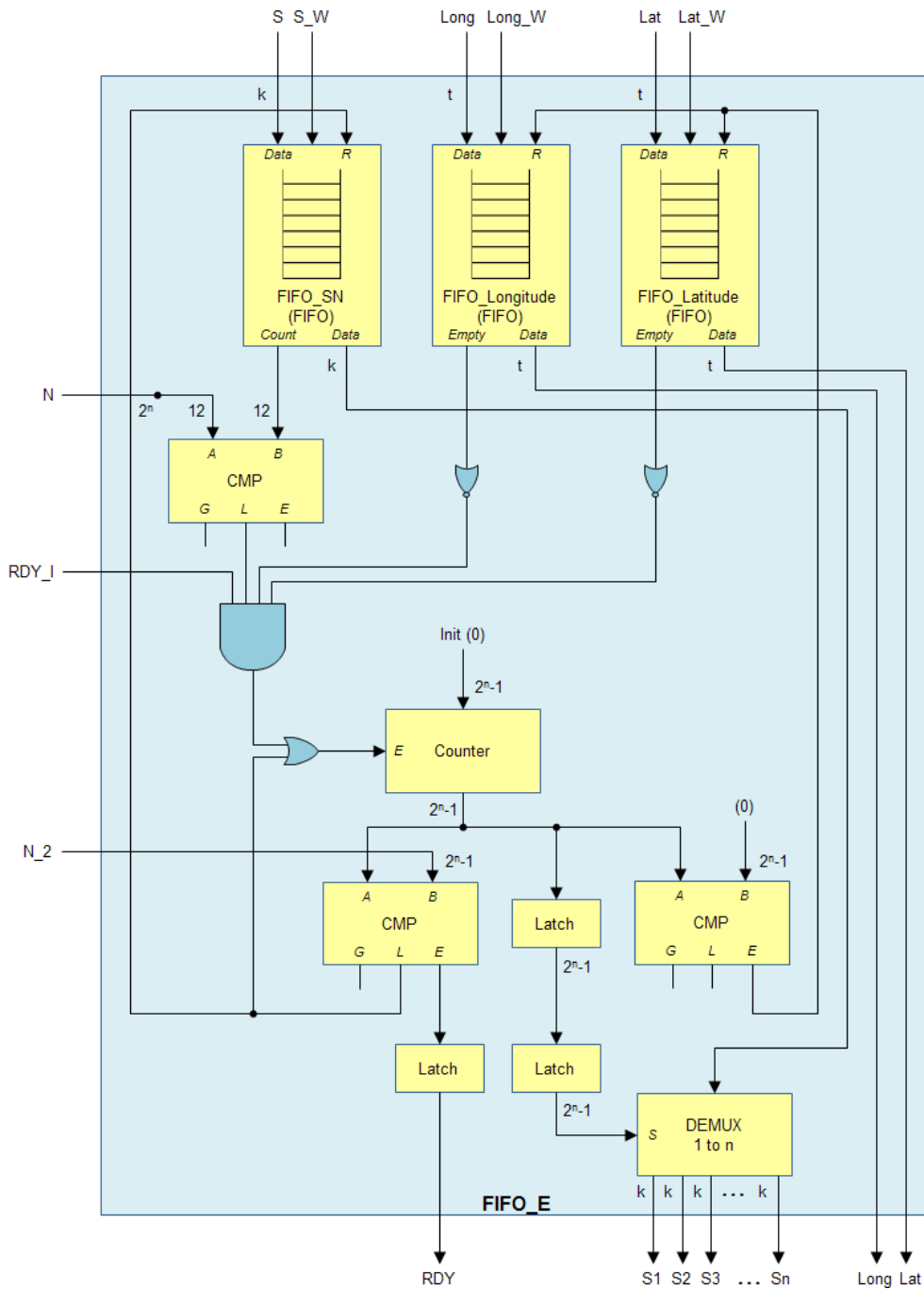


Figura 4.22: Diagrama general del módulo FIFO_E.

en las memorias FIFO. La lógica central compuesta por comparadores (CMP) y latches permiten que se realice una lectura de la memoria FIFO correspondiente a la información instrumental, por lo que se realizarán tantas lecturas como componentes conformen el vector instrumental. En la primera iteración, se obtiene, además del primer valor del vector instrumental, la latitud y longitud correspondientes.

- El demultiplexor final permite enviar los valores de vector instrumental al exterior de acuerdo a su posición, controlado por el número de iteración del contador.
- Una vez se ha terminado de leer todos los valores de vector instrumental de la memoria FIFO (el número de iteración alcanza el número de componentes de vector instrumental, señal Equal (E) del comparador), entonces se envía una señal RDY para indicar al exterior que una nueva medición se encuentra ya disponible para el procesamiento.

4.2.2. El módulo *FIFO_S*.

Tal y como fue presentado anteriormente, el módulo *FIFO_S* se encarga de almacenar la información espectral recién procesada, sus coordenadas, y los resultados obtenidos. Con el objetivo de garantizar un rápido almacenamiento y disponibilidad de dichos resultados, hace uso de memorias FIFO individuales tanto para cada valor de vector instrumental, como para las longitudes, las latitudes, y los resultados. De esta forma se garantiza la escritura y la lectura de toda la información referente a una medición completa en un único ciclo de reloj, estando los resultados disponibles para ser leídos dos ciclos de reloj después de haber sido obtenidos.

Por tanto, el módulo *FIFO_S* está compuesto por n memorias FIFO para los valores de vector instrumental, r memorias FIFO para los resultados generados, y dos memorias FIFO para las latitudes y la longitudes (Figura 4.23). Siguiendo este diseño, es posible que el número de memorias FIFO sea alto, pero como en este caso cada memoria sólo almacenará un único valor de vector instrumental de memoria independiente (no el vector instrumental entero), la capacidad de las mismas será muy inferior a las memorias FIFO presentes en el módulo *FIFO_E*. Por lo que finalmente, el número de componentes lógicos dentro de la FPGA y el consumo de memoria dedicados a la implementación de las memorias FIFO será similar tanto para el módulo *FIFO_S*, como para el módulo *FIFO_E*.

Para comprender el funcionamiento de este módulo hay que tener en cuenta las siguientes consideraciones:

- Cuando los resultados del procesamiento están disponibles, todas las memorias FIFO reciben una señal de escritura (W) para almacenar los datos correspondientes.
- Una vez almacenados los datos, la memoria FIFO correspondiente a la longitud enviará una señal al exterior ($Data_RDY$) indicando que ya no está vacía (señal *Empty* invertida) y que existen resultados procesados disponibles para leer. En realidad, todas las señales *Empty* de todas las memorias FIFO deberían estar a 0 si se da esta circunstancia. Pero al estar todas ellas sincronizadas, en vez de comprobar el estado de las mismas mediante una puerta NAND, se utilizará la correspondiente a uno de ellos (concretamente la de la longitud).
- Cuando existen resultados disponibles para leer (señal $Data_RDY$ activa), entonces desde el exterior (en el sistema propuesto lo realizará el procesador a través del hilo *Thread_Env*) se envía una señal de lectura (R) a todas las memorias FIFO, quedando dichos resultados disponibles en el siguiente ciclo de reloj.

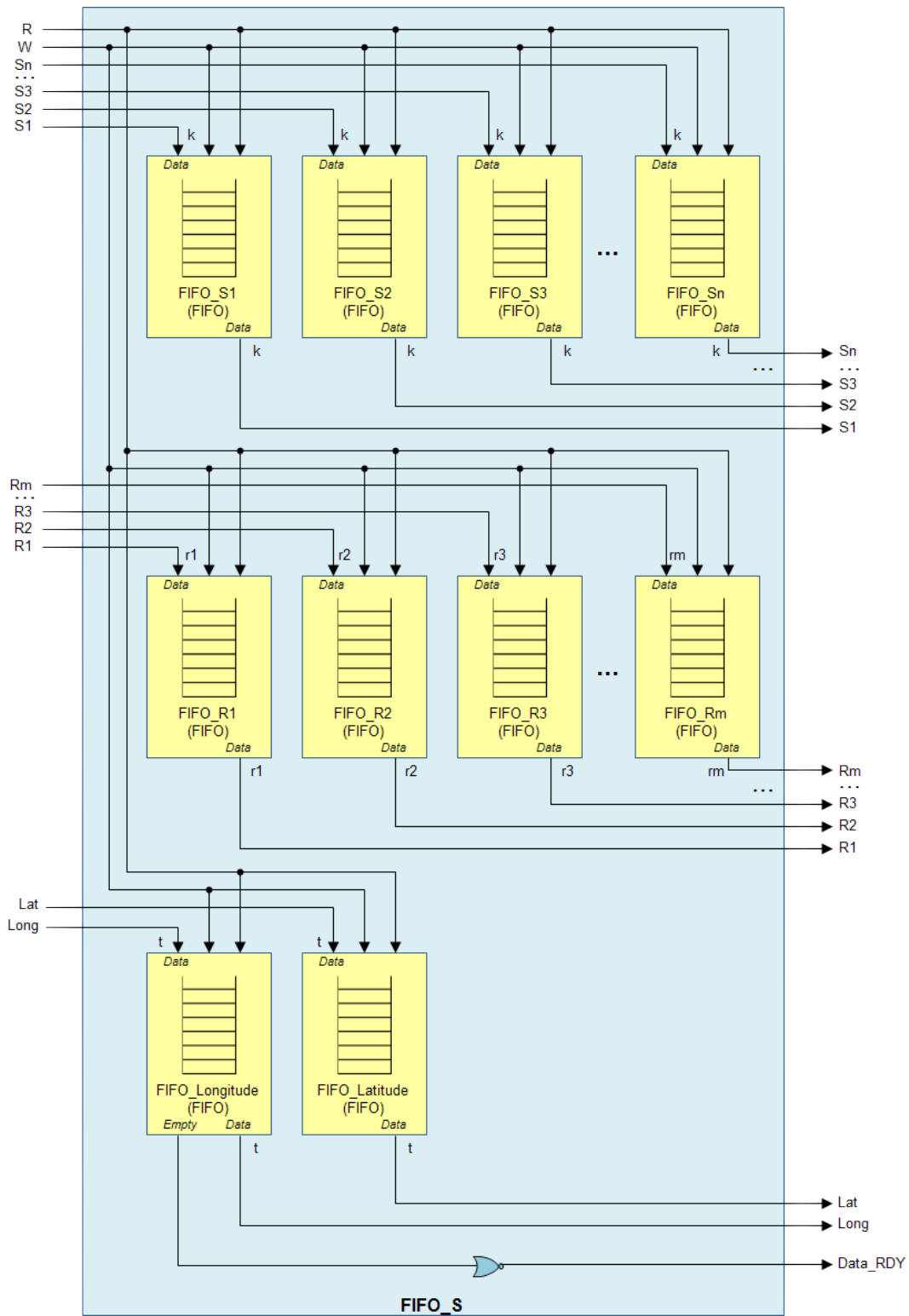


Figura 4.23: Diagrama general del módulo FIFO_S.

El siguiente capítulo realiza una serie de experimentos con el objetivo de estudiar el rendimiento global del formato DMF en diferentes aspectos y situaciones, incluyendo su adaptación y modificación a varios requisitos y necesidades de las misiones. Para la elaboración del estudio se utilizaron un dataset de imágenes creadas de manera sintética a partir de los movimientos a los que puede estar sometida la plataforma donde residen los dispositivos de adquisición, posicionamiento y procesamiento; y un dataset compuesto por imágenes reales adquiridas desde plataformas aero-transportadas y espaciales.

Capítulo 5

Resultados: Estudio del rendimiento del formato DMF

El formato DMF se presenta como una alternativa a los formatos convencionales de representación de datos adquiridos por sensores, bien integrados en espectrógrafos utilizados por un experto en una campaña de campo, o bien situados en plataformas a bordo de aviones o satélites para el escaneo de una determinada escena. Al tratarse de una estructura innovadora, se hace necesario el estudio del rendimiento y la eficiencia, respecto a los formatos de datos tradicionales, a la hora de realizar ciertas operaciones y de adaptarse ante las diferentes dificultades que presentan ciertas metodologías relacionadas con el procesamiento de imágenes. Para ello, se ha realizado un análisis del comportamiento del formato DMF mediante la generación de resultados correspondientes a diferentes ámbitos de procesamiento, así como su adaptación a diferentes situaciones en metodologías más complejas.

Este capítulo está compuesto por una serie de experimentos que utilizarán un conjunto de imágenes sintéticas y reales, las cuales se detallan en la Sección 5.1, con el objetivo de analizar el rendimiento y comportamiento del formato DMF tanto en su versión estándar como en las adaptaciones y modificaciones a la diferentes necesidades de las misiones, y tal y como se detalló en la Sección 4. Cada experimento estará dividido en cuatro apartados:

1. *Objetivos del experimento*: Finalidad del experimento. Descripción de los objetivos que se desean alcanzar mediante la elaboración de dicho experimento.
2. *Descripción del experimento*: Desarrollo detallado de la elaboración del experimento, indicando las imágenes utilizadas y posibles modificaciones del formato DMF.
3. *Resultados del experimento*: Resultados obtenidos al aplicar la metodología detallada en el apartado anterior.
4. *Análisis de los resultados del experimento*: Estudio de los resultados obtenidos para comprobar si con la metodología desarrollada se han conseguido cubrir los objetivos propuestos.

Los experimentos realizados abarcan el análisis del rendimiento del formato DMF en diferentes ámbitos de la computación, así como su adaptación a diferentes necesidades de las misiones, tanto espaciales como en Tierra. Estos experimentos pueden ser clasificados en dos grupos diferentes:

1. *Estudio de operaciones propias del procesamiento de imágenes*: Análisis del comportamiento del formato DMF ante operaciones sencillas que cualquier formato de datos debe ser capaz de afrontar, como son la construcción de la estructura de datos y visualización de los datos (Sección

5.2), selección automática del tamaño de celda DMF (Sección 5.3), consumo de espacio en disco (Sección 5.4), cálculo de los tiempos de ejecución (Sección 5.5), y la precisión espacial (Sección 5.6). Para la generación de estos resultados se han utilizado tanto el *dataset sintético* como el *dataset práctico*.

2. *Estudio del rendimiento frente a metodologías complejas*: Análisis del comportamiento, eficiencia y rendimiento frente a metodologías de procesamiento de imágenes más complejas, y que requieren la modificación de las estructuras y formatos de datos para un correcto funcionamiento, como son la adaptación a campañas de campo (Sección 5.7), la fusión de datos (Sección 5.8) y la detección de cambios (Sección 5.9). Debido a que se tratan de metodologías específicas en las que se pretende mostrar cómo es la adaptación del formato DMF, más que su rendimiento, para la generación de estos resultados sólo se han utilizado las imágenes correspondientes al *dataset real*.

5.1. Presentación de los datasets de imágenes utilizados en los experimentos.

Para la generación de los resultados y su posterior estudio se utilizaron dos datasets de imágenes de diferente naturaleza e índole, los cuales se diferencian principalmente por la manera en la que fueron creados o adquiridos. El primer dataset (*Dataset sintético*) está compuesto por una serie de imágenes sintéticas construidas de manera artificial, y harán posible la realización de los experimentos y su posterior análisis de manera teórica y libre de errores derivados de acuerdo a los diferentes movimientos a los que las plataformas pueden estar sometidas. El segundo dataset (*Dataset real*) está compuesto por diferentes imágenes adquiridas por sensores reales a bordo de aviones, o bien de satélites. Este dataset permitirá realizar conclusiones prácticas teniendo como referente los resultados de los modelos obtenidos con el dataset sintético anterior.

A continuación se proporciona información adicional sobre los dos datasets elaborados:

- *Dataset sintético*: Este dataset está compuesto por una serie de imágenes originadas de manera artificial de acuerdo a los movimientos *pitch*, *yaw* y *roll* a los que puede estar sometida la plataforma que transporta los instrumentos o sensores. Para su elaboración se consideró una escena sintética de 224 bandas cuyas esquinas albergan cuatro espectros puros normalizados de *Buddingtonite* (esquina superior izquierda), *Chlorite* (esquina superior derecha), *Halloysite* (esquina inferior izquierda) y *Jarosite* (esquina inferior derecha) [Cla07]. El resto de las posiciones de la escena están compuestas por una mezcla de estos cuatro espectros referencia, cuyas proporciones quedan determinadas por la localización y la distancia a cada una de las esquinas pertinentes (Figura 5.1). De esta forma, a una determinada posición de la escena $Scn(x, y)$ le corresponderá un espectro mezcla Sp tal que:

$$\begin{aligned}
 Sp(x, y, z) = & UL(z) \times \left(\left(1 - x \times \frac{1}{N_c - 1} \right) \times \left(1 - y \times \frac{1}{N_f - 1} \right) \right) + \\
 & + UR(z) \times \left(\left(x \times \frac{1}{N_c - 1} \right) \times \left(1 - y \times \frac{1}{N_f - 1} \right) \right) + \\
 & + LL(z) \times \left(\left(1 - x \times \frac{1}{N_c - 1} \right) \times \left(y \times \frac{1}{N_f - 1} \right) \right) + \\
 & + LR(z) \times \left(\left(x \times \frac{1}{N_c - 1} \right) \times \left(y \times \frac{1}{N_f - 1} \right) \right)
 \end{aligned} \tag{5.1}$$

Donde x e y representan la posición espacial dentro de la escena, z la banda del espectro, N_c y N_f la resolución espacial de la escena, UL el espectro referencia situado en la esquina superior izquierda (*upper left*), UR el espectro referencia situado en la esquina superior derecha (*upper*

right), *LL* el espectro referencia situado en la esquina inferior izquierda (*lower left*) y *LR* el espectro referencia situado en la esquina inferior derecha (*lower right*).

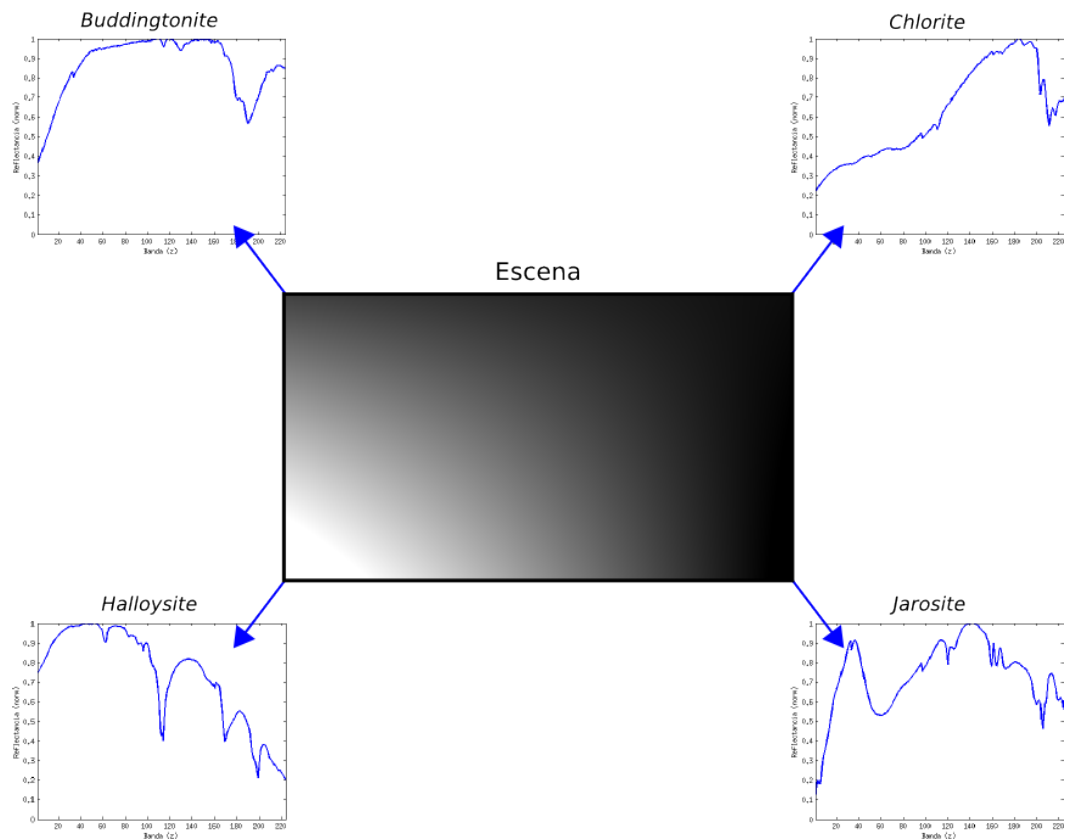
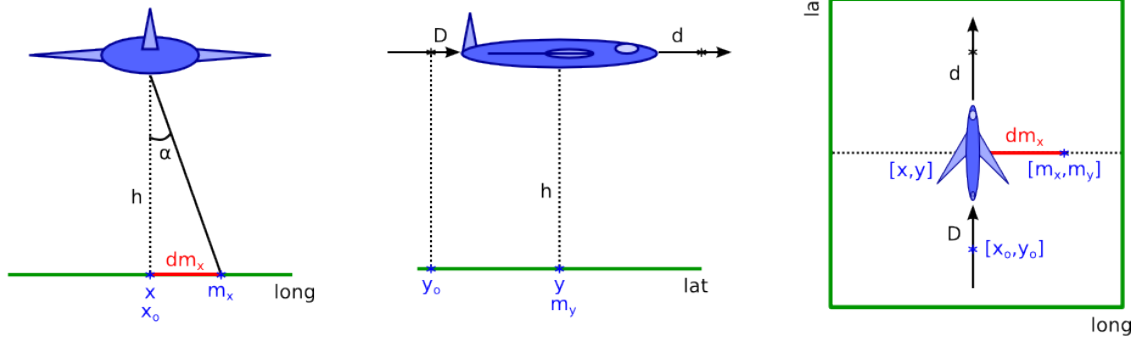


Figura 5.1: *Recreación de la escena sintética (Banda 0) generada a partir de cuatro espectros referencia.*

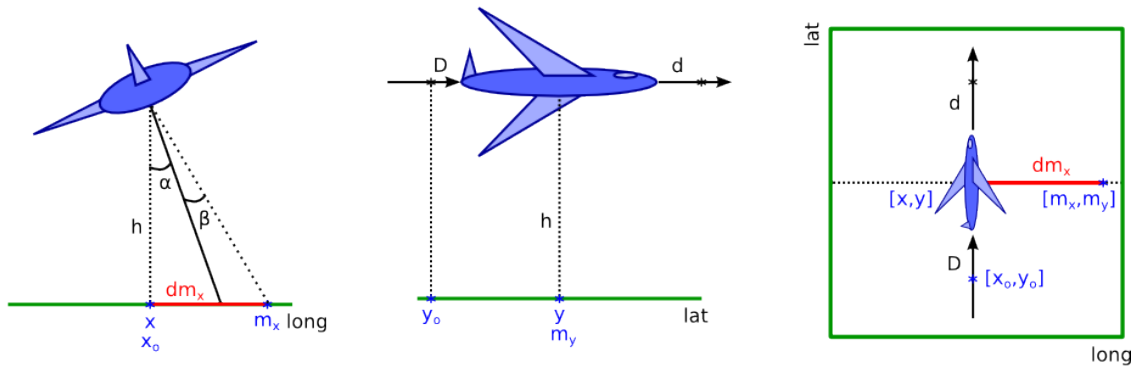
Sobre la escena generada de manera artificial se simularon los vuelos de un avión adquiriendo información espectral mediante un sensor de tipo *whiskbroom*, resultando una serie de imágenes L0 con sus correspondientes ficheros IGM que serán utilizados para la elaboración de los resultados [Ric12, Tak96]. Las series de imágenes resultantes proceden de la simulación de diferentes vuelos considerando los tres tipos de movimientos posibles: *pitch*, *roll* y *yaw*. Para cada tipo de movimiento se simularon 11 vuelos abarcando el rango de ángulos $[-30^\circ - 30^\circ]$, generando una imagen L0 (con fichero IGM asociado) por vuelo. El resultado final son 33 ficheros L0 e IGM, 11 para cada tipo de movimiento. La Figura 5.2 muestra el proceso de obtención de espectros sobre la escena sintética para un sensor de tipo *whiskbroom*.

1. Vuelo normal



$$\begin{aligned}
 dm_x(t) &= h \operatorname{tg}(\alpha(t)) & D(t + 1) &= D(t) + d \\
 m_x(t) &= dm_x(t) + x(t) & x(t) &= x_0 \\
 m_y(t) &= y(t) & y(t) &= y_0 + D(t)
 \end{aligned}$$

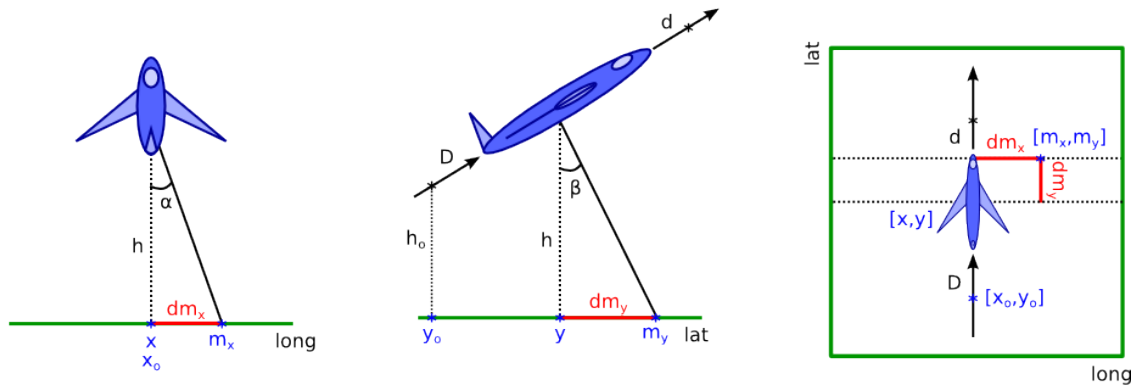
2. Roll



$$\begin{aligned}
 dm_x(t) &= h \operatorname{tg}(\alpha(t) + \beta) & D(t + 1) &= D(t) + d \\
 m_x(t) &= dm_x(t) + x(t) & x(t) &= x_0 \\
 m_y(t) &= y(t) & y(t) &= y_0 + D(t)
 \end{aligned}$$

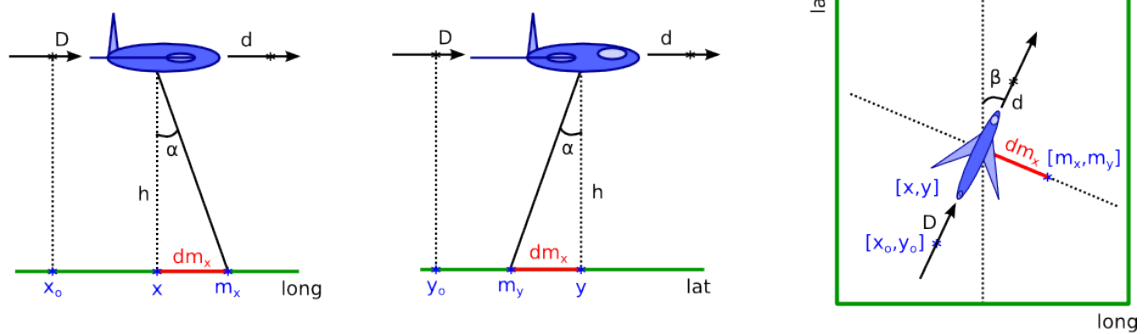
Figura 5.2: Obtención de espectros sobre la escena sintética para cada tipo de movimiento utilizando un sensor de tipo whiskbroom.

3. Pitch



$$\begin{aligned}
 dm_x(t) &= h(t) \operatorname{tg}(\alpha(t)) & D(t+1) &= D(t) + d \\
 dmy(t) &= h(t) \operatorname{tg}(\beta) & h(t) &= h_o + D(t) \sin(\beta) \\
 m_x(t) &= dm_x(t) + x(t) & x(t) &= x_o \\
 m_y(t) &= dm_y(t) + y(t) & y(t) &= y_o + D(t) \cos(\beta)
 \end{aligned}$$

4. Yaw



$$\begin{aligned}
 dm_x(t) &= h \operatorname{tg}(\alpha(t)) & D(t+1) &= D(t) + d \\
 m_x(t) &= dm_x(t) \cos(-\beta) + x(t) & x(t) &= x_o + D(t) \cos(90^\circ - \beta) \\
 m_y(t) &= dm_x(t) \sin(-\beta) + y(t) & y(t) &= y_o + D(t) \sin(90^\circ - \beta)
 \end{aligned}$$

Figura 5.2: (Cont.)

Cada una de las 11 imágenes implica un ángulo de movimiento *pitch*, *roll* y *yaw* abarcando en total el rango $[-30^\circ - 30^\circ]$. La Figura 5.3 muestra la primera banda de los ficheros L0 generados para los casos de ángulos extremos (-30° y 30°) y para dos de los ángulos intermedios (-12° y 12°), de cada uno de los movimientos representados. A partir de los ficheros L0 e IGM obtenidos durante la simulación de adquisición, se realizó una georreferenciación convencional de los datos, obteniendo los ficheros geocorregidos correspondientes, los cuales se muestran, para los ángulos anteriores y para los tres tipos de movimientos posibles, en la Figura 5.4. Sus principales características, como el número de filas y columnas que las componen, así como el tamaño de píxel utilizado, se muestran representadas en la Figura 5.5 y en la Figura 5.6, respectivamente. Para este caso particular, cada vuelo realizado fue iniciado con los ajustes y parámetros mostrados en la Tabla 5.1.

Posición Inicial $[x_o, y_o]$	[5000,5000] m (UTM)
Altura (h)	4000 m
Nº Samples Sensor	501
Nº Líneas Vuelo	800
Nº Bandas	224
Ifov (α)	0.002 rad
Velocidad	40 m/s

Tabla 5.1: Configuración y especificaciones técnicas de los vuelos realizados sobre la escena sintética.

- *Dataset real*: Este dataset está formado por una serie de imágenes adquiridas por sensores reales transportados a bordo de aviones o satélites. La condiciones de vuelo, geometrías de los sensores, e incluso el tipo de georreferenciación es diferente para caso:
 - *Cuprite* (Figuras 5.7(a) y 5.7(b)): Esta imagen fue adquirida en el año 1997 por el sensor aerotransportado NASA/AVIRIS [Boa95] en Nevada (USA). Es una referencia en el campo de la teledetección, ya que el área que abarca ha sido geológicamente estudiado y mapeado en detalle desde 1977.
 - *Barrax* (Figuras 5.7(c) y 5.7(d)): Esta imagen fue adquirida en el año 2004 por el sensor CHRIS [Gua05] a bordo del satélite ESA/PROBA [Bar04] abarcando la zona de Barrax (España). Este área es también una referencia en el campo de la teledetección, ya que se trata de un terreno llano, los cielos suelen estar siempre despejados y existen cultivos en condiciones controladas. Muchas campañas científicas financiadas por la Agencia Espacial Europea (ESA), como DAISEX [Ber01] o SEN2FLEX [Sob08] han sido desarrolladas principalmente en este área.
 - *Mediterráneo* (Figuras 5.7(e) y 5.7(f)): Esta imagen fue adquirida en el año 2010 por el instrumento MODIS a bordo del satélite NASA/Terra [Sal02]. Se trata de una escena que cubre varios países de la zona mediterránea. Los productos ofrecidos por MODIS, operativo desde 1999, y desde 2002 a bordo del satélite Aqua [Par03], son utilizados por miles de personas en el campo de la teledetección y son una referencia en ciencias implicadas en la monitorización y detección de cambios de la corteza terrestre.
 - *GMéxico* (Figuras 5.7(a) y 5.7(h)): Esta imagen fue adquirida en el año 2010 por el sensor aerotransportado NASA/AVIRIS sobre una zona costera del Golfo de México, en una misión que tenía como objetivo principal el estudio del vertido de petróleo de la plataforma Deepwater Horizon que tuvo lugar en Abril de ese mismo año. No obstante, la imagen seleccionada únicamente muestra una zona costera en la que la presencia de mar es mínimo, ya que de lo contrario sería difícil analizar de manera precisa y visual el proceso de georreferenciación. La principal diferencia entre esta imagen y la imagen de *Cuprite* es que en esta última el sensor está actualizado en su última versión tras varias mejoras a lo largo de los años tanto en el sensor como en el sistema de posicionamiento [Eas00a, Eas00b].
 - *Cantera* (Figuras 5.7(i) y 5.7(j)): Esta imagen fue adquirida en el año 2005 por el sensor aerotransportado español INTA/AHS [Fer05], en colaboración con la misión SEN2FLEX

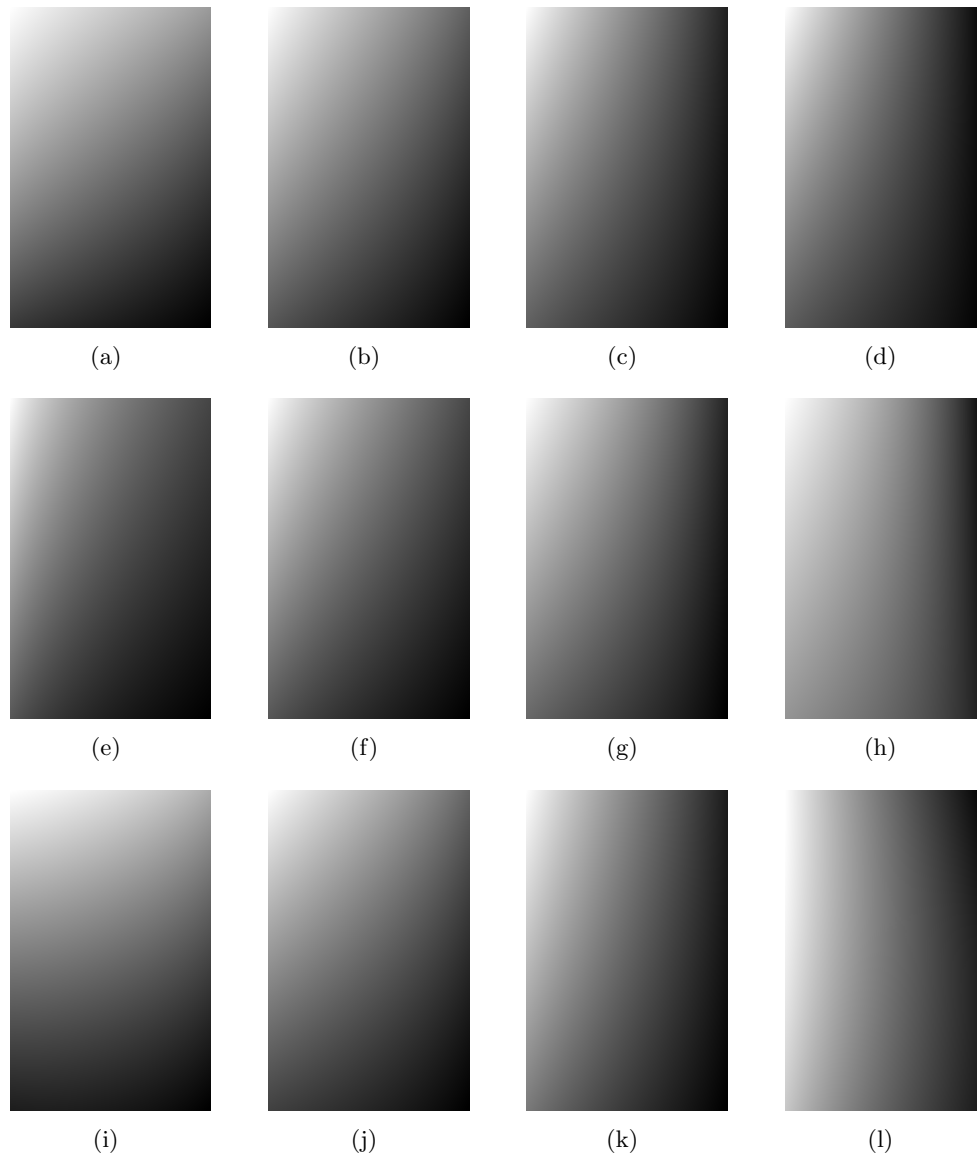


Figura 5.3: Imágenes $L0$ obtenidas mediante la simulación de vuelos sobre la escena sintética considerando los tipos de movimientos posibles con diferentes ángulos (Banda 0).
(a) Pitch -30° . (b) Pitch -12° . (c) Pitch 12° . (d) Pitch 30° . (e) Roll -30° . (f) Roll -12° . (g) Roll 12° . (h) Roll 30° . (i) Yaw -30° . (j) Yaw -12° . (k) Yaw 12° . (l) Yaw 30° .

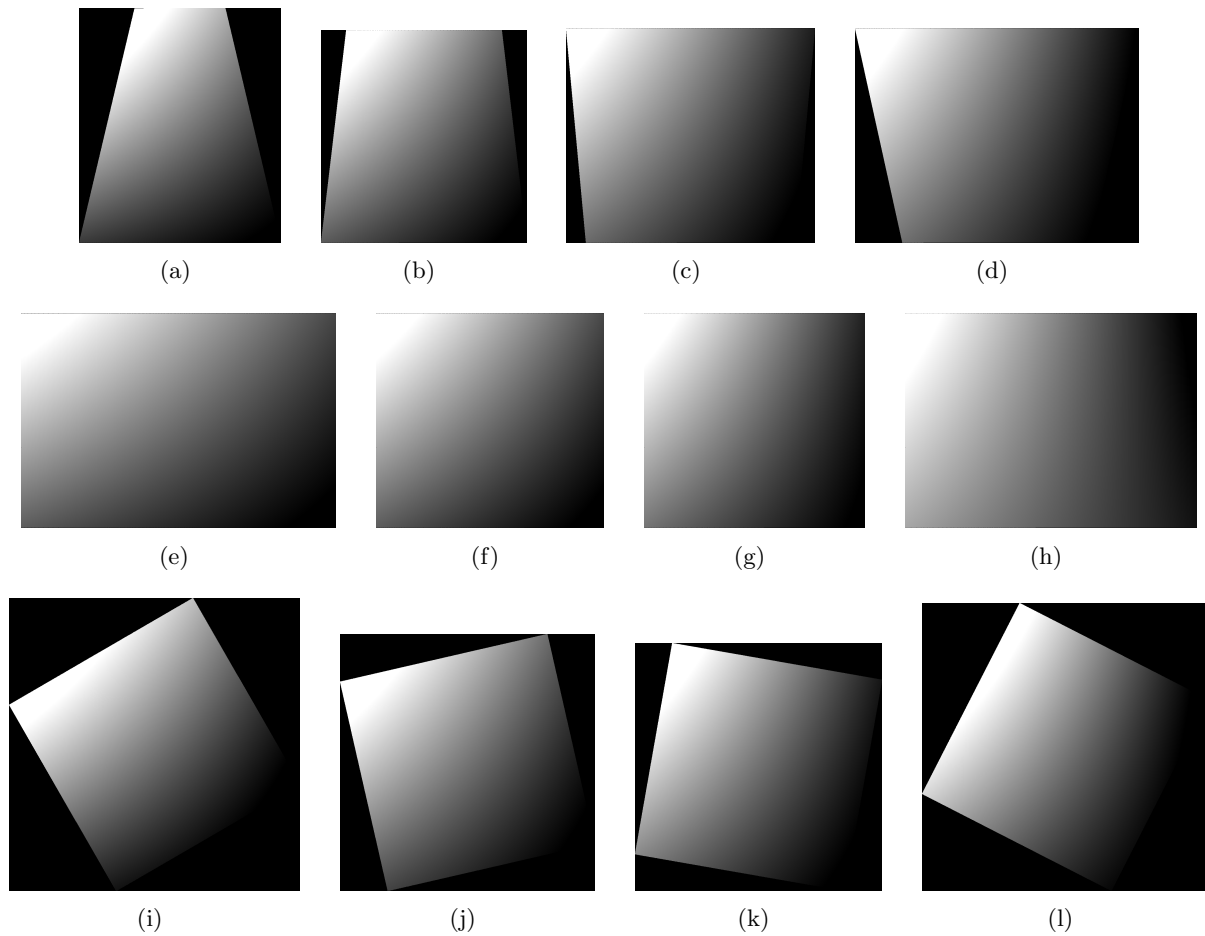


Figura 5.4: Imágenes geocorregidas a partir de los ficheros L0 e IGM obtenidos tras la simulación de adquisición (Banda 0).

(a) Pitch -30° . (b) Pitch -12° . (c) Pitch 12° . (d) Pitch 30° . (e) Roll -30° . (f) Roll -12° . (g) Roll 12° . (h) Roll 30° . (i) Yaw -30° . (j) Yaw -12° . (k) Yaw 12° . (l) Yaw 30° .

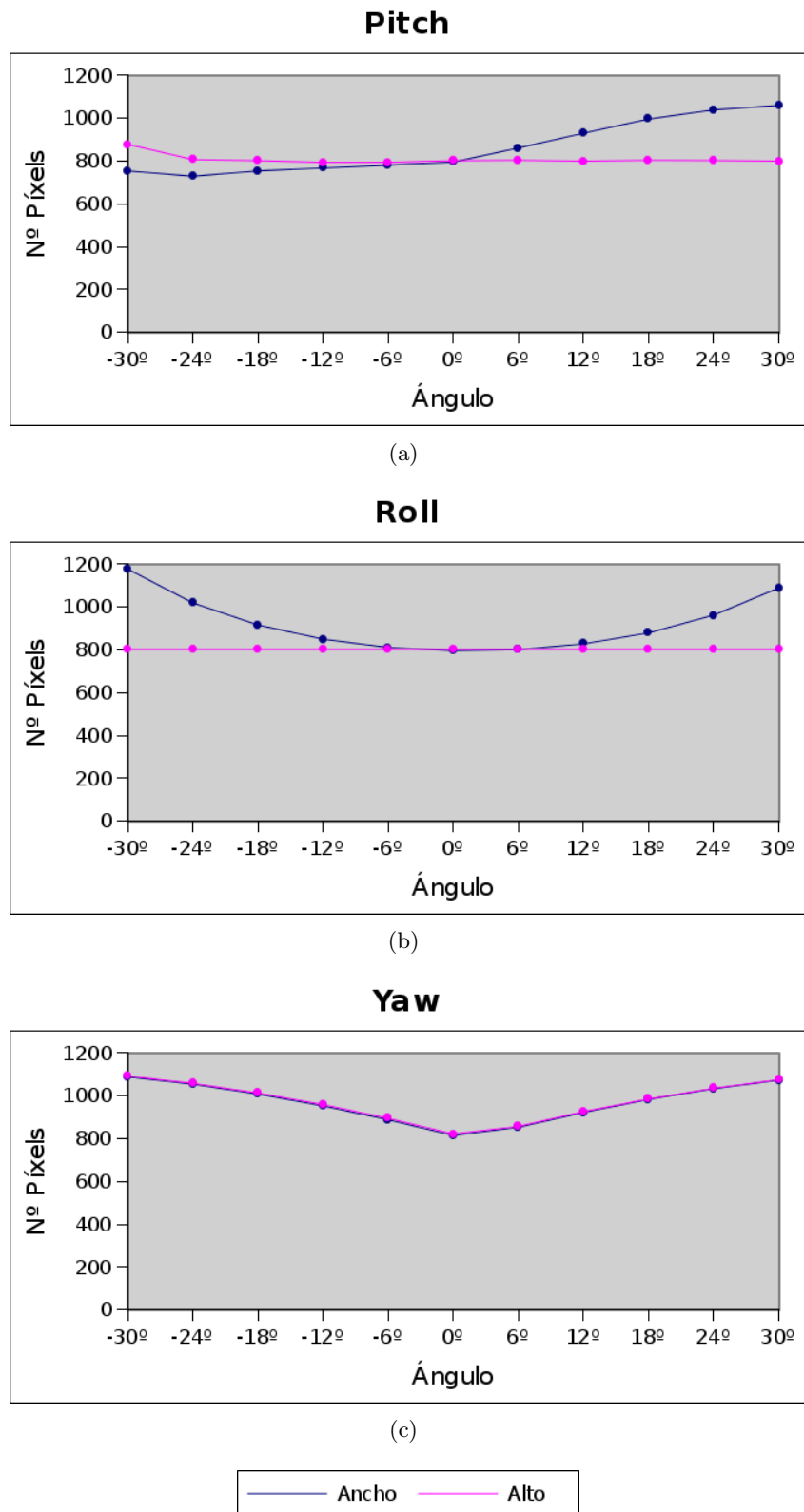


Figura 5.5: Resolución espacial de las imágenes geocorregidas resultantes del vuelo simulado para cada uno de los movimientos.

(a) Pitch. (b) Roll. (c) Yaw.

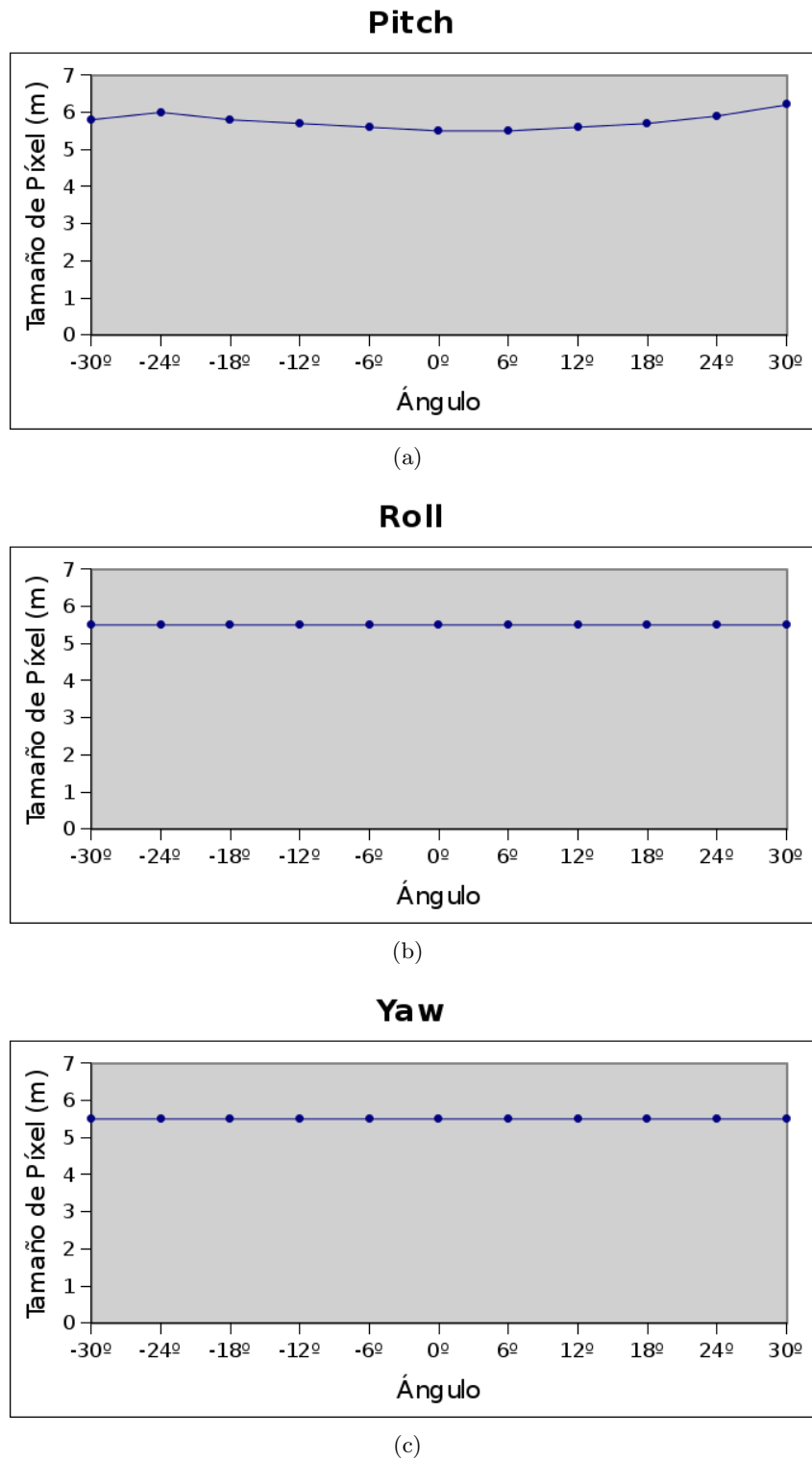


Figura 5.6: *Tamaño de píxel de las imágenes geocorregidas resultantes del vuelo simulado para cada uno de los movimientos.*

(a) Pitch. (b) Roll. (c) Yaw.

de la ESA [Sob09]. La escena representada es la misma que *Barrax*, pero en un área más localizada conocida como La Cantera. Resulta interesante estudiar esta misma zona desde diferentes sensores, tanto aerotransportados como a bordo de satélites.

- *Campaña*: Una imagen que pretende simular los datos que se recogen durante una campaña de campo, en la cual un experto toma mediciones de un área de interés. Fue realizada *in situ* en el área de Barrax, utilizando un espectrómetro durante la elaboración del proyecto Datasat [DATwww]. El resultado es un fichero de texto en el que se encuentran, para cada medición adquirida, su localización espacial, su espectro y un identificador que indica el tipo de material al que pertenece:
 - *Festuca*.
 - *Agua*.
 - *Suelo descubierto*.
 - *Maíz*.
 - *Trigo*.

Las características de las imágenes previamente presentadas se encuentran detalladas en la Tabla 5.2.

	Cuprite	Barrax	Mediterráneo	GMéxico	<i>Cantera</i>
<i>Sensor</i>	AVIRIS	CHRIS	MODIS	AVIRIS	AHS
<i>Plataforma</i>	Avión	Satélite	Satélite	Avión	Avión
<i>Sist. Coordenadas</i>	UTM (m)	Geográficas (°)	Geográficas (°)	Geográficas (°)	UTM (m)
<i>Fil. × Col. L0</i>	1087 × 677	374 × 370	2030 × 1354	2485 × 677	2000 × 750
<i>Fil. × Col. Geo</i>	1358 × 1048	444 × 623	2278 × 3416	2108 × 911	1458 × 1975
<i>Nº Bandas</i>	224	62	22	224	63
<i>Tamaño Píxel</i>	9.8 m	0.000315°	0.009411°	0.000037°	4.4 m

Tabla 5.2: Detalle de las características de las imágenes que conforman el dataset sintético.

5.2. Construcción y visualización de la matriz DMF.

En esta sección se muestra el resultado de procesar todas las mediciones procedentes de los sensores a bordo de la plataforma para la construcción de la matriz DMF, así como el aspecto visual de la misma al asignar cada medición a su celda DMF correspondiente.

5.2.1. Objetivos del experimento.

Para poder comprobar de manera visual la distribución espacial de los datos adquiridos por los sensores, así como la información que almacenan, es necesario desarrollar una metodología que permita convertir del formato DMF a un tipo de imagen convencional que pueda ser reconocido por las aplicaciones de tratamiento de imágenes del usuario. De esta forma, con este experimento se pretenden conseguir los siguientes objetivos:

- Estudiar la manera en la que se almacenan y visualizan los datos procedentes de los sensores en la matriz DMF.
- Realizar un estudio extensivo sobre las ventajas y deficiencias que presenta el formato DMF como medio de análisis visual de imágenes de teledetección.

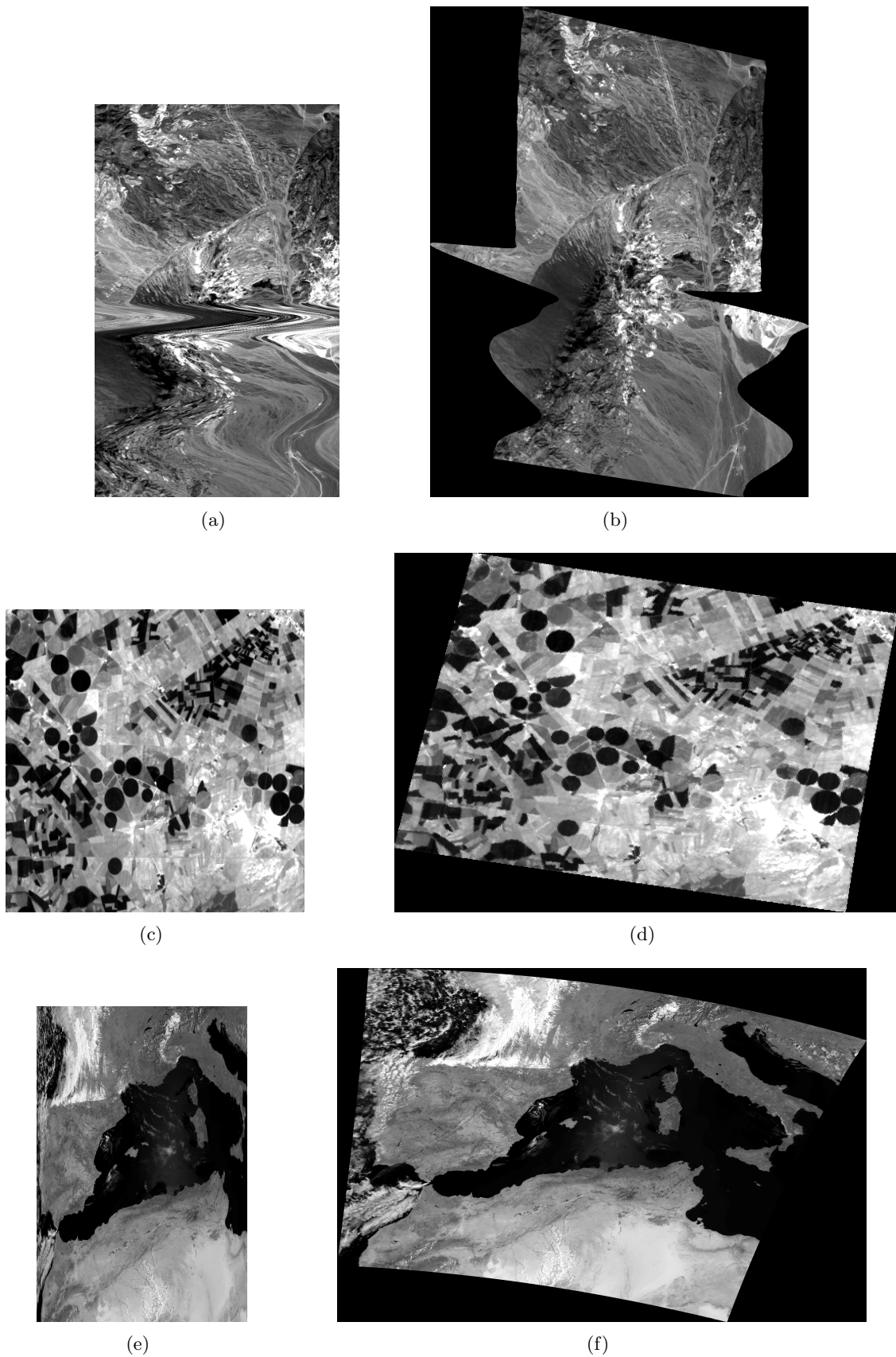


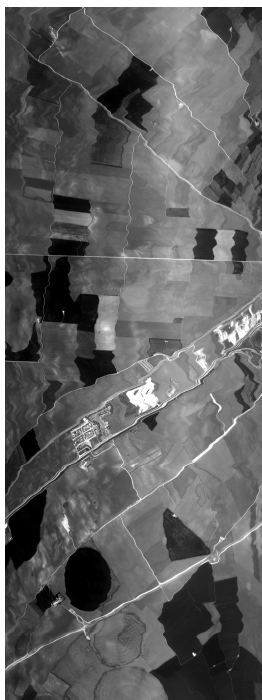
Figura 5.7: Visualización del set de imágenes seleccionado para realizar los experimentos. (a) Versión L0 de la escena Cuprite [Banda 10]. (b) Versión geocorregida de la escena Cuprite [Banda 10]. (c) Versión L0 de la escena Barrax [Banda 5]. (d) Versión geocorregida de la escena Barrax [Banda 5]. (e) Versión L0 de la escena Mediterráneo [Banda 4]. (f) Versión geocorregida de la escena Mediterráneo [Banda 4]. (g) Versión L0 de la escena GMéxico [Banda 10]. (h) Versión geocorregida de la escena GMéxico [Banda 10]. (i) Versión L0 de la escena Cantera [Banda 0]. (j) Versión geocorregida de la escena Cantera [Banda 0].



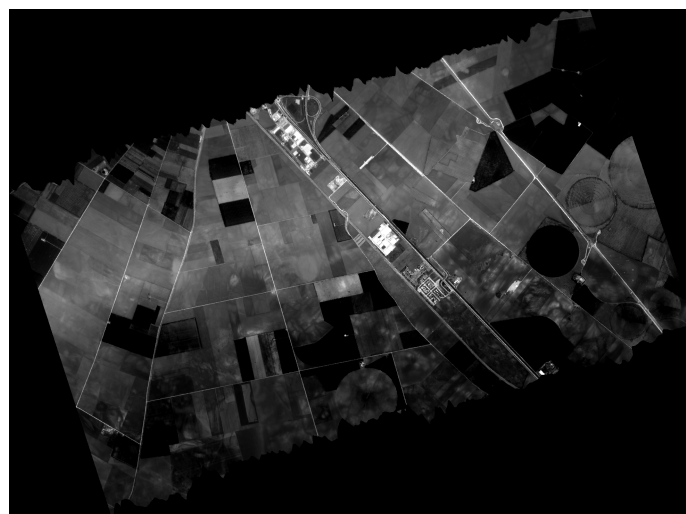
(g)



(h)



(i)



(j)

Figura 5.7: (Cont.)

5.2.2. Descripción del experimento.

Para analizar el comportamiento del formato DMF a la hora de construir la matriz y visualizar los datos en pantalla, se utilizaron tanto las imágenes del *dataset sintético* como el del *dataset real*.

La manera en la que se construye y visualiza la matriz DMF fue detallada en la Sección 4.1.1.1 y en la Sección 4.1.1.2. A partir de los ficheros IGM que almacenan las coordenadas de cada una de las mediciones y de los ficheros L0 que almacenan los espectros, se construyeron las matrices DMF correspondientes utilizando el mismo tamaño de celda DMF que el tamaño de píxel ofrecida por los proveedores en sus versiones geocorregidas (Figura 5.6 y Tabla 5.1), para una directa comparación con éstas últimas.

A la hora de mostrar la matriz DMF en pantalla, se creará una imagen RGB con las mismas dimensiones y en la que cada píxel representará la información almacenada en la celda DMF análoga.

5.2.3. Resultados del experimento.

Tras el proceso de creación de las matrices DMF, se realizó una visualización de las mismas en pantalla. El resultado es una imagen RGB en donde a cada celda DMF le corresponde un píxel de pantalla, mostrando el valor de espectro de la medición más cercana al centro de dicha celda. Los píxeles en negro son celdas DMF que no contienen ninguna medición. Los valores de espectro siguen un esquema de color tal que los valores altos se representan en color blanco, y los más bajos, de color azul.

La Figura 5.8 muestra la visualización de las matrices DMF de varias de las imágenes simuladas del *dataset sintético*, mostrando las correspondientes a los ángulos $\pm 30^\circ$ y $\pm 12^\circ$. La Figura 5.9 muestra también dicha visualización para las escenas *Cuprite*, *Barrax*, *Mediterráneo*, *GMéxico* y *Cantera*.

5.2.4. Análisis de los resultados del experimento.

A la hora de adaptar el formato DMF a un tipo de estructura de representación que permita el análisis visual de la disposición de los datos, se obtuvieron los siguientes resultados:

1. Adaptación visual para el *dataset sintético*:
 - a) Representación de los datos mediante un esquema de color basado en RGB (Figura 5.8).
2. Adaptación visual para el *dataset real*:
 - a) Representación de los datos mediante un esquema de color basado en RGB (Figura 5.9).

Este tipo de representación de la matriz DMF mediante una imagen RGB únicamente es útil para visualizar la disposición de los datos, ya que puede tener varias carencias. Una de ellas es la imposibilidad de mostrar de manera adecuada mediciones con diferentes tamaños y formas de *footprint*. Al asignar una medición por píxel estamos considerando que todas las mediciones tienen la misma forma y tamaño, y no se diferencian casos en los cuales estos parámetros pueden variar, como por ejemplo, mediciones al borde de la escena cuyo *footprint* no es circular por la inclinación del terreno, y de mayor tamaño por la distancia al Nadir. Este problema puede ser más o menos resuelto representando las mediciones de la imagen no a nivel de píxel, si no a una escala de la misma. De esta manera, una medición podría ser representada por varios píxeles (según su tamaño) de una manera no necesariamente circular (según su forma).

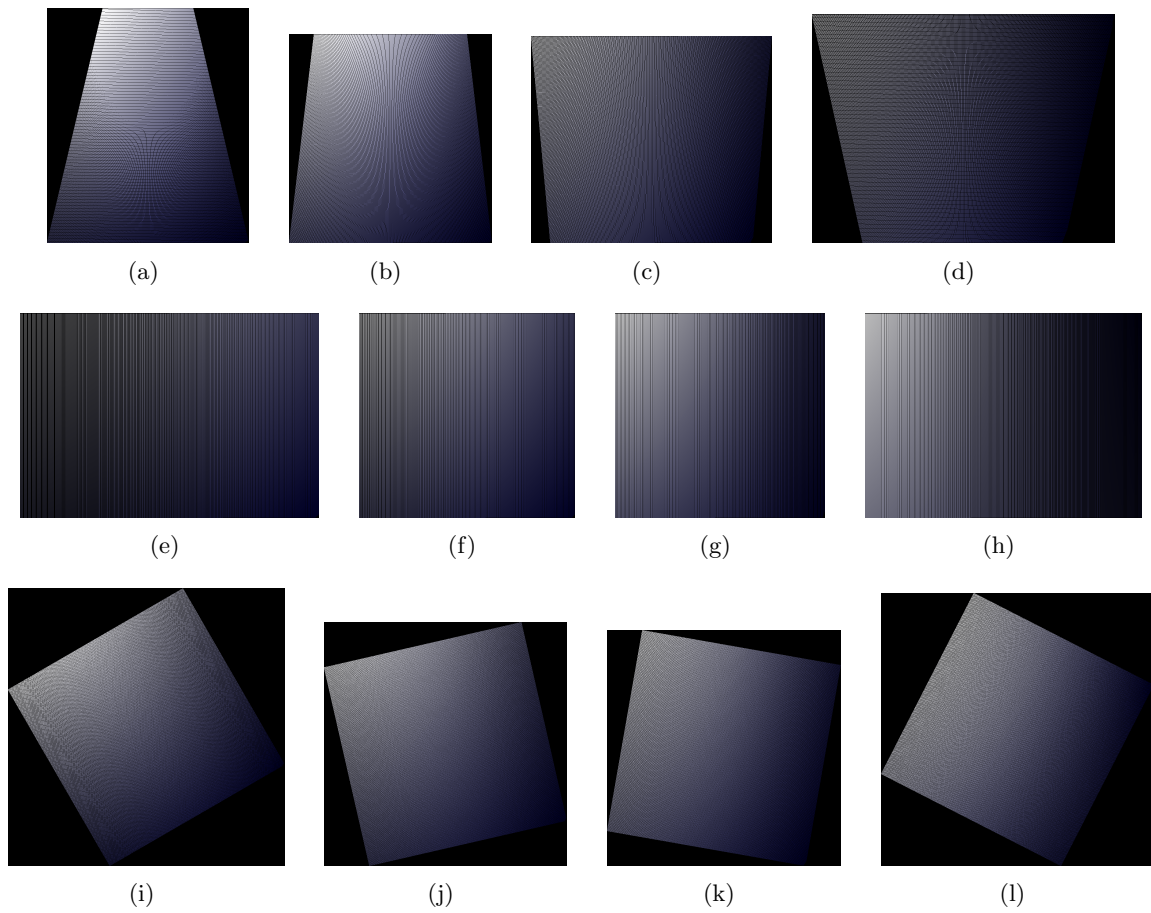
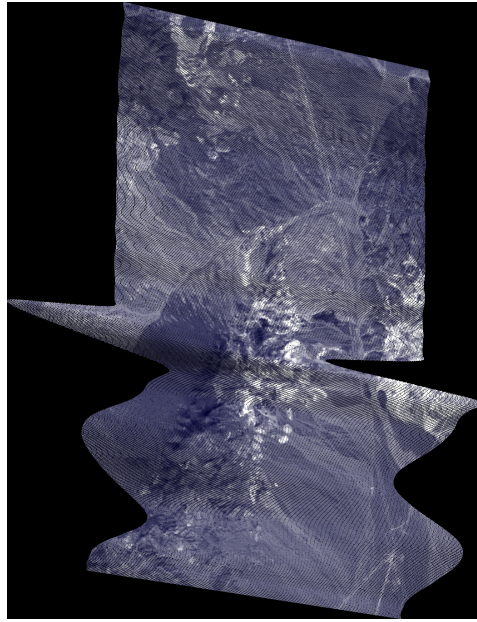
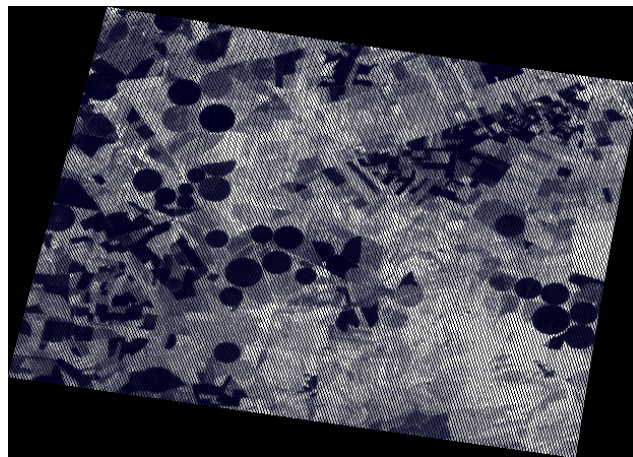


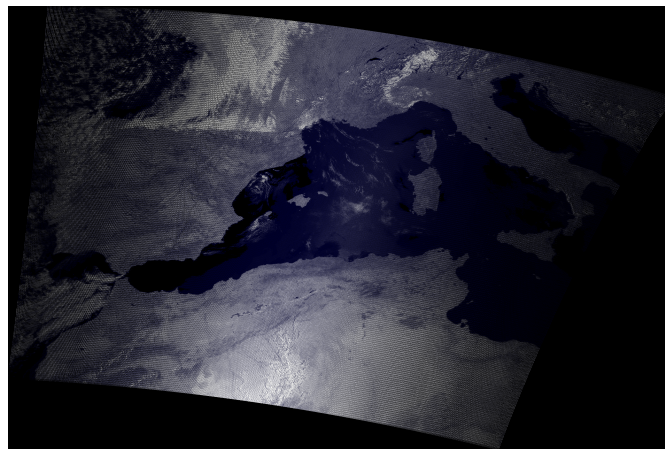
Figura 5.8: Visualización de las matrices DMF para el dataset de imágenes sintético (Banda 0).
 (a) Pitch -30° . (b) Pitch -12° . (c) Pitch 12° . (d) Pitch 30° . (e) Roll -30° . (f) Roll -12° . (g) Roll 12° . (h) Roll 30° . (i) Yaw -30° . (j) Yaw -12° . (k) Yaw 12° . (l) Yaw 30° .



(a)

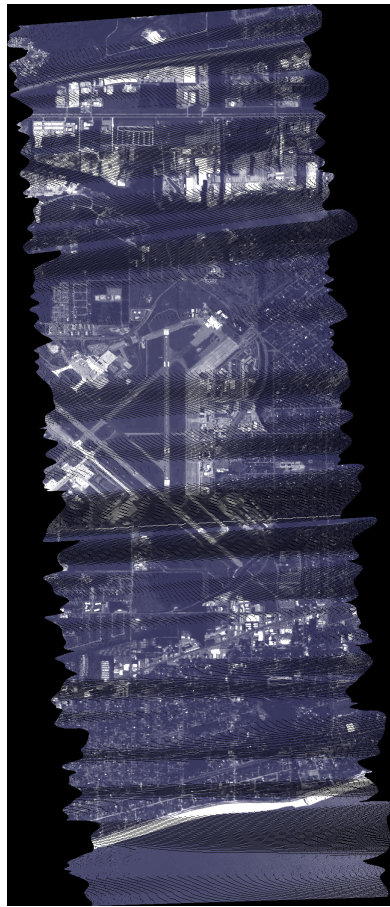


(b)

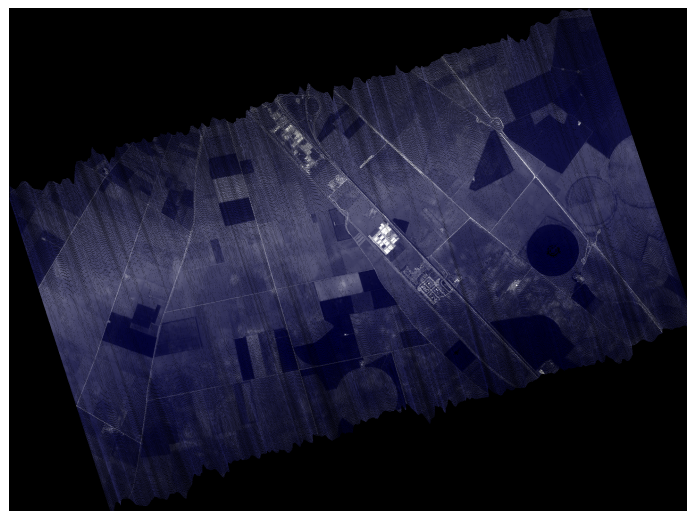


(c)

Figura 5.9: Visualización de las matrices DMF para el dataset de imágenes real. (a) Escena Cuprite [Banda 10]. (h) Escena Barrax [Banda 0]. (c) Escena Mediterráneo [Banda 4]. (g) Escena GMéxico [Banda 10]. (b) Escena Cantera [Banda 5].



(g)



(h)

Figura 5.9: (Cont.)

Otro problema que no considera este tipo de visualización es el muestreo de varias mediciones asignadas a una misma celda DMF. El hecho de mostrar únicamente la medición más cercana al centro supone que un gran porcentaje de información almacenada en la matriz DMF no es representada en pantalla. Existen diferentes maneras que podrían optimizar este proceso. Una de ellas consistiría en crear una imagen RGB con una dimensión espacial que garantice una medición por píxel. Pero esto puede provocar imágenes muy grandes y difíciles de visualizar si existen dos mediciones muy cercanas. Otra posible solución puede ser la creación de un interfaz gráfico interactivo en el que fuera posible seleccionar la celda DMF de interés, y que la disposición de los datos en ésta sean representados correctamente mediante otra imagen RGB.

En conclusión, la visualización de la matriz DMF genera una imagen RGB que únicamente debe ser utilizada para comprobar la disposición espacial de las mediciones. En ningún caso esta imagen RGB generada debería ser utilizada para procesamiento, ya que únicamente considera una medición por píxel e ignora diferentes parámetros físicos del *footprint* de la medición. Si se desean procesar los datos adquiridos por el sensor, se debe utilizar la matriz DMF original, considerando que no es necesario procesar las celdas DMF vacías (*Null*) y que en una celda DMF es posible encontrar una o más mediciones.

5.3. Cálculo automático del tamaño de celda DMF.

Existen ocasiones en las cuales la información del vuelo del avión o satélite, o bien algunos datos físicos del sensor, no están disponibles, por lo que no es posible calcular a priori y de manera directa el tamaño de píxel a utilizar para reconstruir la escena escaneada. En estos casos, se hace necesaria una metodología que permita analizar la disposición espacial de los datos adquiridos y estime de manera automática la distancia entre las mediciones, intentando garantizar una medición por celda para una mejor representación y procesado.

5.3.1. Objetivos del experimento.

En este apartado se ha diseñado un algoritmo que, a través del estudio de las coordenadas de las mediciones adquiridas por el sensor, determine cuál es el tamaño de celda DMF que se debe utilizar para construir la matriz DMF, tanto en longitud como en latitud. Cabe recordar que el tamaño de celda DMF es un parámetro que puede ser variable, por lo que para su construcción puede utilizarse el tamaño estimado por esta metodología (o puede ser meramente orientativo si se duda en qué tamaño utilizar), el proporcionado por el proveedor de los datos, o cualquiera de acuerdo a las necesidades de las técnicas de procesamiento requeridos.

De esta manera, los objetivos que se pretenden cubrir con la elaboración de este experimento son los siguientes:

- Detallar la metodología mediante la cual se realiza el cálculo automático de celda DMF.
- Realizar un estudio sobre la disposición de los datos en una matriz DMF para comprobar que se garantiza una medición por celda DMF y una reducción en el porcentaje de celdas DMF sin medición.
- Mostrar el aspecto visual que presentan las matrices DMF tras aplicar los tamaños de celda DMF proporcionada por el procedimiento diseñado.
- Analizar qué tipo de movimiento de plataforma afecta más a la distribución de las mediciones sobre la matriz DMF de acuerdo al tamaño de celda DMF aplicado.
- Estudiar cómo la magnitud de los movimientos de plataforma afectan a la distribución de las mediciones sobre la matriz DMF de acuerdo al tamaño de celda DMF aplicado.

5.3.2. Descripción del experimento.

El algoritmo de estimación de tamaño de celda DMF hará uso de la información almacenada en el fichero IGM, el cual proporciona las coordenadas espaciales de todas las mediciones adquiridas por el sensor. A partir de dicha información, se estudiará la distancia en longitud de cada medición a la siguiente (en columna, en el sentido de adquisición de datos) y la distancia en latitud de cada medición a la siguiente (en fila, en el sentido de vuelo). De esta manera, es posible construir una tabla de distancias entre mediciones en longitud y otra en latitud. Estas dos tablas pueden ser utilizadas para extraer el tamaño de celda DMF de manera automática mediante el estudio estadístico de las distancias entre mediciones, intentando garantizar siempre una medición por celda DMF.

Existen diversas maneras de realizar este procedimiento, que pueden variar desde simples metodologías estadísticas (como la media, mediana, moda, cuartiles, etc.) hasta otras más complejas (como estudio de los datos en el dominio de la frecuencia, etc.). La metodología diseñada, sin embargo, se basa en el estudio de las frecuencias de las distancias entre las mediciones. Mediante este procedimiento, el cual se detalla a continuación, es posible construir un histograma para poder visualizar en qué rango de distancias están situados los datos, y debe realizarse tanto para las distancias en longitudes (Figura 5.10(a)) como para las distancias en latitudes (Figura 5.10(b)):

1. Cálculo del tamaño de celda DMF en longitud:

- a) Para cada fila de la matriz IGM, recorrer todas las columnas y realizar el cálculo $IGM[columna][fila][0] - IGM[columna + 1][fila][0]$. El resultado es una tabla matricial de distancias en longitudes.
- b) Realizar el histograma de la tabla de distancias en longitudes, en donde se indica, para cada valor de distancia, su frecuencia.
- c) El valor más frecuentado se considera el tamaño de celda DMF en longitud.

2. Cálculo del tamaño de celda DMF en latitud:

- a) Para cada columna de la matriz IGM, recorrer todas las filas y realizar el cálculo $IGM[columna][fila][1] - IGM[columna][fila + 1][1]$. El resultado es una tabla matricial de distancias en latitudes.
- b) Realizar el histograma de la tabla de distancias en latitudes, en donde se indica, para cada valor de distancia, su frecuencia.
- c) El valor más frecuentado se considera el tamaño de celda DMF en latitud.

El resultado proporcionado por la metodologías será siempre un tamaño de celda DMF independiente tanto para la longitud como para la latitud, lo que implica que no sea necesariamente cuadrado.

El hecho de utilizar un procedimiento basado en distancias entre mediciones para calcular el tamaño de celda DMF de manera automática, garantiza que la mayoría de celdas DMF contengan únicamente un registro DMR. Esto puede ser beneficioso para el procesamiento de imágenes en dicho formato, ya que optimiza la localización de la información, reduciendo el número de celdas a *Null* y la longitud de listas DMR. Sin embargo, aunque se haya calculado un tamaño de celda DMF para una determinada imagen, y éste sea totalmente válido para construir su versión en formato DMF, hay que tener en cuenta las siguientes consideraciones:

- Las mediciones no tienen por qué haber sido adquiridas en dirección N/S, por lo que las distancias entre ellas sí pueden corresponder al tamaño de celda DMF en longitud y latitud (Figura 5.10(c)), pero éstas pueden no quedar distribuidas uniformemente. Se consigue de esta

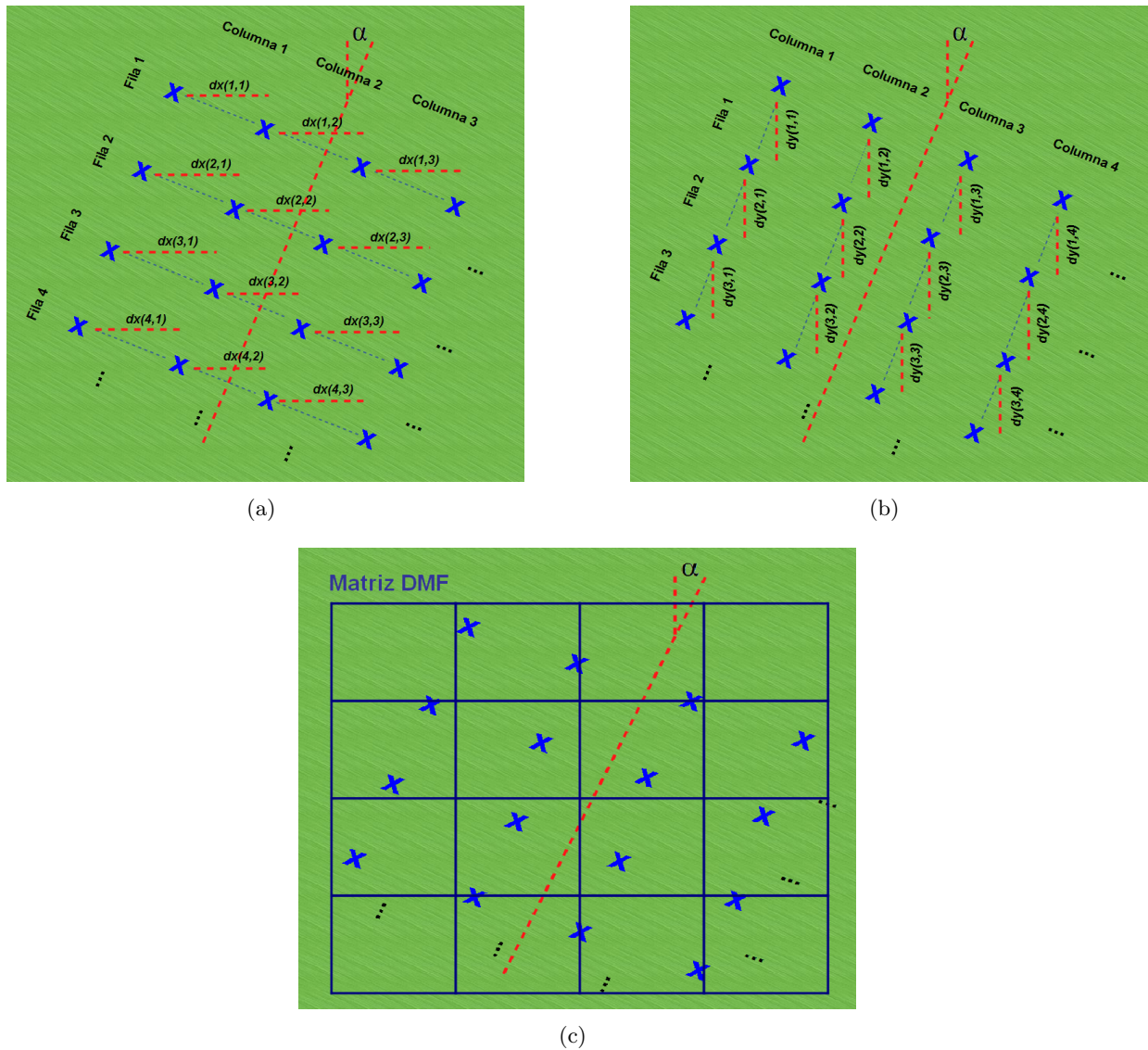


Figura 5.10: Proceso de estimación del tamaño de celda DMF.

(a) Cálculo de distancias en longitud entre mediciones en columnas. (b) Cálculo de distancias en latitud entre mediciones en filas. (c) Superposición de la matriz DMF con tamaño de celda DMF estimado.

manera una reducción en el número de celdas a *Null* y en la longitud de las listas DMR, pero esta reducción no es máxima. Para maximizar esta reducción, es necesario considerar las distancias entre los datos respecto al ángulo de vuelo (α) (Figuras 5.11(a) y 5.11(b)), aplicando la siguiente ecuación:

$$\begin{aligned} longitud' &= ((longitud - m_L)\cos(\alpha) - (latitud - m_l)\sin(\alpha)) + m_L \\ latitud' &= ((longitud - m_L)\sin(\alpha) + (latitud - m_l)\cos(\alpha)) + m_l \end{aligned} \quad (5.2)$$

Donde m_L y m_l son las longitudes y latitudes mínimas encontradas en el fichero IGM. De esta manera las distancias calculadas sí corresponden con la distribución espacial de los datos (Figura 5.11(c)).

Para realizar un cálculo automático del tamaño de celda DMF considerando el ángulo de vuelo es necesario modificar ligeramente el procedimiento, quedando optimizado tal y como se detalla a continuación:

1. Cálculo optimizado del tamaño de celda DMF en longitud:

- a) Para cada fila de la matriz IGM, recorrer todas las columnas y calcular los valores rotados de $IGM[columna][fila][0]$ e $IGM[columna + 1][fila][0]$ mediante la aplicación de la Ecuación 5.2, obteniendo $longitud_{\alpha}$ y $longitud_{\alpha}^{+1}$.
 - b) Realizar el cálculo $longitud_{\alpha} - longitud_{\alpha}^{+1}$. El resultado es una tabla matricial de distancias en longitudes.
 - c) Realizar el histograma de la tabla de distancias en longitudes, en donde se indica, para cada valor de distancia, su frecuencia.
 - d) El valor más frecuentado se considera el tamaño de celda DMF en longitud.
2. Cálculo optimizado del tamaño de celda DMF en latitud:
- a) Para cada columna de la matriz IGM, recorrer todas las filas y calcular los valores rotados de $IGM[columna][fila][1]$ e $IGM[columna][fila + 1][1]$ mediante la aplicación de la Ecuación 5.2, obteniendo $latitud_{\alpha}$ y $latitud_{\alpha}^{+1}$.
 - b) Realizar el cálculo $latitud_{\alpha} - latitud_{\alpha}^{+1}$. El resultado es una tabla matricial de distancias en longitudes.
 - c) Realizar el histograma de la tabla de distancias en latitudes, en donde se indica, para cada valor de distancia, su frecuencia.
 - d) El valor más frecuentado se considera el tamaño de celda DMF en latitud.
- Las distancias entre mediciones es mayor a medida que éstas se alejan del Nadir. De esta manera, las distancias más fiables nos las proporcionarían aquéllas entre las mediciones situadas dentro de esta región. Éste es un problema derivado de la geometría del sensor, y es imposible de resolver mediante la utilización de un tamaño de celda DMF estático. Posibles soluciones podrían involucrar la utilización de un tamaño de celda DMF dinámico, con tamaños de celda DMF menores para mediciones cercanas al Nadir (y menor distancias entre ellas), y tamaños de celda DMF mayores para mediciones alejadas del Nadir (y mayor distancias entre ellas).

5.3.3. Resultados del experimento.

Para analizar el funcionamiento de la metodología de cálculo automático del tamaño de celda DMF, se utilizó, en un primer lugar, las imágenes del *dataset sintético*, y en segundo lugar, las imágenes del *dataset real*.

Para empezar, el tamaño de celda DMF, tanto en longitud como en latitud, fue automáticamente calculado para todas las imágenes sintéticas siguiendo la metodología presentada anteriormente, y considerando todos los ángulos posibles de movimiento de plataforma *pitch*, *roll* y *yaw*. Los tamaños de celda DMF obtenidos han sido representados en la Figura 5.12, en donde la serie azul indica el tamaño de celda DMF en longitud, y la serie rosa el tamaño de celda DMF en latitud. Para una mejor comparativa, se ha incluido una serie de color negro que indica el tamaño de píxel proporcionado por el proveedor.

Las matrices DMF correspondientes al *dataset sintético* fueron construidas utilizando el tamaño de celda DMF calculado de manera automática. La Figura 5.13 muestra el aspecto visual de dichas matrices para varios ángulos de todos los movimientos de plataforma.

Para poder optimizar el procedimiento de creación de las matrices DMF, se consideró también el ángulo de vuelo respecto a la dirección N/S, por lo que el tamaño de celda DMF fue calculado de nuevo pero considerando este nuevo parámetro. Sin embargo, para los movimientos de plataforma *pitch* y *roll* la dirección del vuelo siempre fue N/S para todos los ángulos, así que el procedimiento optimizado se utilizó únicamente para todos los ángulos del movimiento de plataforma *yaw*. Para este último caso, el estudio del tamaño de celda DMF de acuerdo a la dirección de adquisición sí tiene sentido, pues es precisamente el ángulo respecto a la dirección N/S el ángulo variable para este tipo de movimiento (Figura 5.14).

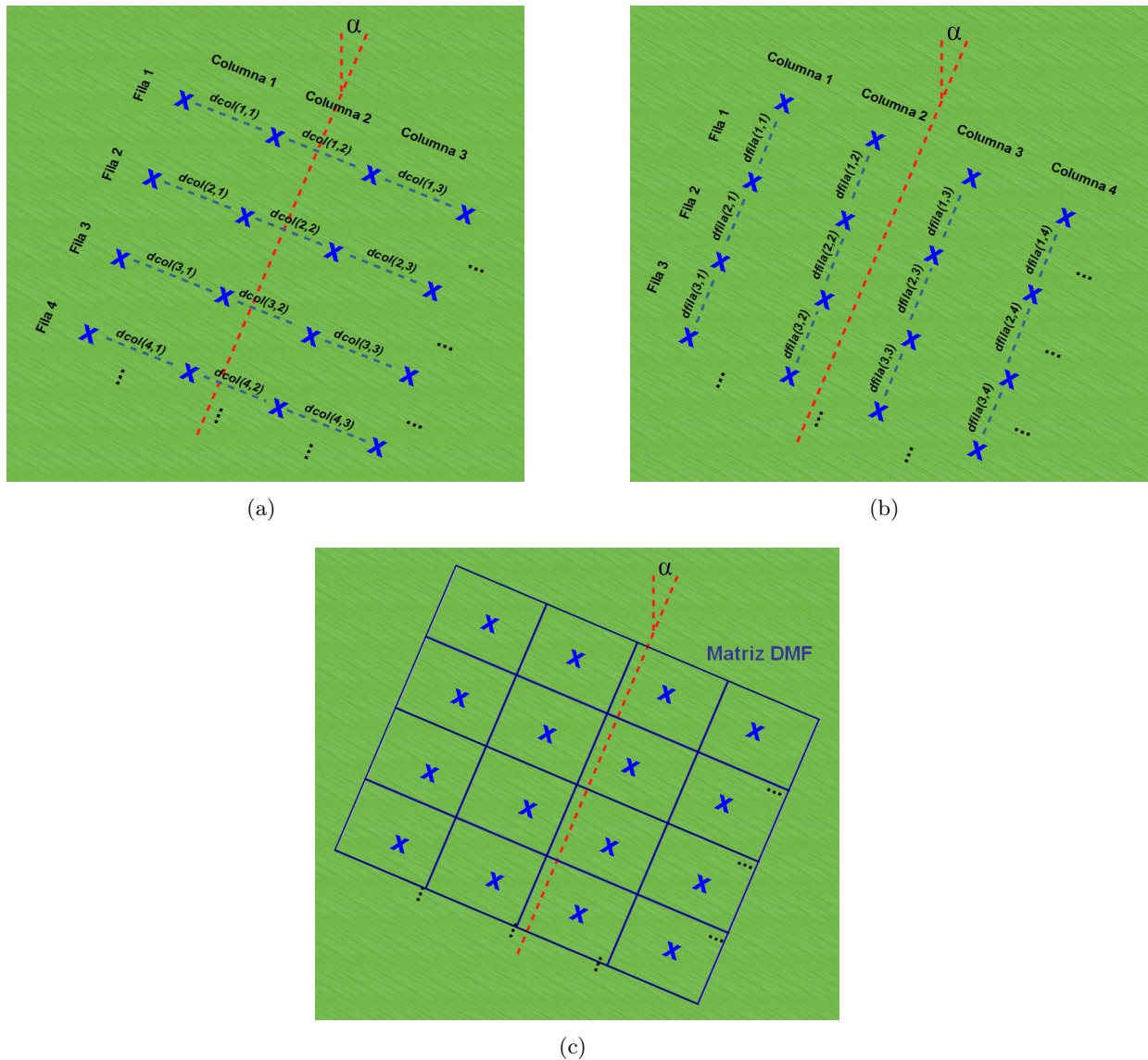


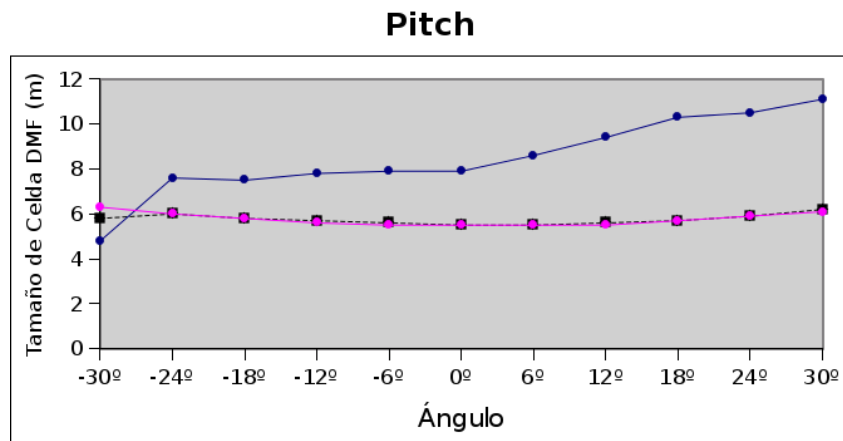
Figura 5.11: Proceso de estimación del tamaño de celda DMF (Optimizado).

- (a) Cálculo de distancias entre mediciones en columnas. (b) Cálculo de distancias entre mediciones en filas.
 (c) Superposición de la matriz DMF rotada con tamaño de celda DMF estimado.

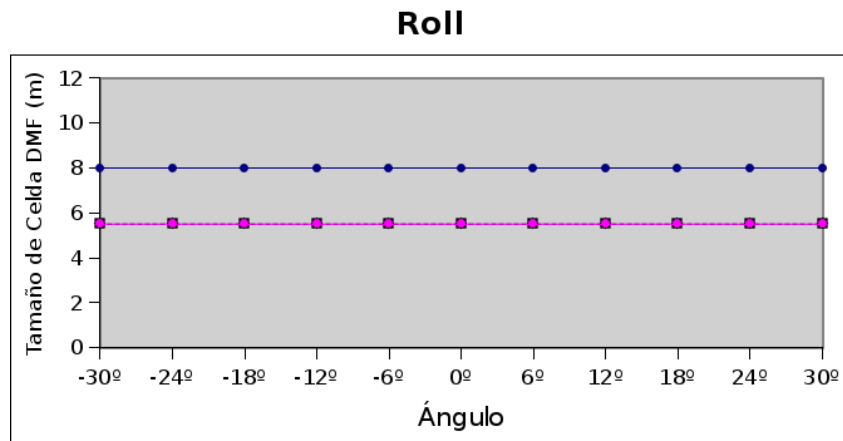
Al igual que con el procedimiento sin optimizar, el tamaño de celda DMF, tanto en longitud como en latitud, fue automáticamente calculado para todas las imágenes sintéticas, considerando todos los ángulos posibles del movimiento de plataforma *yaw*, así como el sentido del vuelo. Los tamaños de celda DMF obtenidos han sido representados en la Figura 5.15, en donde la serie azul indica el tamaño de celda DMF en longitud, la serie rosa el tamaño de celda DMF en latitud, y la serie negra el tamaño de píxel proporcionado por el proveedor.

El aspecto visual de las matrices DMF, correspondientes al movimiento de plataforma *yaw*, y resultantes de aplicar el tamaño de celda DMF mediante la metodología optimizada se presentan en la Figura 5.16.

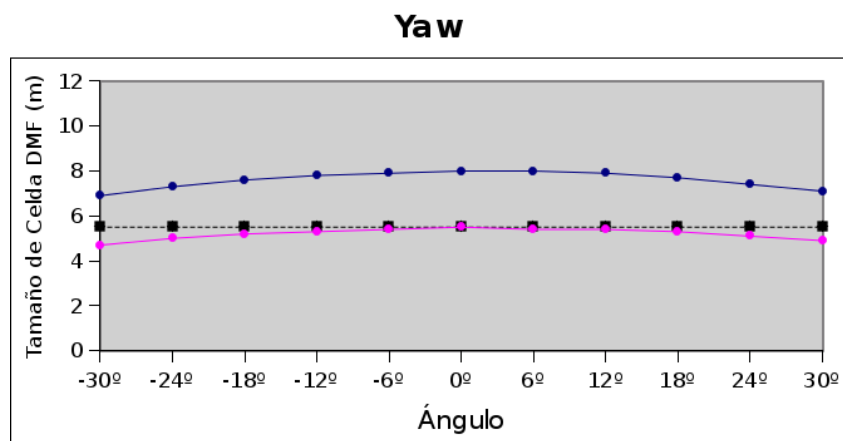
Mediante el uso de una metodología de cálculo automático del tamaño de celda DMF, basado en las distancias entre las mediciones tanto en longitud como en latitud, es posible realizar el supuesto de que, debido a sus características, garantiza una distribución de las mediciones sobre la matriz DMF de la manera más regular posible. Esto debe implicar que tras el cálculo automático, el número de registros DMR medio por celda DMF debe tender a 1 (una medición DMR por celda DMF), mientras que el porcentaje de celdas DMF a *Null* (sin medición) debe tender al 0%. Para estudiar este supuesto,



(a)



(b)



(c)

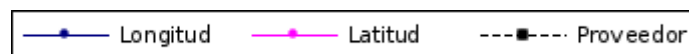


Figura 5.12: Tamaños de celda DMF en longitud y latitud proporcionados por la metodología de cálculo automático para cada uno de los movimientos de plataforma.

(a) Pitch. (b) Roll. (c) Yaw.

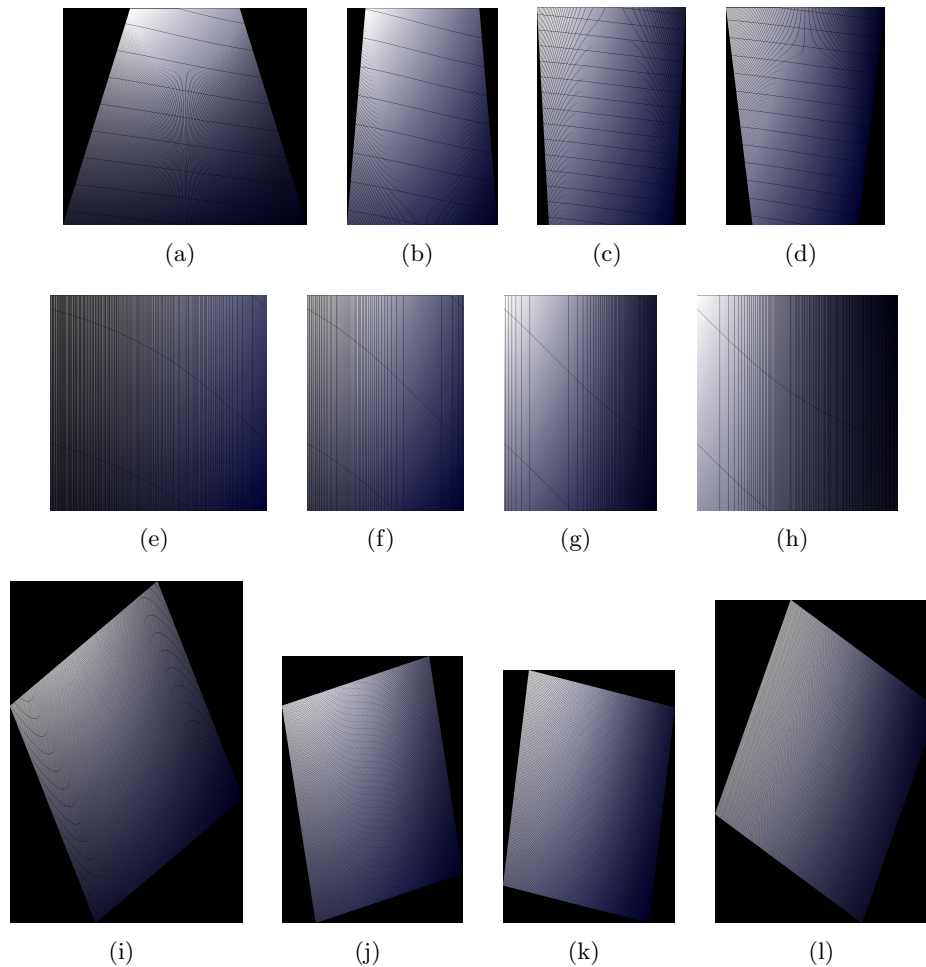


Figura 5.13: Visualización de las matrices DMF para el dataset de imágenes sintético, utilizando un tamaño de celda DMF calculado de manera automática (Banda 0).

(a) Pitch -30° . (b) Pitch -12° . (c) Pitch 12° . (d) Pitch 30° . (e) Roll -30° . (f) Roll -12° . (g) Roll 12° . (h) Roll 30° . (i) Yaw -30° . (j) Yaw -12° . (k) Yaw 12° . (l) Yaw 30° .

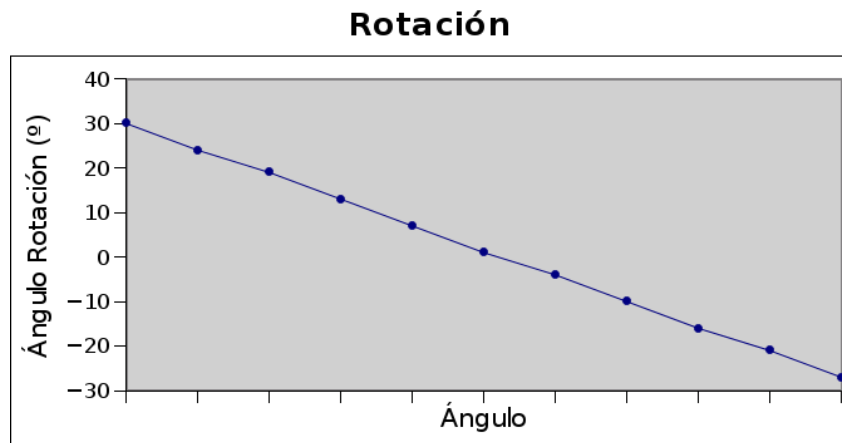


Figura 5.14: Ángulos de la dirección del vuelo para el movimiento de plataforma yaw.

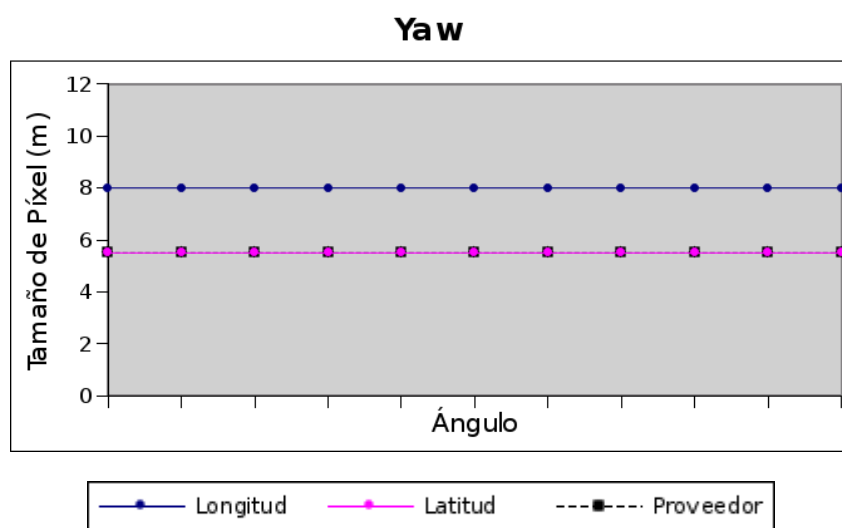


Figura 5.15: Tamaños de celda DMF en longitud y latitud proporcionados por la metodología de cálculo automático optimizada para el movimiento de plataforma yaw.

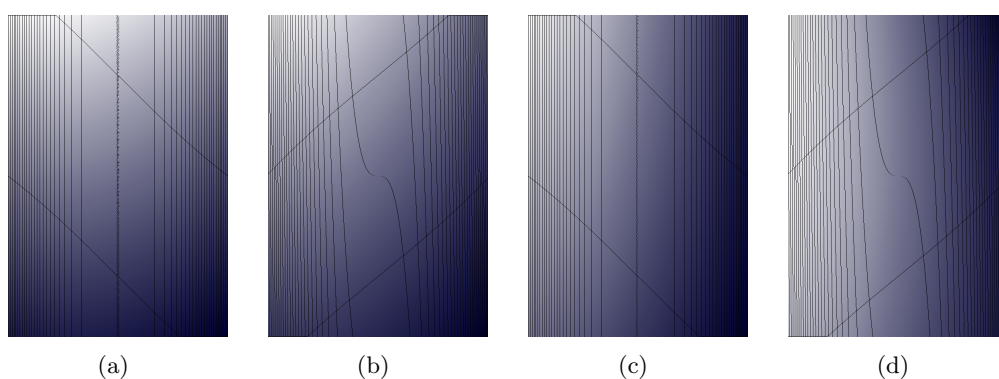
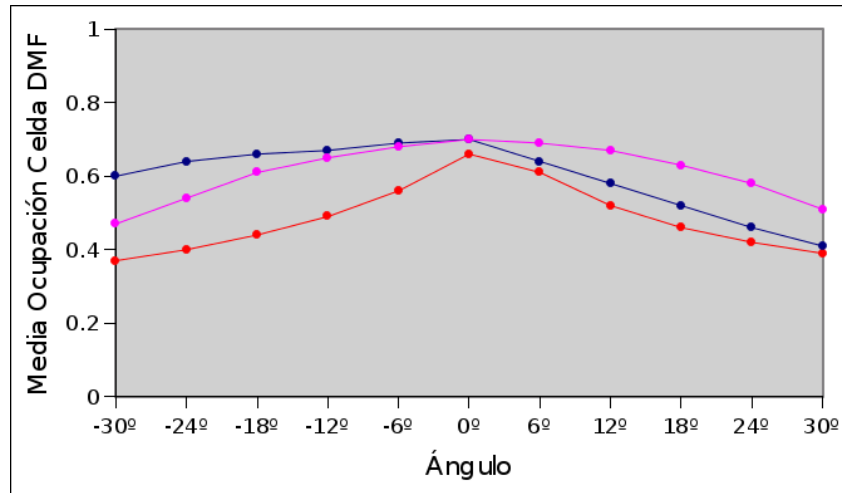


Figura 5.16: Visualización de las matrices DMF para el movimiento de plataforma yaw, utilizando un tamaño de celda DMF calculado de manera automática y considerando el sentido del vuelo (Banda 0).

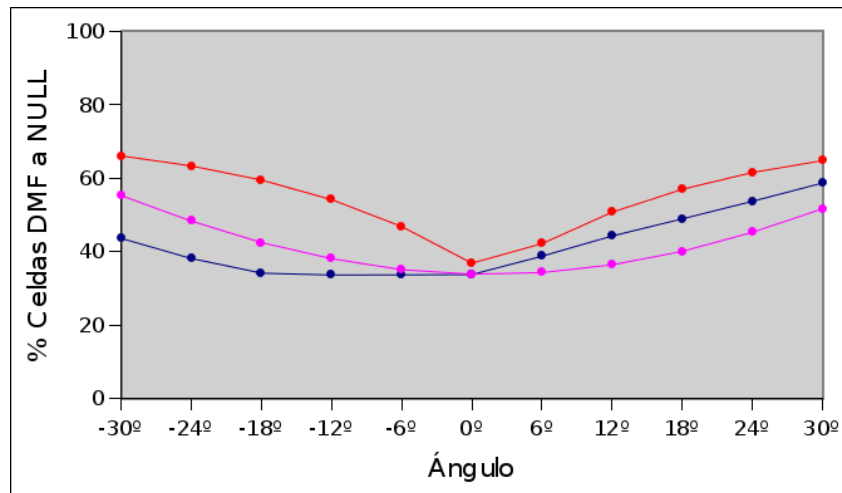
(a) Yaw -30° . (b) Yaw -12° . (c) Yaw 12° . (d) Yaw 30° .

se determinó el número medio de registros DMR almacenados por celda DMF, así como el porcentaje de celdas DMF a *Null*. En un principio, se consideraron estos dos parámetros para todos los ángulos de los tres tipos de movimientos de plataforma, pero considerando el tamaño de píxel proporcionado por el proveedor (sin cálculo automático). De esta manera, es posible observar cuál de los tres tipos de movimientos de plataforma produce de manera natural una peor distribución de los datos. La Figura

5.17 muestra los resultados obtenidos, teniendo en cuenta que la serie azul representa el movimiento *pitch*, la serie rosa el movimiento *roll*, y la serie roja el movimiento *yaw*.



(a)



(b)

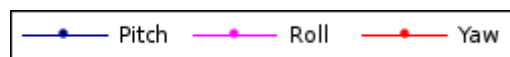


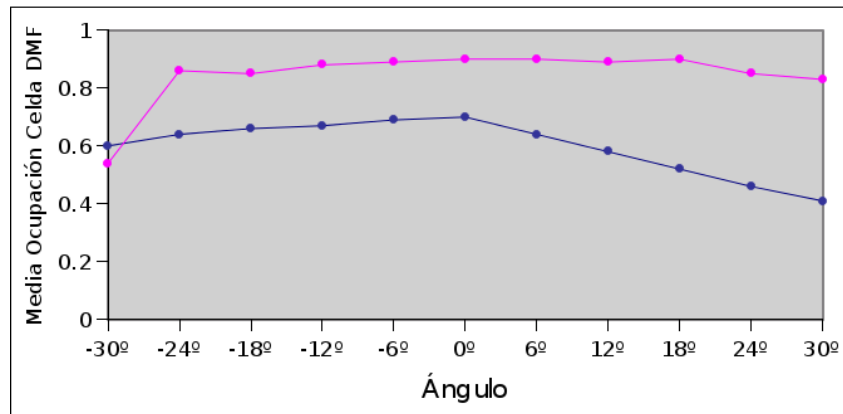
Figura 5.17: Estudio de la distribución de los registros DMR sobre la matriz DMF para todos los ángulos de los tres tipos de movimiento de plataforma.

(a) Número medio de registros DMR por celda DMF. (b) Porcentaje de celdas DMF a Null.

A continuación, el mismo estudio se realizó pero de manera independiente para cada uno de los movimientos de plataforma. El análisis se centró, en este caso, en cómo varía la distribución de los registros DMR sobre la matriz DMF correspondiente cuando se utilizan el tamaño de celda DMF proporcionado por el proveedor (*DMF*), el calculado de manera automática por el procedimiento previamente descrito (*DMF Automat.*), y el calculado de manera automática por este mismo procedimiento considerando el ángulo de adquisición de datos (*DMF Automat. Áng.*). Éste último caso aplicable únicamente al tipo de movimiento de plataforma *yaw*, ya que los otros dos restantes fueron realizados con una dirección de vuelo N/S. La Figura 5.18 y la Figura 5.19 muestran el número medio de registros DMR por celda DMF, así como el porcentaje de celdas DMF a *Null*, respectivamente.

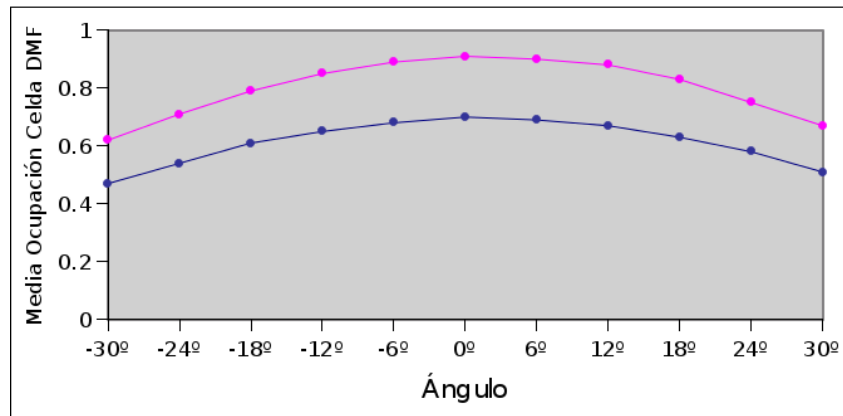
Por otro lado, todos estos mismos análisis fueron realizados a todas las imágenes del *dataset real*. En un primer lugar, el tamaño de celda DMF fue automáticamente calculado sin tener en cuenta el ángulo de vuelo (*Tam. Celda DMF Automat.*). Posteriormente, dicho ángulo fue considerado, y

Pitch



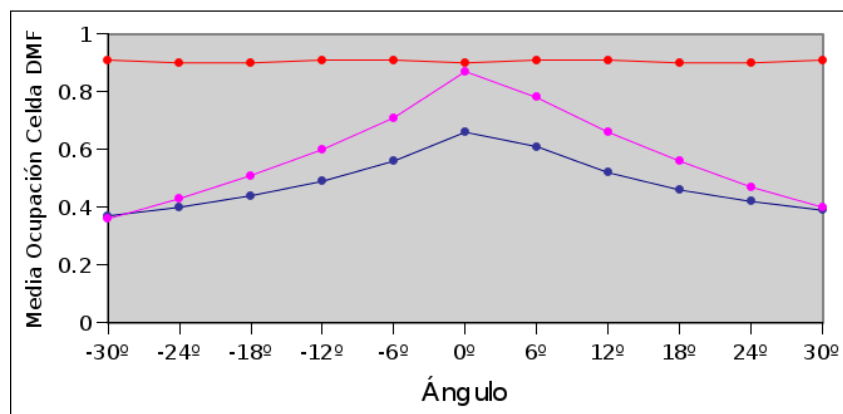
(a)

Roll



(b)

Yaw



(c)

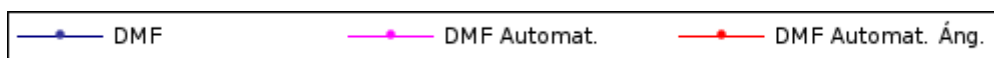
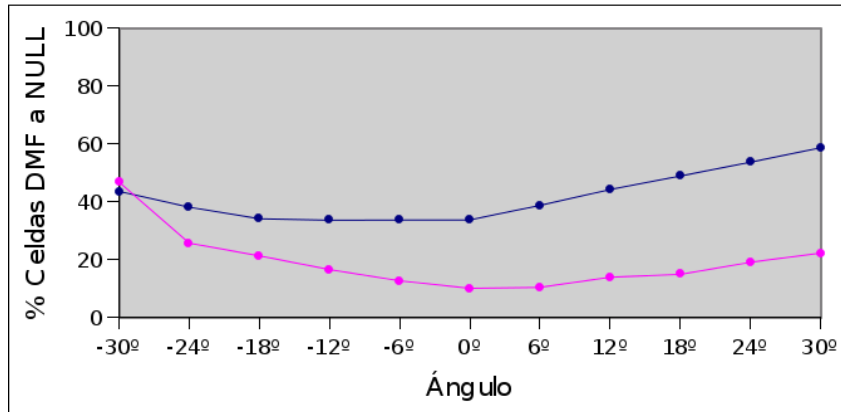


Figura 5.18: Estudio del número medio de registros DMR por celda DMF, para todos los ángulos de los tres tipos de movimiento de plataforma de acuerdo al tamaño de celda DMF.

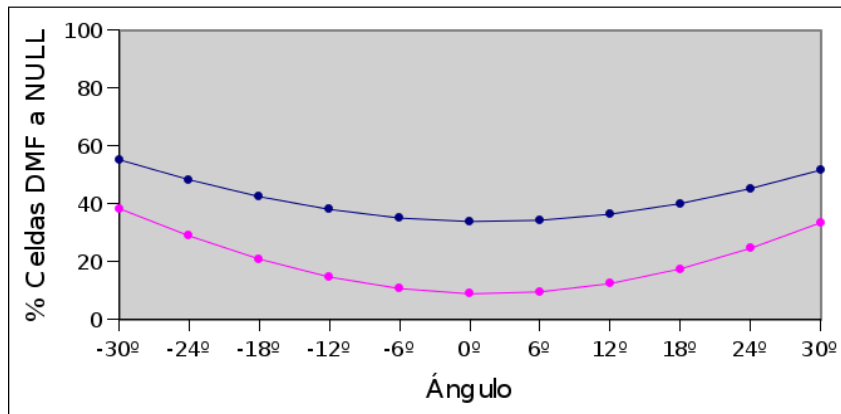
(a) Pitch. (b) Roll. (c) Yaw.

Pitch



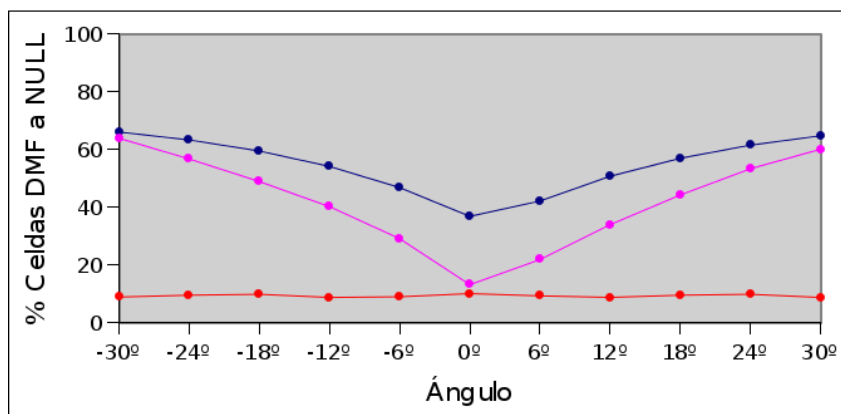
(a)

Roll



(b)

Yaw



(c)

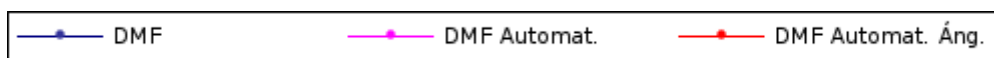


Figura 5.19: Estudio del porcentaje de celdas DMF a Null, para todos los ángulos de los tres tipos de movimiento de plataforma de acuerdo al tamaño de celda DMF.

(a) Pitch. (b) Roll. (c) Yaw.

se calculó nuevamente el tamaño de celda DMF mediante el procedimiento optimizado (*Tam. Celda DMF Automat. Áng.*). Los tamaños de celda DMF obtenidos en ambos casos, así como el ángulo de vuelo, han sido representados en la Tabla 5.3. Además, aunque ya fue presentado en la Tabla 5.2, se ha añadido el tamaño de píxel proporcionado por el proveedor para una mejor comparativa (*Tam. Píxel Prov.*).

	Cuprite	Barrax	Mediterráneo	GMéxico	Cantera
<i>Tam. Píxel Prov.</i>	9.8 m	0.000315°	0.009411°	0.000037°	4.4 m
<i>Tam. Píxel Automat. (Long.)</i>	9.6 m	0.00046°	0.012°	0.000036°	3.5 m
<i>Tam. Píxel Automat. (Lat.)</i>	11 m	0.0003°	0.009°	0.00003°	4.2 m
<i>Ángulo Vuelo</i>	-9°	-8°	-8°	4°	-73°
<i>Tam. Píxel Automat. Áng. (Long.)</i>	9.3 m	0.00046°	0.012°	0.000036°	4.4 m
<i>Tam. Píxel Automat. Áng. (Lat.)</i>	10.8 m	0.00031°	0.009°	0.000036°	3.7 m

Tabla 5.3: Tamaños de celda DMF calculados de manera automática por la metodología y ángulos de vuelo para las imágenes del dataset real.

La Figura 5.20 muestra el aspecto visual de las matrices DMF resultantes al aplicar el tamaño de celda DMF obtenido, primero sin considerar el ángulo de vuelo, y posteriormente, considerándolo.

Del mismo modo en el que se realizó para el *dataset sintético*, para el *dataset real* también se analizó el número medio de registros DMR por celda DMF, así como el porcentaje de celdas DMF a *Null*. Para comprobar la eficiencia de la metodología, ambos parámetros fueron determinados tanto para el tamaño de píxel proporcionado por el proveedor (*Tam. Píxel Prov.*), como para el calculado de manera automática sin considerar el ángulo de vuelo (*Tam. Celda DMF Automat.*), y posteriormente considerando dicho ángulo (*Tam. Celda DMF Automat. Áng.*). Los resultados de este análisis se muestran en la Figura 5.21.

5.3.4. Análisis de los resultados del experimento.

En este apartado se ha diseñado e implementado un procedimiento para el cálculo automático del tamaño de celda DMF, a partir de la disposición de las mediciones en latitud y longitud, obteniendo los siguientes resultados:

1. Aplicación del procedimiento para el *dataset sintético*:
 - a) Cálculo automático del tamaño de celda DMF en latitud y longitud (Figura 5.12).
 - b) Análisis visual de las matrices DMF tras aplicar el tamaño de celda calculado de manera automática (Figura 5.13).
 - c) Cálculo automático del tamaño de celda DMF en latitud y longitud, considerando el ángulo de vuelo (Figura 5.15).
 - d) Análisis visual de las matrices DMF tras aplicar el tamaño de celda calculado de manera automática, considerando el ángulo de vuelo (Figura 5.16).
 - e) Análisis de la distribución de los datos en la matriz DMF tras aplicar el tamaño de celda calculado de manera automática (Figuras 5.17, 5.18 y 5.19).
2. Aplicación del procedimiento para el *dataset real*:
 - a) Cálculo automático del tamaño de celda DMF en latitud y longitud, considerando y sin considerar ángulo de vuelo (Tabla 5.3).
 - b) Análisis visual de las matrices DMF tras aplicar el tamaño de celda calculado de manera automática, considerando y sin considerar ángulo de vuelo (Figura 5.20).

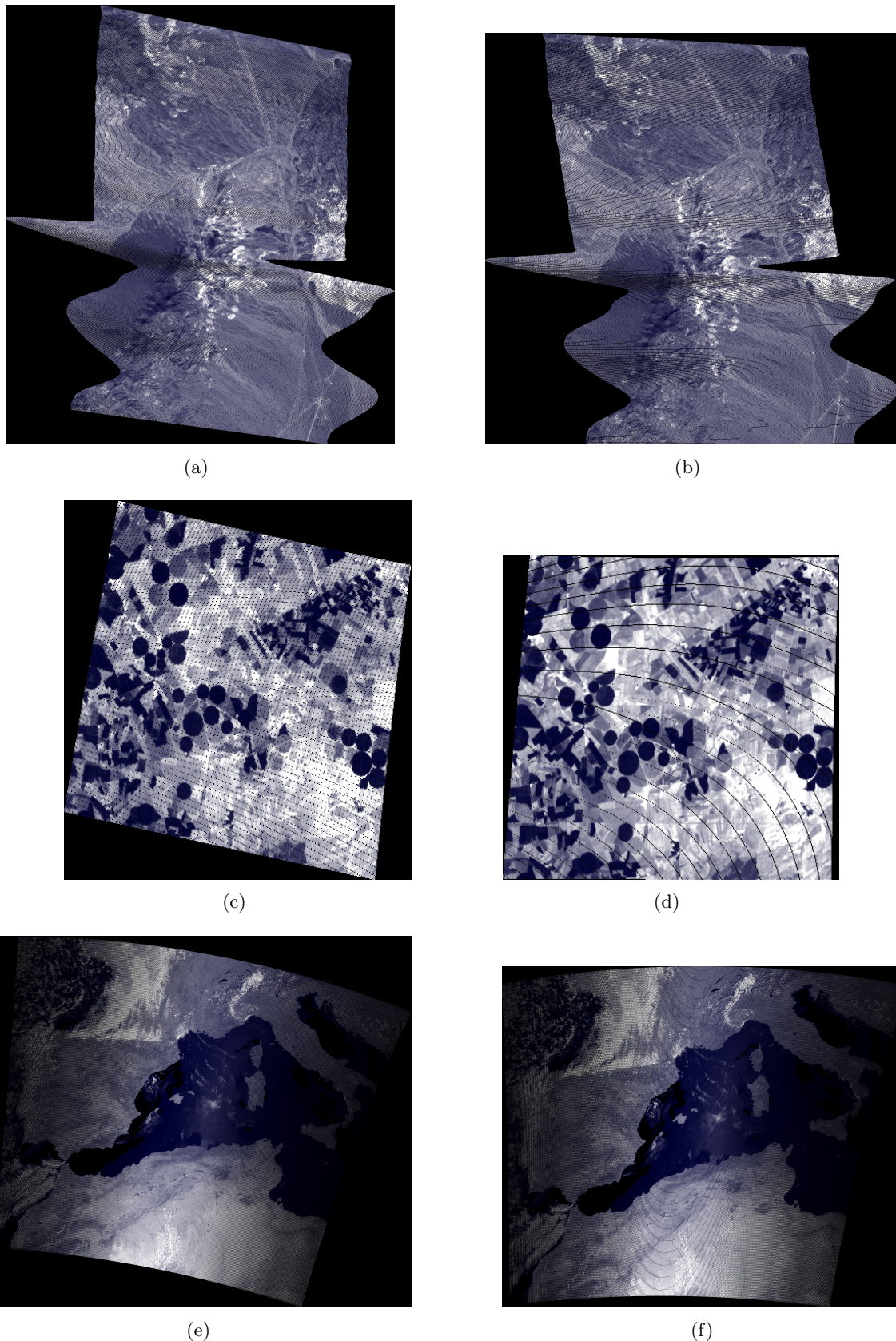
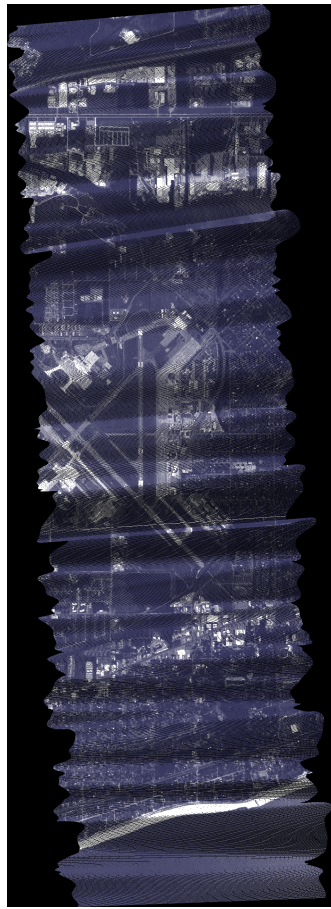
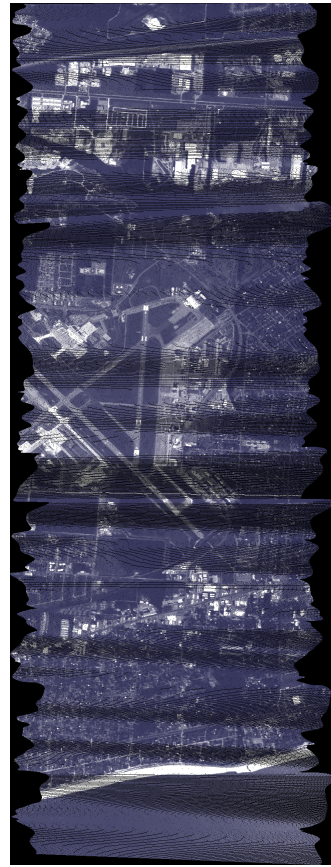


Figura 5.20: Visualización de las matrices DMF para el dataset de imágenes real, aplicando un tamaño de celda DMF calculado de manera automática por la metodología.

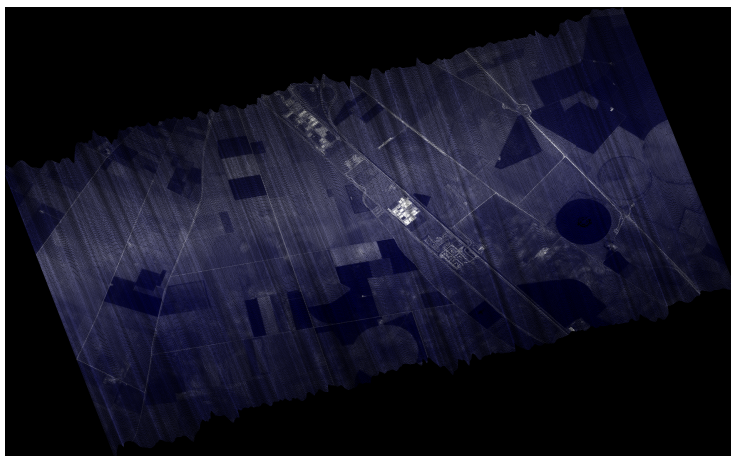
(a) Escena Cuprite [Banda 10], sin considerar ángulo de vuelo. (b) Escena Cuprite [Banda 10], considerando ángulo de vuelo. (i) Escena Barrax [Banda 0], sin considerar ángulo de vuelo. (j) Escena Barrax [Banda 0], considerando ángulo de vuelo. (e) Escena Mediterráneo [Banda 4], sin considerar ángulo de vuelo. (f) Escena Mediterráneo [Banda 4], considerando ángulo de vuelo. (g) Escena GMéxico [Banda 10], sin considerar ángulo de vuelo. (h) Escena GMéxico [Banda 10], considerando ángulo de vuelo. (c) Escena Cantera [Banda 5], sin considerar ángulo de vuelo. (d) Escena Cantera [Banda 5], considerando ángulo de vuelo.



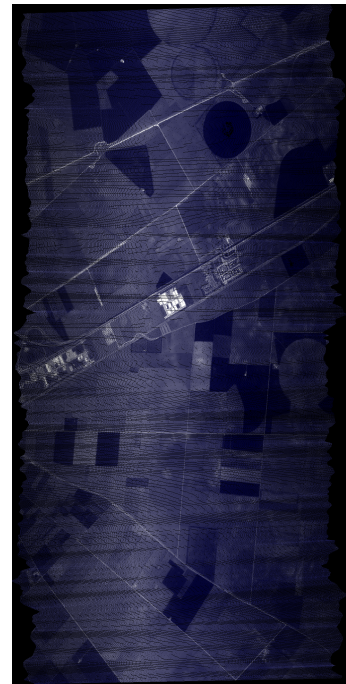
(g)



(h)

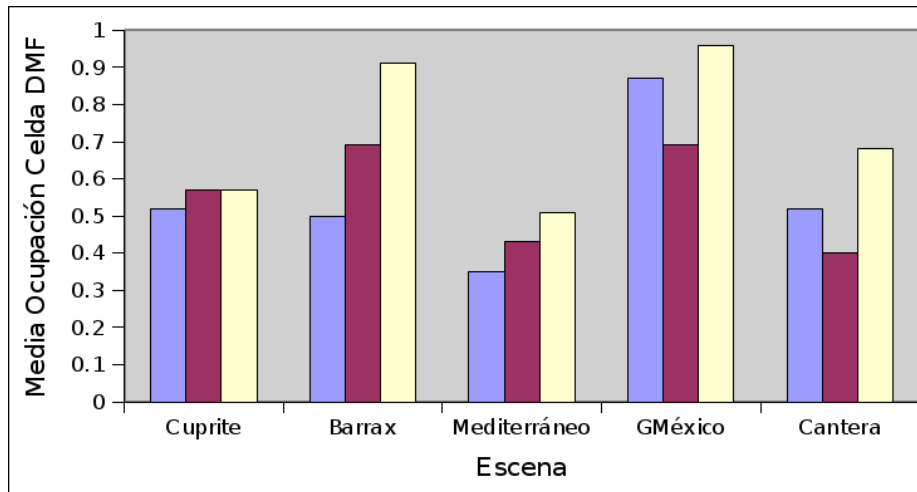


(i)

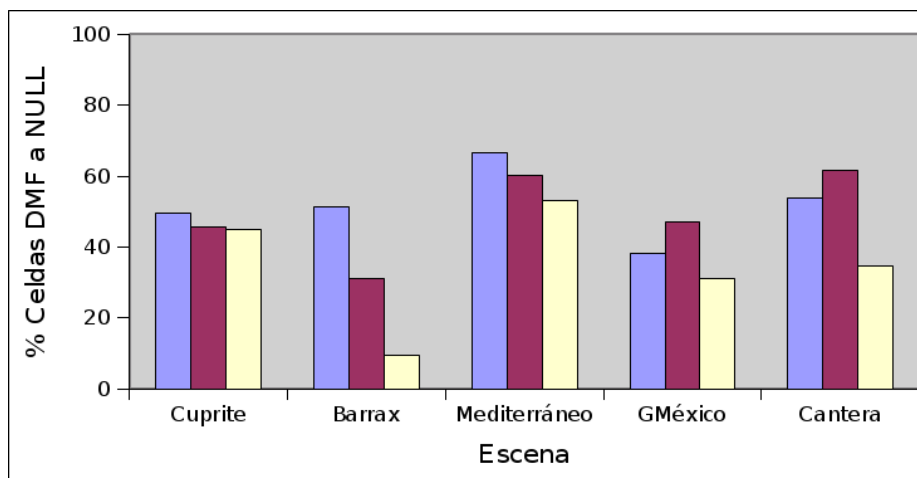


(j)

Figura 5.20: (Cont.)



(a)



(b)



Figura 5.21: Estudio de la distribución de los registros DMR sobre la matriz DMF para todas las imágenes del dataset real.

(a) Número medio de registros DMR por celda DMF. (b) Porcentaje de celdas DMF a Null.

- c) Análisis de la distribución de los datos en la matriz DMF tras aplicar el tamaño de celda calculado de manera automática, considerando y sin considerar ángulo de vuelo (Figura 5.21).

Con este experimento se ha conseguido estimar de manera automática el tamaño de celda DMF a partir de la disposición espacial de los datos adquiridos por el sensor a bordo del avión o satélite. Se trata de un método basado en el estudio de las distancias entre mediciones, en el cual la distancia con más frecuencia es seleccionada como tamaño de celda DMF, garantizando que a un gran porcentaje de las celdas DMF les será asignada una única medición, disminuyendo a su vez el número de celdas a *Null*.

El tamaño de celda DMF en longitud no tiene por qué coincidir con el tamaño en latitud, ya que las distancias entre las mediciones, si se considera un vuelo en sentido N/S, quedan principalmente determinadas en longitud por las características y geometría del sensor, y en latitud por la velocidad del avión o satélite. Además, también entran en juego diferentes parámetros, como las turbulencias, altura de la plataforma, etc., por lo que el cálculo automático debe ser realizado de manera independiente tanto para las longitudes como para las latitudes. De esta forma, la matriz DMF resultante puede

aparecer ligeramente deformada al utilizar diferentes escalas en los ejes X e Y. Esto se puede comprobar en las matrices DMF correspondientes a los datasets sintético y real (Figuras 5.13, 5.16 y 5.20). No obstante, para el caso del formato DMF, y como se comentó en la Sección 5.2, la representación de la matriz DMF no debería ser relevante y únicamente ha de ser utilizada para visualizar la disposición de los datos. Desde el punto de vista del procesamiento, esta deformación espacial puede ser beneficiosa ya que el hecho de garantizar una medición por celda DMF implica un menor número de celdas a *Null* y una reducción en la longitud de las listas DMR.

Con el método de cálculo automático del tamaño de celda DMF propuesto, las distancias se realizan tanto en longitud como en latitud. Sin embargo, la disposición espacial de los datos no es regular si la dirección del vuelo no se realizó en sentido N/S, dirección en la que está orientada la matriz DMF. Si se desea una correcta disposición de los datos, basta con calcular la distancia Euclídea entre las mediciones en el sentido de adquisición de los datos y en el sentido del vuelo, y superponer la matriz DMF girada el ángulo de vuelo respecto a la dirección N/S. De esta manera, las mediciones adquiridas por el sensor se distribuyen de manera regular por la matriz DMF. Esta modificación de la metodología, por tanto, permite la construcción de una matriz DMF más optimizada respecto a la generada mediante la primera versión del procedimiento.

Para comprobar la eficiencia del procedimiento de cálculo del tamaño de celda DMF, se analizaron dos parámetros. El primero de ellos es el número medio de registros DMR por celda DMF, el cual debe aproximarse a 1 a medida que se realiza una mejor estimación de dicho tamaño. El segundo es el porcentaje de celdas DMF a *Null*, el cual debe aproximarse al 0% a medida que se realiza una mejor estimación del tamaño de celda DMF. En primer lugar, se realizó un análisis de cuál de los tres movimientos de plataforma supone una peor distribución de las mediciones (Figura 5.17), teniendo en cuenta el tamaño de píxel proporcionado por el proveedor, el calculado de manera automática por el procedimiento propuesto, y el calculado de manera automática por este mismo procedimiento pero considerando el ángulo de vuelo respecto a la dirección N/S. En dichas gráficas, cuanto menor sea la tendencia a 1 registro DMR por celda, y cuanto menor sea la tendencia a un porcentaje de 0% en número de celdas DMF a *Null*, peor será la distribución de las mediciones con el tamaño de celda DMF seleccionado. Por tanto, teniendo en cuenta esto, es posible comprobar que una alteración en cualquiera de los tres tipos de movimiento supone una peor distribución de las mediciones, que será mayor cuanto mayor sea el ángulo de incidencia. Además, también se puede sacar la conclusión de que el movimiento de plataforma que supone una peor distribución de las mediciones es el de tipo *yaw*. Es decir, a la hora de georreferenciar los datos, siempre será peor que el avión cambie el sentido del vuelo a que sufra aleteos o cabeceos. Otra observación que es posible denotar es que, para el tipo de movimiento *pitch*, la distribución de las mediciones es siempre peor si el avión tiene una trayectoria ascendente a si la tiene descendente. Esto se produce porque, a medida que el avión coge altura, existe más distancia entre las mediciones, a lo que hay que añadir que los efectos producidos por la geometría del sensor quedan potenciados.

Considerando estos dos mismos parámetros, pero para cada movimiento de plataforma de manera independiente, es posible determinar tras analizar los resultados tanto para el *dataset sintético* (Figura 5.18 y Figura 5.19) como el para el *dataset real* (Figura 5.21), que la tendencia a 1 registro DMR por celda DMF, y la tendencia a un porcentaje de 0% de celdas DMF a *Null* es mayor cuando se estima de manera automática el tamaño de celda DMF para la longitud y para la latitud. Esto garantiza una mejor distribución de las mediciones sobre la matriz DMF posibilitando un rápido, eficaz y más preciso procesamiento. Esta tendencia se potencia si además se considera y se distribuyen los datos de acuerdo al ángulo del vuelo. De hecho, éste último parámetro debería ser fundamental, pues, como se puede observar en la Figura 5.21(a) y en la Figura 5.21(b), para las escenas *GMéxico* y *Cantera*, si el ángulo de vuelo no es considerado, no siempre se consigue una mejora en la distribución de los registros DMR. Esto se produce mayormente por la cantidad de celdas DMF a *Null* que aparecen al considerar una escena rotada, produciendo un incremento de celdas DMF fuera de escena como bordes, que es necesario representar.

5.4. Espacio en disco.

Este experimento pretende estudiar el tamaño de almacenamiento en disco requerido por el formato DMF, respecto a los formatos de almacenamiento de imágenes convencionales.

5.4.1. *Objetivos del experimento.*

Es muy importante analizar la eficiencia del formato DMF a la hora de almacenar los datos adquiridos por el sensor en un dispositivo externo, ya que ese es el modo en el que ha de ser ofrecido a la comunidad científica. Generalmente, las imágenes hiperespectrales geocorregidas utilizadas por los expertos suelen ser imágenes de gran tamaño (pueden llegar al orden de Gigabytes) y bastante pesadas de procesar. Como se ha demostrado en el Capítulo 3, gran cantidad de los datos es información redundante, así como los píxeles fuera de la escena son representados por espectros a 0, consumiendo también espacio de memoria. Sin embargo, el formato DMF únicamente almacena los datos tal y como fueron adquiridos por el sensor, reduciendo la información a almacenar al mínimo.

De esta manera, con este experimento se intentan abarcar los siguientes objetivos:

- Estudiar la eficiencia del formato DFM respecto al consumo de espacio de disco para los diferentes modos de almacenamiento.
- Comparación del tamaño en disco del formato DMF respecto a diferentes formatos de almacenamiento convencionales.

5.4.2. *Descripción del experimento.*

El análisis del consumo de espacio en disco fue realizado para todas las imágenes tanto del *dataset sintético*, como del *dataset real*. Cada imagen, de manera independiente, fue geocorregida mediante procedimientos convencionales y almacenada en disco, considerando tanto el formato estándar BIP (cuyo tamaño es el mismo que para los formatos BSQ y BIL), así como sus transformaciones a los formatos GeoTIFF, HDF, CDF y NetCDF, los cuales fueron detallados en la Sección 3.2.6.3. Posteriormente, para cada imagen, se construyeron las matrices DMF correspondientes a partir de los ficheros L0 e IGM, y fueron almacenados en disco siguiendo las tres políticas presentadas en la Sección 4.1.1.3:

1. Almacenamiento de cada celda DMF con su cabecera previa (de tamaño 1 Byte) correspondiente y sus registros DMR (*DMF1*).
2. Almacenamiento una máscara cabecera binaria y las celdas DMF ocupadas (diferentes a *Null*), con su cabecera previa (de tamaño 1 Byte) correspondiente y sus registros DMR (*DMF2*).
3. Almacenamiento de los registros DMR de manera secuencial (*DMF3*).

Tras esto, y teniendo en cuenta que en muchos casos la imagen geocorregida es rotada para minimizar el número de píxeles fuera de la escena, y por tanto, el consumo de espacio en disco, las versiones geocorregidas de las imágenes fueron rotadas respecto al ángulo de vuelo. Para el caso del *dataset sintético*, únicamente las imágenes correspondientes al movimiento de plataforma *yaw*, pues el resto de movimientos ya siguen la dirección N/S. Aparte de haber sido rotadas y orientadas en la dirección N/S, para el caso de almacenamiento DMF también se estimaron, de manera automática, los tamaños de celda DMF, garantizando de esta manera una máxima tendencia de un registro DMR por

celda DMF y una minimización en el porcentaje de celdas DMF a *Null*. Los ángulos de vuelo fueron presentados en la Tabla 5.3.

De esta manera, y en resumen, para todas las imágenes de cada dataset se realizará una comparación del tamaño en disco consumido en sus versiones DMF (*DMF1*, *DMF2* y *DMF3*) y geocorregida por métodos convencionales (BSQ/BIL/BIP, GeoTIFF, HDF, CDF y NetCDF) sin considerar el ángulo de vuelo, y posteriormente considerándolo, junto con un cálculo automático del tamaño de celda DMF para el formato propuesto.

5.4.3. Resultados del experimento.

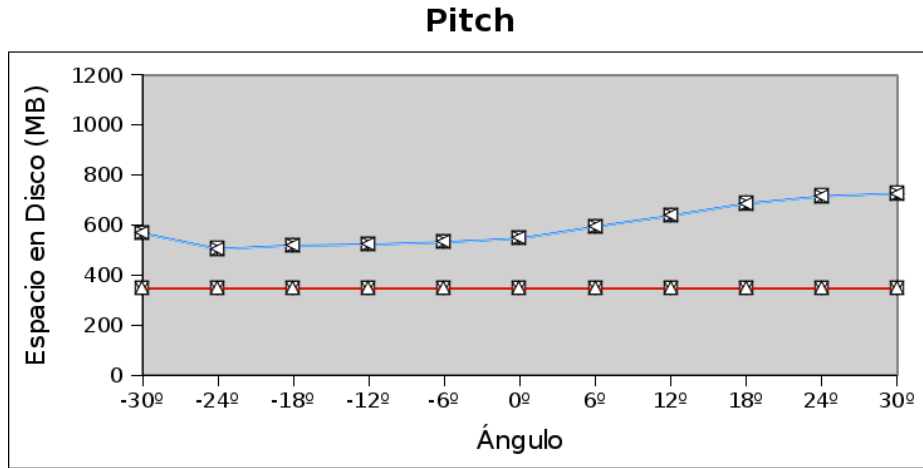
En primer lugar, se realizó una comparación de los tamaños de los ficheros originados para el *dataset sintético*, considerando las versiones de almacenamiento DMF (*DMF1*, *DMF2* y *DMF3*) y los formatos convencionales BSQ/BIL/BIP, GeoTIFF, HDF, CDF y NetCDF. La Figura 5.22 muestra el consumo de espacio en disco para dicho dataset, de acuerdo a los tres tipos diferentes de movimientos de plataforma (*pitch*, *roll* y *yaw*) y considerando el rango de ángulos $[-30^\circ, 30^\circ]$. En dicha figura, las series correspondientes a los formatos de datos convencionales poseen diferentes tonalidades de azul, mientras que los correspondientes al formato DMF poseen diferentes tonalidades de rojo.

La Figura 5.22 muestra cómo se comporta el tamaño en disco a medida que el ángulo varía para cada movimiento de plataforma. Sin embargo, no es apreciable con exactitud en cuánto difiere dicho tamaño de acuerdo al formato utilizado. Para una mejor visualización de este parámetro, la Tabla 5.4 muestra la diferencia de tamaño (en MB) entre dichos formatos respecto a *DMF3*, ya que éste almacena menos información adicional (presentando el menor consumo de espacio en disco de entre todos los formatos evaluados), y coincidiendo dicho tamaño con el de los datos originales recogidos por los sensores (información L0 e IGM). Para una mejor y rápida visualización (en color azul), a dicha tabla se le ha añadido la diferencia de tamaño media presentados por todos los formatos para cada movimiento de plataforma (\overline{dif}), así como el porcentaje de incremento en tamaño de disco que supone utilizar cada uno de los formatos respecto a la referencia *DMF3* ($\overline{\Delta}\%$).

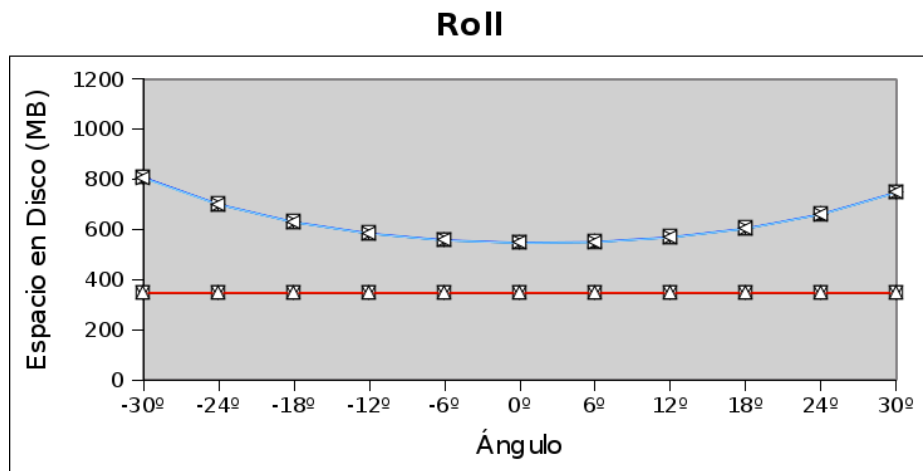
El mismo procedimiento fue realizado para las imágenes geocorregidas considerando el ángulo de vuelo, situando la la dirección de adquisición de datos como N/S. Esto permite la reducción del número de píxeles con un valor de espectro a 0 (bordes), y por tanto, un decremento considerable del tamaño en disco. Para una justa comparación, se consideró dicho ángulo también al construir la matriz DMF y en su posterior almacenamiento, así como un tamaño de celda DMF calculado de manera automática para maximizar la tendencia a un registro DMR por celda DMF y minimizar el número de celdas a *Null*. Para el caso del *dataset sintético*, únicamente el movimiento de plataforma *yaw* presenta direcciones de vuelo diferentes a N/S, por lo que se analizaron los tamaños de disco para este caso. Una previsualización de las imágenes geocorregidas en dirección N/S mediante procedimientos convencionales se muestran en la Figura 5.23. Los resultados obtenidos respecto al consumo de espacio en disco, para este caso, se muestran en la Figura 5.24. Para visualizar de manera detallada en cuánto difiere el almacenamiento para cada uno de los formatos, se han representado en la Tabla 5.5 la diferencia de tamaño media (\overline{dif}) y el porcentaje de incremento en tamaño de disco ($\overline{\Delta}\%$), teniendo como referencia, de igual manera que en los análisis anteriores, la política de almacenamiento *DMF3*.

Posteriormente, el mismo tipo de análisis fue también realizado para el *dataset real*. En primer lugar, y tal como se realizó para el *dataset sintético*, se realizó una comparación del tamaño en disco para cada una de las imágenes, considerando las versiones de almacenamiento DMF (*DMF1*, *DMF2* y *DMF3*) y las correspondientes geocorregidas convertidas a los diferentes formatos convencionales BSQ/BIL/BIP, GeoTIFF, HDF, CDF y NetCDF. La Figura 5.25 muestra el consumo de espacio en disco requerido para este dataset.

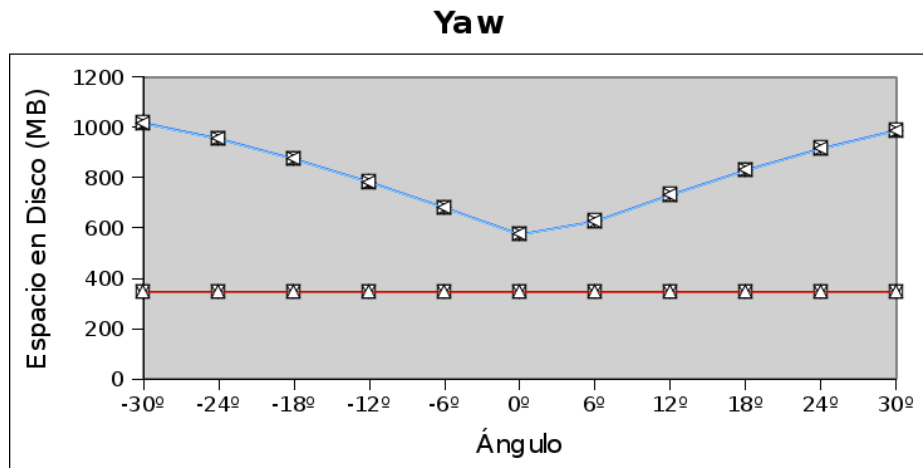
Tal y como ocurría para el caso del *dataset sintético*, resulta interesante representar en detalle tanto las diferencias de tamaño en disco respecto al formato de almacenamiento referencia *DMF3* (*dif*),



(a)



(b)



(c)

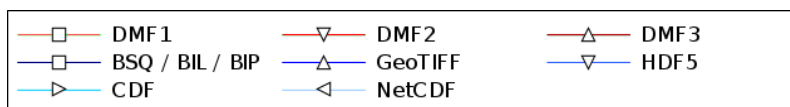


Figura 5.22: Tamaños de disco de los formatos de almacenamiento DMF y convencional para cada uno de los movimientos de plataforma.

(a) Pitch. (b) Roll. (c) Yaw.

Pitch							
	DMF1	DMF2	BSQ/BIL/BIP	GeoTIFF	HDF5	CDF	NetCDF
-30°	+0.63	+0.43	+220.16	+221.66	+223.06	+220.16	+220.16
-24°	+0.6	+0.45	+157.16	+158.56	+159.86	+157.16	+157.16
-18°	+0.58	+0.45	+171.16	+172.56	+173.86	+171.16	+171.16
-12°	+0.56	+0.44	+174.86	+176.16	+177.56	+174.86	+174.86
-6°	+0.55	+0.43	+184.36	+185.66	+187.06	+184.36	+184.36
0°	+0.55	+0.43	+199.26	+200.66	+201.96	+199.26	+199.26
6°	+0.59	+0.44	+245.96	+247.36	+248.66	+245.96	+245.96
12°	+0.66	+0.45	+290.16	+291.56	+292.86	+290.16	+290.16
18°	+0.74	+0.46	+338.76	+340.06	+341.46	+338.76	+338.76
24°	+0.82	+0.48	+367.36	+368.76	+370.06	+367.36	+367.36
30°	+0.92	+0.5	+379.06	+380.46	+381.76	+379.06	+379.06
<i>dif</i>	+0.75	+0.46	+400.76	+402.16	+403.46	+400.76	+400.76
$\Delta\%$	+0.19%	+0.13%	+71.78%	+72.18%	+72.57%	+71.78%	+71.78%
Roll							
	DMF1	DMF2	BSQ/BIL/BIP	GeoTIFF	HDF5	CDF	NetCDF
-30°	+0.81	+0.46	+461.06	+462.46	+461.76	+458.96	+458.96
-24°	+0.7	+0.45	+352.76	+354.16	+355.46	+352.76	+352.76
-18°	+0.63	+0.44	+282.16	+283.56	+284.86	+282.16	+282.16
-12°	+0.58	+0.43	+236.96	+238.36	+239.66	+236.96	+236.96
-6°	+0.56	+0.43	+210.26	+211.66	+212.96	+210.26	+210.26
0°	+0.55	+0.43	+199.96	+201.36	+202.66	+199.96	+199.96
6°	+0.55	+0.43	+203.36	+204.76	+206.06	+203.36	+203.36
12°	+0.57	+0.43	+221.86	+223.26	+224.56	+221.86	+221.86
18°	+0.6	+0.44	+256.86	+258.26	+259.56	+256.86	+256.86
24°	+0.66	+0.44	+313.76	+315.06	+316.46	+313.76	+313.76
30°	+0.75	+0.46	+400.76	+402.16	+403.46	+400.76	+400.76
<i>dif</i>	+0.63	+0.44	+285.43	+286.82	+287.95	+285.24	+285.24
$\Delta\%$	+0.18%	+0.13%	+82.6%	+83%	+83.33%	+82.55%	+82.55%
Yaw							
	DMF1	DMF2	BSQ/BIL/BIP	GeoTIFF	HDF5	CDF	NetCDF
-30°	1.02	+0.47	+669.66	+671.56	+673.36	+669.66	+669.66
-24°	+0.96	+0.47	+607.36	+609.16	+610.86	+607.36	+607.36
-18°	+0.87	+0.46	+526.96	+528.76	+530.36	+526.96	+526.96
-12°	+0.78	+0.46	+435.36	+437.06	+438.66	+435.36	+435.36
-6°	+0.68	+0.45	+333.56	+335.06	+336.56	+333.56	+333.56
0°	+0.57	+0.43	+226.86	+228.36	+229.66	+226.96	+226.96
6°	+0.63	+0.44	+279.16	+280.56	+281.96	+279.16	+279.16
12°	+0.73	+0.45	+383.96	+385.56	+387.16	+383.96	+383.96
18°	+0.83	+0.46	+481.86	+483.56	+485.16	+481.86	+481.86
24°	+0.92	+0.47	+568.06	+569.86	+571.56	+568.06	+568.06
30°	+0.99	+0.47	+640.06	+641.96	+643.66	+640.06	+640.06
<i>dif</i>	+0.82	+0.46	+468.44	+470.13	+471.72	+468.45	+468.45
$\Delta\%$	+0.24%	+0.13%	+135.57%	+136.06%	+136.52%	+135.57%	+135.57%

Tabla 5.4: Diferencia de tamaños de espacio en disco (MB) de todos los formatos de almacenamiento respecto a la política DMF3 para cada movimiento de plataforma.

así como el porcentaje de incremento correspondiente ($\Delta\%$). Ambos parámetros han sido representados en la Tabla 5.6 y en la Tabla 5.7, respectivamente.

De igual manera que con el *dataset sintético*, se consideró el ángulo de vuelo para cada una de las imágenes del *dataset real*, y fueron geocorregidas mediante procedimientos convencionales considerando este parámetro. Posteriormente, fueron convertidas y almacenadas de acuerdo a los tipos de datos BSQ/BIL/BIP, GeoTIFF, HDF, CDF y NetCDF. El aspecto visual que presentan las imágenes geocorregidas respecto al ángulo de vuelo se muestran en la Figura 5.26. El ángulo de

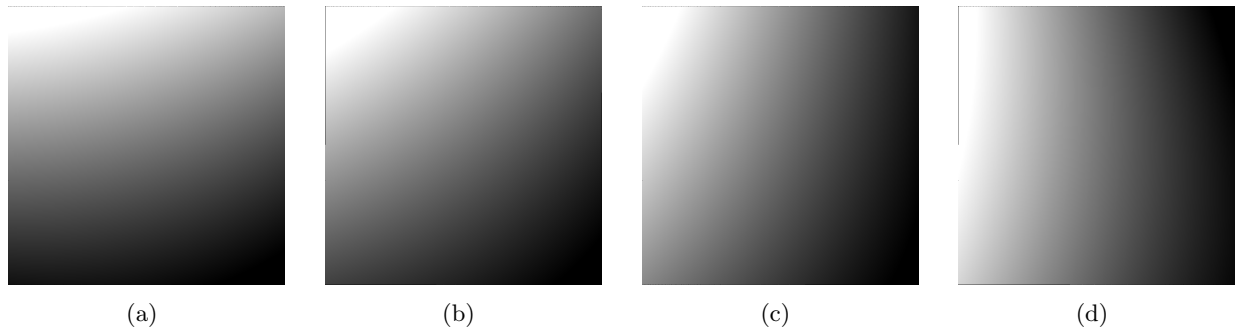


Figura 5.23: Imágenes geocorregidas a partir de los archivos L0 e IGM correspondientes al movimiento de plataforma yaw, considerando el ángulo de vuelo (Banda 0).
 (a) Yaw -30° . (b) Yaw -12° . (c) Yaw 12° . (d) Yaw 30° .

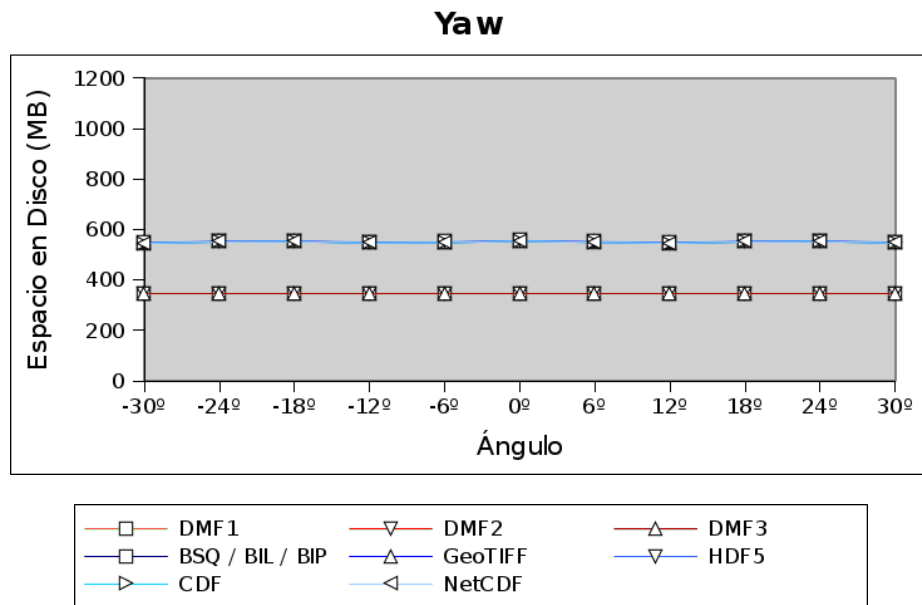


Figura 5.24: Tamaños de disco de los formatos de almacenamiento DMF y convencional para el movimiento de plataforma yaw, considerando ángulo de vuelo y tamaño de celda DMF automático.

	Yaw						
	DMF1	DMF2	BSQ/BIL/BIP	GeoTIFF	HDF5	CDF	NetCDF
-30°	+0.42	+0.43	+199.26	+200.66	+201.96	+199.26	+199.26
-24°	+0.42	+0.43	+204.76	+206.16	+207.46	+204.76	+204.76
-18°	+0.42	+0.43	+206.16	+207.46	+208.86	+206.16	+206.16
-12°	+0.42	+0.43	+200.66	+202.06	+203.36	+200.66	+200.66
-6°	+0.42	+0.43	+201.96	+203.36	+204.76	+201.96	+201.96
0°	+0.42	+0.43	+207.46	+208.86	+210.26	+207.46	+207.46
6°	+0.42	+0.43	+203.36	+204.76	+206.06	+203.36	+203.36
12°	+0.42	+0.43	+199.26	+200.66	+201.96	+199.26	+199.26
18°	+0.42	+0.43	+204.76	+206.16	+207.46	+204.76	+204.76
24°	+0.42	+0.43	+206.16	+207.46	+208.86	+206.16	+206.16
30°	+0.42	+0.43	+200.66	+202.06	+203.36	+200.66	+200.66
<i>dif</i>	+0.42	+0.43	+203.13	+204.51	+205.85	+203.13	+203.13
$\Delta \%$	+0.12 %	+0.12 %	+58.79 %	+59.19 %	+59.57 %	+58.79 %	+58.79 %

Tabla 5.5: Diferencia de tamaños de espacio en disco (MB) de todos los formatos de almacenamiento respecto a la política DMF3 para el movimiento de plataforma yaw, considerando ángulo de vuelo y tamaño de celda DMF automático.

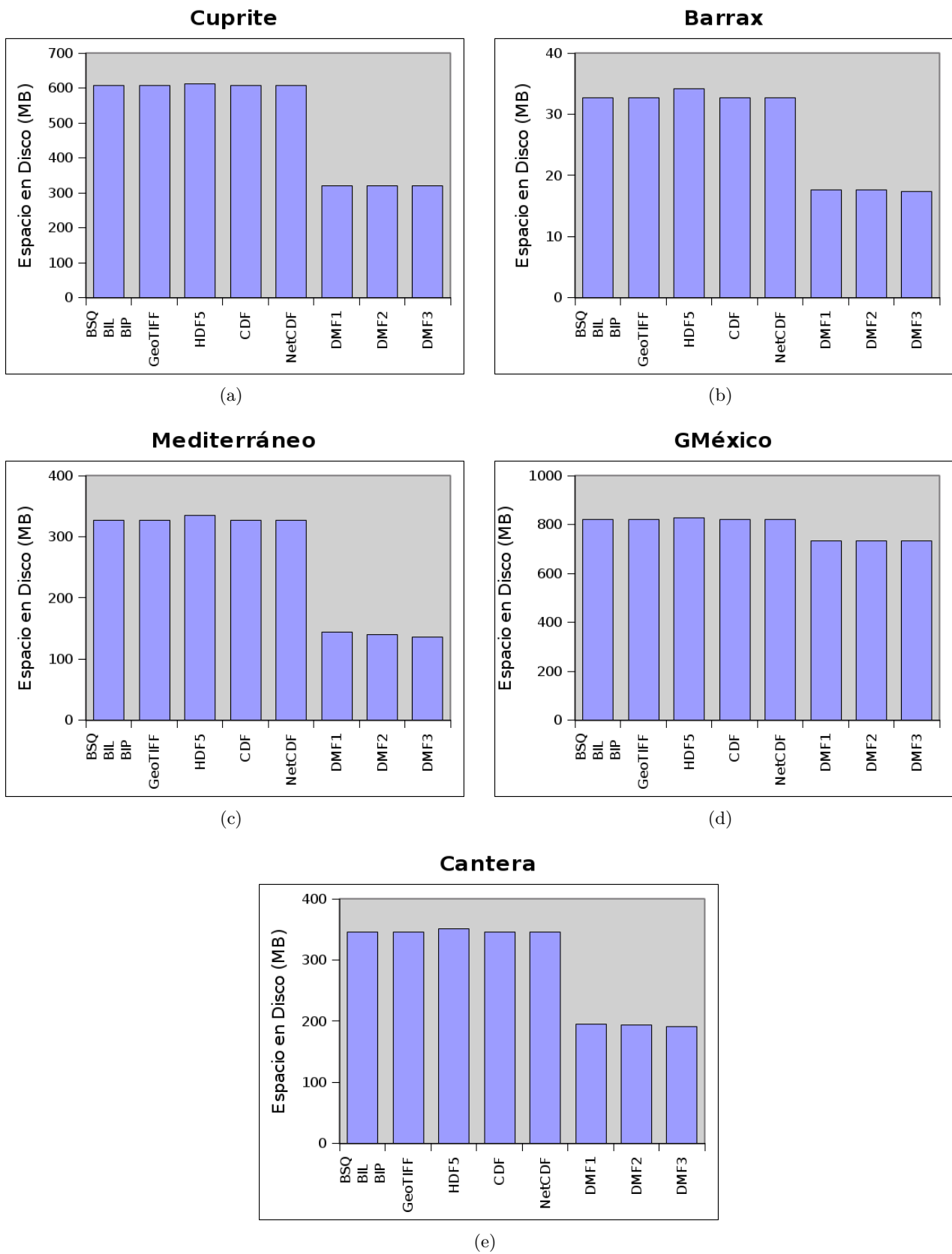


Figura 5.25: Tamaños de disco de los formatos de almacenamiento DMF y convencional para el dataset de imágenes real.

(a) Escena Cuprite. (b) Escena Barrax. (c) Escena Mediterráneo. (d) Escena GMéxico. (e) Escena Cantera.

	DMF1	DMF2	BSQ/BIL/BIP	GeoTIFF	HDF5	CDF	NetCDF
<i>Cuprite</i>	+1.36	+0.86	+287.98	+288.08	+292.58	+288.08	+287.98
<i>Barrax</i>	+0.26	+0.16	+15.28	+15.28	+16.78	+15.28	+15.28
<i>Mediterráneo</i>	+7.42	+3.41	+190.19	+190.19	+197.89	+190.19	+190.19
<i>GMéxico</i>	+1.83	+1.36	+88.89	+88.89	+292.58	+95.99	+88.89
<i>Cantera</i>	+2.75	+1.61	+154.31	+154.31	+159.21	+154.31	+154.31

Tabla 5.6: Diferencia de tamaños de espacio en disco (MB) de todos los formatos de almacenamiento respecto a la política DMF3 para el dataset real (*dif*).

	DMF1	DMF2	BSQ/BIL/BIP	GeoTIFF	HDF5	CDF	NetCDF
<i>Cuprite</i>	+0.42%	+0.27%	+89.99%	+90.02%	+91.43%	+90.02%	+89.99%
<i>Barrax</i>	+1.49%	+0.92%	+87.72%	+87.72%	+96.32%	+87.72%	+87.72%
<i>Mediterráneo</i>	+5.44%	+2.5%	+139.53%	+139.53%	+145.18%	+139.53%	+139.53%
<i>GMéxico</i>	+0.25%	+0.18%	+12.15%	+12.15%	+13.12%	+12.15%	+12.15%
<i>Cantera</i>	+1.43%	+0.84%	+80.5%	+80.5%	+83.06%	+80.5%	+80.5%

Tabla 5.7: Porcentajes de incremento de tamaños de espacio en disco de todos los formatos de almacenamiento respecto a la política DMF3 para el dataset real ($\Delta\%$).

vuelo también fue considerado a la hora de construir y almacenar las matrices DMF correspondientes siguiendo las diferentes políticas presentadas previamente, así como un tamaño de celda DMF calculado de manera automática, permitiendo una maximización en la tendencia de 1 registro DMR por celda DMF, y una minimización del porcentaje de celdas DMF a *Null*.

Los resultados correspondientes a los tamaños de almacenamiento en disco obtenidos para todas las imágenes del *dataset real* se muestran en la Figura 5.27.

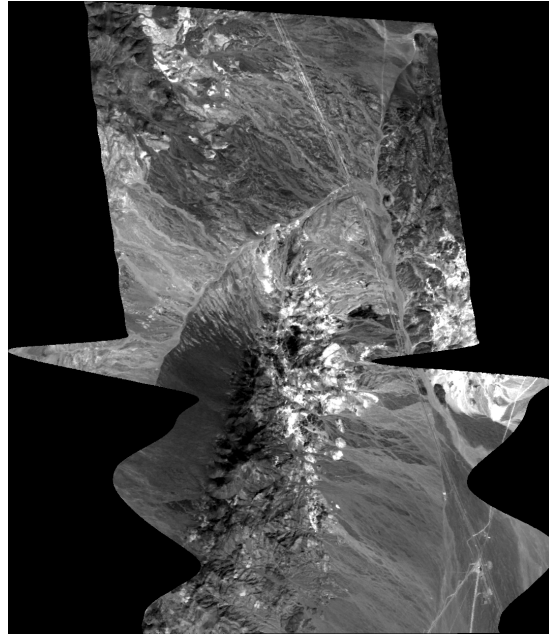
Para una detallada visualización de los resultados obtenidos, las diferencias de tamaño en disco respecto al formato de almacenamiento referencia *DMF3 (dif)*, así como el porcentaje de incremento correspondiente ($\Delta\%$), han sido también representados en la Tabla 5.8 y en la Tabla 5.9, respectivamente.

	DMF1	DMF2	BSQ/BIL/BIP	GeoTIFF	HDF5	CDF	NetCDF
<i>Cuprite</i>	+1.23	+0.83	+254.78	+254.78	+258.98	+254.78	+254.78
<i>Barrax</i>	+0.14	+0.15	+8.18	+8.18	+10.08	+8.18	+8.18
<i>Mediterráneo</i>	+5.17	+3.07	+141.29	+141.29	+152.29	+141.29	+141.29
<i>GMéxico</i>	+1.68	+1.36	-21.21	-21.21	-14.21	-21.21	-21.21
<i>Cantera</i>	+2.11	+1.64	+32.01	+32.01	+38.01	+32.01	+32.01

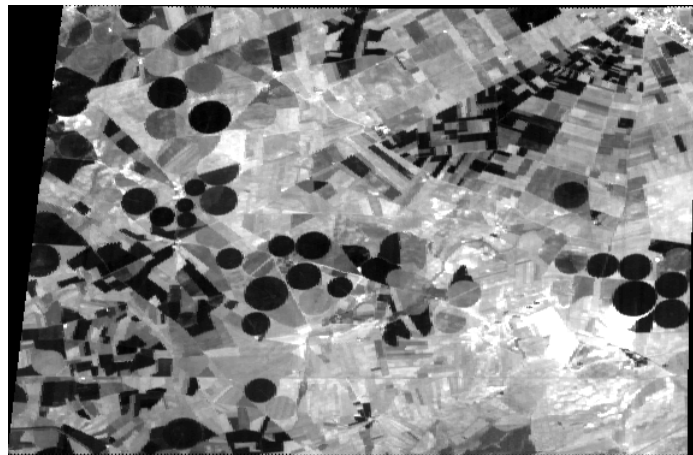
Tabla 5.8: Diferencia de tamaños de espacio en disco (MB) de todas las imágenes correspondientes al dataset real respecto a la política DMF3 (*dif*), considerando el ángulo de vuelo y un tamaño de celda DMF automático.

	DMF1	DMF2	BSQ/BIL/BIP	GeoTIFF	HDF5	CDF	NetCDF
<i>Cuprite</i>	+0.38%	+0.26%	+79.61%	+79.61%	+80.93%	+79.61%	+79.61%
<i>Barrax</i>	+0.8%	+0.86%	+46.96%	+46.96%	+57.86%	+46.96%	+46.96%
<i>Mediterráneo</i>	+3.79%	+2.25%	+103.65%	+103.65%	+111.72%	+103.65%	+103.65%
<i>GMéxico</i>	+0.23%	+0.19%	-2.9%	-2.9%	-1.94%	-2.9%	-2.9%
<i>Cantera</i>	+1.1%	+0.86%	+16.7%	+16.7%	+19.83%	+16.7%	+16.7%

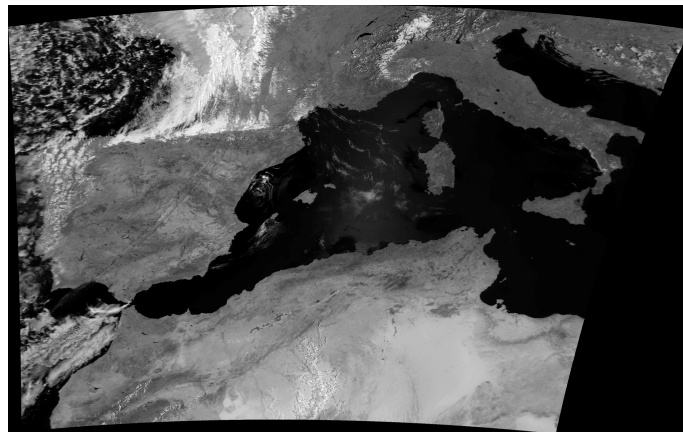
Tabla 5.9: Porcentajes de incremento de tamaños de espacio en disco de todas las imágenes correspondientes al dataset real respecto a la política DMF3 ($\Delta\%$), considerando el ángulo de vuelo y un tamaño de celda DMF automático.



(a)



(b)



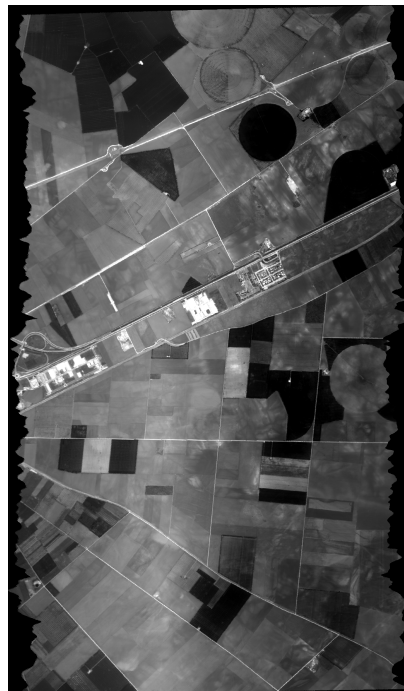
(c)

Figura 5.26: Visualización de las imágenes geocorregidas considerando el ángulo de vuelo para el dataset de imágenes real.

(a) Escena Cuprite [Banda 10]. (h) Escena Barrax [Banda 0]. (c) Escena Mediterráneo [Banda 4]. (g) Escena GMéxico [Banda 10]. (b) Escena Cantera [Banda 5].



(g)



(h)

Figura 5.26: (Cont.)

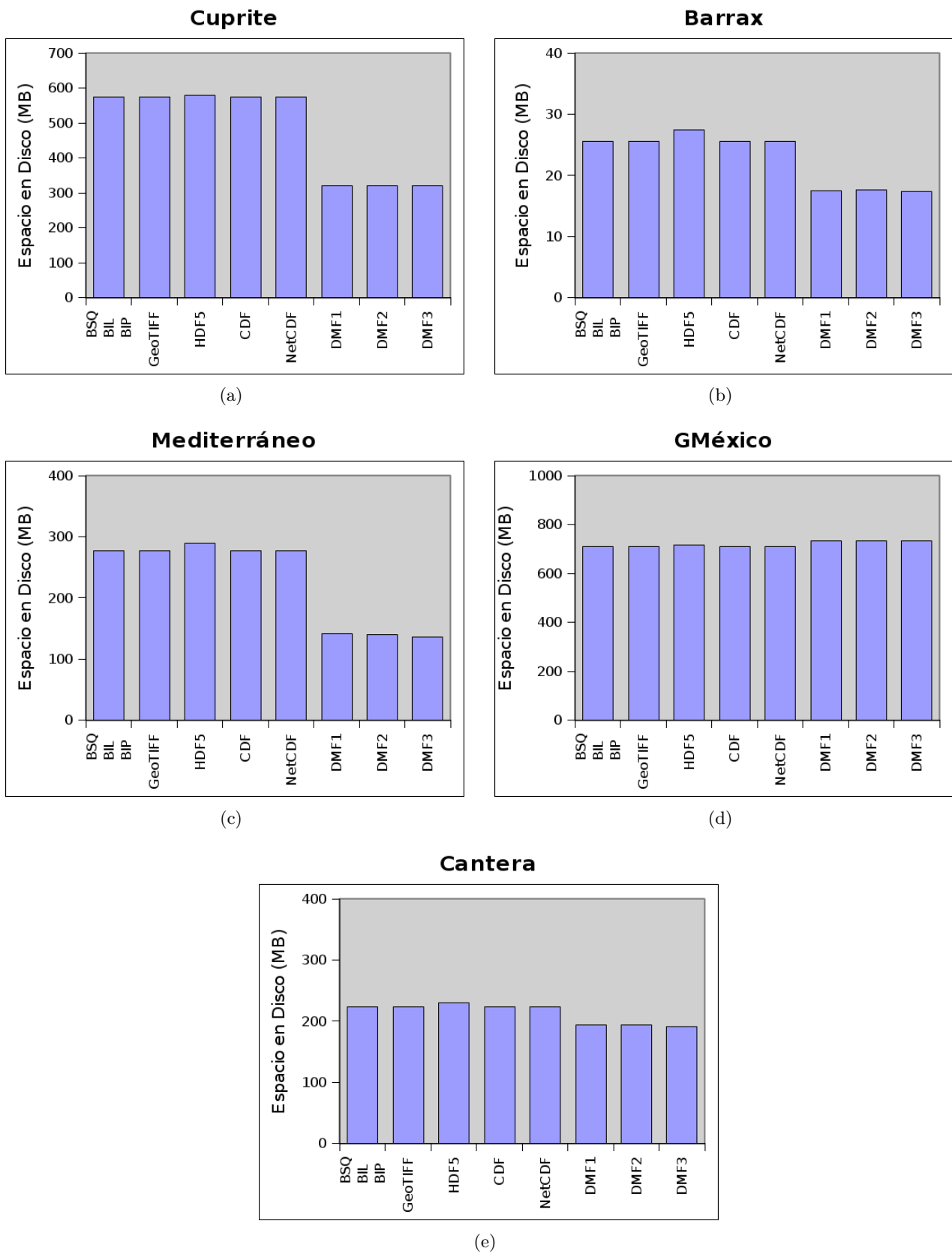


Figura 5.27: Tamaños de disco de los formatos de almacenamiento DMF y convencional para cada imagen del dataset real, considerando el ángulo de vuelo y un tamaño celda DMF automático.
 (a) Escena Cuprite. (b) Escena Barrax. (c) Escena Mediterráneo. (d) Escena GMéxico. (e) Escena Cantera.

5.4.4. Análisis de los resultados del experimento.

Para el análisis de los tamaños en discos de los diferentes formatos de imágenes, incluyendo el formato DMF, se realizaron los siguientes experimentos:

1. Estudio del tamaño en disco para el *dataset sintético*:
 - a) Análisis del consumo de espacio en disco para los diferentes formatos (Figura 5.22 y Tabla 5.4).
 - b) Análisis del consumo de espacio en disco para los diferentes formatos, considerando ángulo de vuelo y tamaño de celda DMF calculado de manera automática (Figura 5.24 y Tabla 5.5).
2. Estudio del tamaño en disco para el *dataset real*:
 - a) Análisis del consumo de espacio en disco para los diferentes formatos (Figura 5.25, Tabla 5.6 y Tabla 5.7).
 - b) Análisis del consumo de espacio en disco para los diferentes formatos, considerando ángulo de vuelo y tamaño de celda DMF calculado de manera automática (Figura 5.27, Tabla 5.8 y Tabla 5.9).

Analizando detalladamente los resultados obtenidos respecto al tamaño de espacio en disco requerido por cada imagen en cada uno de los formatos es posible determinar varias conclusiones.

En un primer lugar, es posible comprobar que, de manera general, en todos los casos el formato DMF presenta siempre un consumo de tamaño en disco menor que las versiones geocorregidas análogas en los diferentes formatos convencionales estudiados. Este incremento de consumo de tamaño, para el caso de las imágenes geocorregidas, es producido por el hecho de almacenar los píxeles del exterior de la escena como espectros de n bandas con valores a 0. Es por ello por lo que, cuanto más borde muestre la imagen, más consumo de espacio en disco requiere. Principalmente este es el motivo por el cual las imágenes geocorregidas se ofrecen siempre orientadas en dirección N/S, corrigiendo el ángulo de vuelo. Este hecho queda patente si se comparan los resultados de las gráficas correspondientes a cualquier imagen geocorregida sin tener en cuenta la corrección del ángulo de vuelo, y las correspondientes a si se considera dicho ángulo. En este último caso el tamaño en disco siempre es notablemente menor, implicando en casos extremos, como por ejemplo el del movimiento de plataforma *yaw* (Figura 5.22(c) y Figura 5.24), una reducción del tamaño hasta de un 300%. En el *dataset real* es también posible observar este fenómeno, y comprobar cómo el tamaño en disco de las imágenes geocorregidas disminuye considerablemente al considerar el ángulo de vuelo en el proceso de geocorrección, acercándose más al obtenido por el formato DMF (Figura 5.25 y Figura 5.27).

Si se analiza de manera independiente los resultados obtenidos para el *dataset sintético*, se puede comprobar que las imágenes geocorregidas siempre suponen un mayor consumo de espacio en disco, pero en el caso del tipo de movimiento *yaw* (Figura 5.22(c)), la diferencia de tamaños es notable. Para este tipo de movimiento, el tamaño es un parámetro que depende directamente del ángulo de vuelo, pues la imagen presenta un mínimo de bordes cuando ha sido adquirida en dirección N/S (y por tanto un mínimo de consumo de espacio en disco), el cual se hace mayor a medida que el ángulo de adquisición aumenta (incrementando también el consumo de espacio en disco) al no estar la escena completamente alineada con la matriz que representa la imagen. Si al construir la matriz DMF se considera el ángulo de vuelo, entonces es posible ver que este efecto desaparece, quedando un tamaño de disco constante independiente de dicho ángulo (Figura 5.24). El resto de movimientos de plataforma no muestra una diferencia de tamaños respecto al formato DMF tan agresiva, aunque sí considerable, mostrando también sus peculiaridades. Para el caso del tipo de movimiento *pitch* (Figura 5.22(a)), el tamaño en disco para las imágenes geocorregidas aumentan a medida que el ángulo se incrementa. Esto

es debido a que a mayor ángulo, más altura adquiere el avión por unidad de tiempo, implicando una mayor distancia incremental (a medida que el avión asciende) entre líneas de adquisición. El espaciado entre estas líneas, al considerar un tamaño de píxel fijo para toda la imagen, ha de rellenarse con espectros a 0, aumentando de esta manera el consumo de espacio en disco. Para el caso del tipo de movimiento *roll* (Figura 5.22(b)), la gráfica presenta un comportamiento similar a la del tipo de movimiento *yaw*, pero más suavizada, mostrando un mayor consumo de espacio a medida que el ángulo es mayor. Esto es producido por el incremento gradual de las distancias entre mediciones (*sampleado*) a medida que se alejan del Nadir, la cual se potencia al estar el avión también inclinado.

Respecto a los resultados obtenidos para las imágenes correspondientes al *dataset real*, el formato DMF presenta siempre un consumo de espacio considerablemente menor, aunque la diferencia entre los formatos geocorregidos y DMF se decrementa cuando se considera el ángulo de rotación en el proceso de geocorrección. De hecho, para el caso de la escena *GMéxico*, el tamaño en disco presentados por las versiones geocorregidas pueden llegar a ser menor que el de las versiones DMF (Figura 5.27(d)), incluyendo la política de almacenamiento *DMF3*, que posee el mismo tamaño que la totalidad de datos recogida por el sensor y procedentes del sistema de posicionamiento. Esto implica que en verdad se está almacenando menos información de la que correspondería, siendo el principal causante la selección de un tamaño de píxel demasiado grande, lo que provoca la eliminación de muchas mediciones al corresponder sólo una por píxel. La diferencia de tamaños respecto a la política *DMF3* (*dif*), así como su porcentaje de incremento ($\Delta\%$) han sido marcados de color rojo en la Tabla 5.7 y en la Tabla 5.9.

La diferencia entre las tres políticas de almacenamiento DMF *DMF1*, *DMF2* y *DMF3* no es del todo notable, sobrepasando ésta el MB en sólo en muy casos puntuales. Comparado con el tamaño total del fichero resultante, el tamaño dedicado a máscaras binarias y cabeceras empleadas en las políticas *DMF1* y *DMF2* es despreciable. Como cabía esperar, es la política *DMF1* la que conlleva un mayor consumo de espacio en disco, al almacenar una cabecera por celda DMF; mientras que es la política *DMF3* la que conlleva un menor consumo, al almacenar únicamente la información recogida por el sensor o proporcionado por el sistema de posicionamiento, sin datos adicionales. Por ello, se hará referencia a esta política siempre que se referencia el formato de almacenamiento DMF en el resto de experimentos.

La diferencia entre los propios formatos de almacenamiento convencionales BSQ/BIL/BIP, GeoTIFF, HDF, CDF y NetCDF también es mínima, siendo ésta, generalmente, de varios MB. Esta diferencia de tamaños queda determinada por el nivel de información adicional (cabeceras, máscaras, etc.) que cada uno de los formatos emplea, siendo los que presentan los mayores niveles el formato HDF5, seguido por el formato GeoTIFF. El consumo de espacio en disco empleado por los formatos BSQ/BIL/BIP, CDF y NetCDF, presentan los mismos resultados en todas las tablas, sin embargo, la diferencia de consumo entre ellas es aún menor que para los formatos anteriores, del orden de KB, y correspondientes a la diferencia de campos utilizados en sus cabeceras principales.

5.5. Tiempos de ejecución.

El formato DMF es una estructura de datos basada en una matriz que es capaz de almacenar en cada celda ninguna, una o varias mediciones adquiridas por el sensor. De esta manera, es posible encontrar zonas en dicha matriz con diferentes densidades de datos.

Las metodologías de procesamiento de imágenes convencionales deben ser adaptadas para que puedan considerar el hecho de que el número de mediciones existentes en las celdas de la matriz sea variable.

Esta sección se centra en la manera en la cual dichas metodologías han sido adaptadas y cómo el hecho de considerar la posibilidad de almacenar diferentes mediciones por celda DMF afecta al tiempo de procesamiento de las imágenes respecto a sus análogas geocorregidas.

5.5.1. *Objetivos del experimento.*

Los procedimientos y metodologías de procesamiento de imágenes hiperespectrales convencionales necesitan ser adaptadas al formato DMF, una estructura igualmente matricial pero cuyas celdas pueden almacenar ninguna, una o varias mediciones.

De este modo, los principales objetivos que se persiguen en este experimento son los siguientes:

- Estudio del rendimiento de metodologías de procesamiento de imágenes hiperespectrales basados y no basados en vecindad a la hora de procesar imágenes en formato DMF convencional.
- Análisis de diseños más avanzados del formato DMF orientados a la minimización de los tiempos de procesamiento, como la indexación de los datos dentro de las celdas DMF mediante árboles Kd-tree, o la lista secuencial de punteros a todos los registros DMR almacenados.
- Comparación de los tiempos de ejecución al considerar el ángulo de vuelo y un tamaño de celda DMF automático.

5.5.2. *Descripción del experimento.*

Este experimento fue realizado con la intención de comprobar el rendimiento de la matriz DMF al aplicar técnicas de procesamiento de imágenes convencionales, pero adaptadas para aceptar celdas de listas de mediciones dinámicas. Para ello, se seleccionaron dos metodologías muy utilizadas en el campo de la teledetección: una que no necesita análisis de vecindad, el umbralizado; y otra que sí requiere análisis de vecindad, la dilatación.

La técnica del umbralizado consiste en la eliminación de los píxeles cuyo valor no supera un determinado límite o *umbral* [Sez04, Ots79]. En una imagen convencional, han de recorrerse todos los píxeles y comprobar cuáles son los que superan el límite y cuáles no. Para el caso del formato DMF, habrá que recorrer todas las celdas DMF, y en cada celda, recorrer todos los registros DMR para comprobar cuáles superan el límite y cuáles no. Sin embargo, los píxeles y las mediciones de las imágenes hiperespectrales no almacenan un único valor, sino una secuencia de valores que representa el espectro. Para convertir esta secuencia de valores a un único valor representativo, se optó por aplicar la Norma Euclídea [Noy07], la cual se especifica como:

$$\|x\| = \sqrt{x_1^2 + x_2^2 + \dots + x_n^2} \quad (5.3)$$

y en donde $[x_1, x_2, \dots, x_n]$ representa el espectro.

Además, en el proceso de umbralizado no se eliminarán ni píxeles ni mediciones, sino que a todos los valores del espectro se les asignará el valor de 0. De esta manera, los espectros de las mediciones y píxeles cuya Norma Euclídea no supere un determinado valor umbral, serán puestos a 0.

La técnica de la dilatación consiste en la asignación del mayor valor de los píxeles comprendidos en una ventana de tamaño $n \times n$ (denominada *Kernel*), a un píxel referencia [Ser84, Har87, Shi09]. Igual que en el procedimiento anterior, al considerar espectros enteros en cada píxel o medición, se considerará como *espectro mayor* a aquél cuya Norma Euclídea es mayor. En una imagen convencional, han de recorrerse todos los píxeles, y para cada uno de ellos, asignarles el *mayor espectro* de todos los píxeles comprendidos en el *Kernel*, siendo el centro de la ventana el píxel que se está analizando. Para el caso del formato DMF, no se considerará un tamaño de ventana, sino un radio r . De esta manera, será necesario recorrer todas las celdas DMF, y en cada celda, recorrer todos los registros DMR. A cada registro DMR, se le asignará el *mayor espectro* de todas las mediciones comprendidas dentro

de ese radio r . Tras aplicar una dilatación, la imagen resultante parece haber *dilatado* las zonas más luminosas.

La manera de aplicar ambas metodologías, para el caso del formato DMF, está detallado en la Sección 4.1.1.4. El estudio se realizó con todas las imágenes tanto del *dataset sintético* como del *dataset real*, considerando y sin considerar el ángulo de vuelo y el cálculo automático del tamaño de celda DMF. Para una mejor comparación de la eficiencia, cada escena estará representada en las versiones geocorregidas, DMF y DMF optimizada, siendo este último caso una variación del formato DMF en el que los registros DMR almacenados en una misma celda DMF están indexados mediante un árbol Kd-tree, y en el que está presente una lista dinámica que permite el acceso directo a todos los registros DMR de la matriz DMF (Figura 4.5(b) y Figura 4.12). La indexación de cada celda DMF mediante un árbol Kd-tree permitirá un más rápido acceso a los datos de las celdas DMF ocupadas (no a *Null*) por lo que permitirá optimizar la operación de dilatación. Por el otro lado, la lista de accesos directos a los registros DMR permitirá optimizar la operación de umbralizado, al no tener que perder tiempo en analizar si una celda está a *Null*.

Aparte de las puntualizaciones anteriores, es necesario también tener en cuenta las siguientes consideraciones:

- Debido al consumo de tiempo que implica utilizar raíces cuadradas, para el cálculo de la Norma Euclídea esta operación fue eliminada. De esta manera, las operaciones de umbralizado y dilatación no se ven alteradas (la eliminación de esta operación supone resultados de Norma Euclídea mayores, pero equivalentes y proporcionales) y los tiempos de ejecución son más fáciles de analizar, al no estar camuflados por los correspondientes a la raíz cuadrada. Por lo tanto, se consideró la norma euclídea como:

$$\|x\|^2 = x_1^2 + x_2^2 + \dots + x_n^2 \quad (5.4)$$

en donde $[x_1, x_2, \dots, x_n]$ representa el espectro.

- Para el caso de la técnica de umbralizado, el valor umbral utilizado varía para cada escena, consistente en un valor medio aproximado de Norma Euclídea $\|x\|^2$ (sin considerar raíces cuadradas). Esto producirá que los valores de espectro de más o menos la mitad de los datos almacenados sean puestos a 0. Se aplicará el mismo valor de umbral para los formatos geocorregidos, DMF y DMF optimizado de una misma escena, incluyendo las versiones en las que se considera el ángulo de vuelo y el cálculo automático del tamaño de celda DMF. La Tabla 5.10 muestra los valores de umbral aplicados para cada una de las escenas, tanto del *dataset sintético* como del *dataset real*.

Dataset sintético	
<i>Pitch</i>	70.139
<i>Roll</i>	70.841
<i>Yaw</i>	70.701
Dataset real	
<i>Cuprite</i>	887123759.557
<i>Barrax</i>	11452672676.988
<i>Mediterráneo</i>	61703.504
<i>GMéxico</i>	299807188.795
<i>Cantera</i>	2190265606.692

Tabla 5.10: Valores de umbral $\|x\|^2$ seleccionados para cada uno de los datasets de imágenes.

- Para el caso de la técnica de dilatación, el radio de vecindad r seleccionado para cada imagen es el mismo que el tamaño de píxel proporcionado por el proveedor (Tabla 5.2). De esta forma se garantiza un tamaño de ventana de 3×3 celdas DMF o píxeles. Se aplicará el mismo valor

de umbral para los formatos geocorregidos, DMF y DMF optimizado de una misma escena, incluyendo las versiones en las que se considera el ángulo de vuelo y el cálculo automático del tamaño de celda DMF.

5.5.3. Resultados del experimento.

Las operaciones de umbralizado y dilatación fueron adaptadas al formato DMF tal y como se detalló en la sección anterior (Sección 5.5.2). En un primer lugar, dichas operaciones fueron aplicadas a las imágenes correspondientes al *dataset sintético*, utilizando el tamaño de píxel / celda DMF proporcionado por el proveedor y en las versiones geocorregida, DMF y DMF optimizada. Este último caso, consiste en una versión modificada del formato DMF que incluye la indexación de los datos almacenados en las celdas DMF mediante árboles Kd-tree, así como una lista dinámica de accesos directos a los registros DMR almacenados en la matriz DMF. En un segundo lugar, el mismo experimento fue realizado para el *dataset real*. Las gráficas que se presentan a continuación siempre muestran la media de tiempos de ejecución al aplicar 100 iteraciones.

Respecto al *dataset sintético*, el aspecto visual resultante, para las versiones geocorregidas y DMF, se muestran en las Figuras 5.28, 5.29, 5.30 y 5.31. Los tiempos de ejecución también fueron medidos, y los resultados para ambas técnicas aparecen representados en la Figura 5.32 y en la Figura 5.33, respectivamente. En estas gráficas, la serie azul representa los tiempos de ejecución de las escenas en su versión geocorregida (*Geo*); la roja oscura los correspondientes a la matriz DMF convencional (*DMF*); mientras que la roja clara punteada representa la DMF optimizada (*DMF Opt*).

El mismo procedimiento fue aplicado al mismo dataset pero para el tipo de movimiento de plataforma *yaw*, considerando el ángulo de vuelo, y además un tamaño de celda DMF automático para las versiones DMF y DMF optimizada. Los tiempos de ejecución correspondientes aparecen representados en la Figura 5.34 y en la Figura 5.35, respectivamente. Igual que en el caso anterior, en estas gráficas la serie azul representa los tiempos de ejecución de las escenas en su versión geocorregida (*Geo α*); la roja oscura los correspondientes a la matriz DMF convencional (*DMF α*); mientras que la roja clara punteada representa la DMF optimizada (*DMF Opt α*).

Tras aplicar las técnicas de umbralizado y dilatación al *dataset sintético*, éstas fueron aplicadas del mismo modo a todas las escenas del *dataset real* en sus versiones geocorregida (*Geo*), DMF (*DMF*) y DMF optimizada (*DMF Opt*). El aspecto visual resultante, para las versiones geocorregidas y DMF, se muestran en las Figuras 5.36, 5.37, 5.38 y 5.39. Los tiempos de ejecución fueron medidos, y los resultados para ambas técnicas aparecen representados en la Figura 5.40 y en la Figura 5.41, respectivamente.

Finalmente, las técnicas de umbralizado y dilatación fueron también aplicadas al *dataset sintético*, pero considerando el ángulo de vuelo, y además un tamaño de celda DMF automático para las versiones DMF y DMF optimizada. Los tiempos de ejecución correspondientes aparecen representados en la Figura 5.42 y en la Figura 5.43, respectivamente.

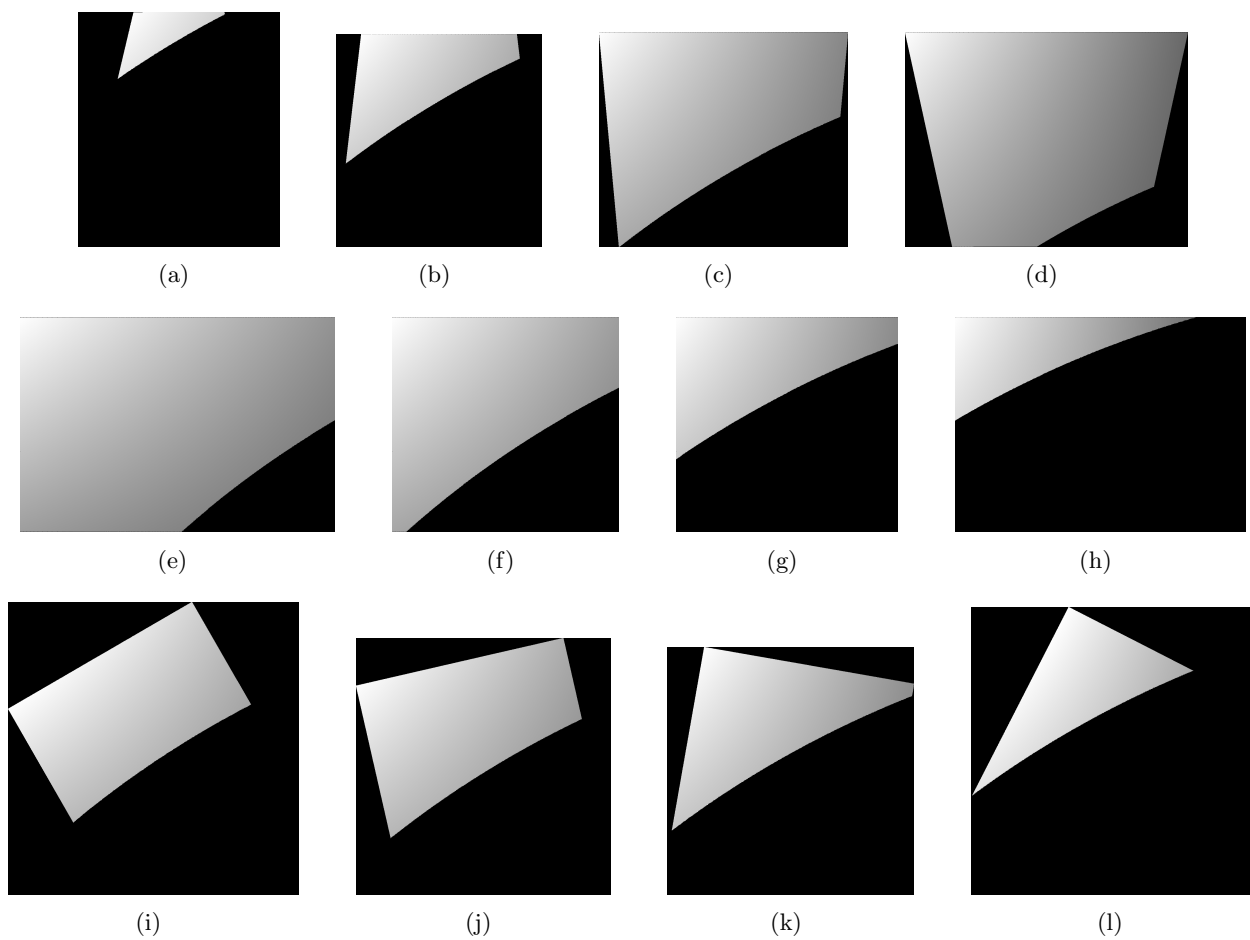


Figura 5.28: Visualización de las imágenes del dataset sintético (formato geocorregido) tras aplicar la operación de umbralizado (Banda 0).

(a) Pitch -30° . (b) Pitch -12° . (c) Pitch 12° . (d) Pitch 30° . (e) Roll -30° . (f) Roll -12° . (g) Roll 12° . (h) Roll 30° . (i) Yaw -30° . (j) Yaw -12° . (k) Yaw 12° . (l) Yaw 30° .

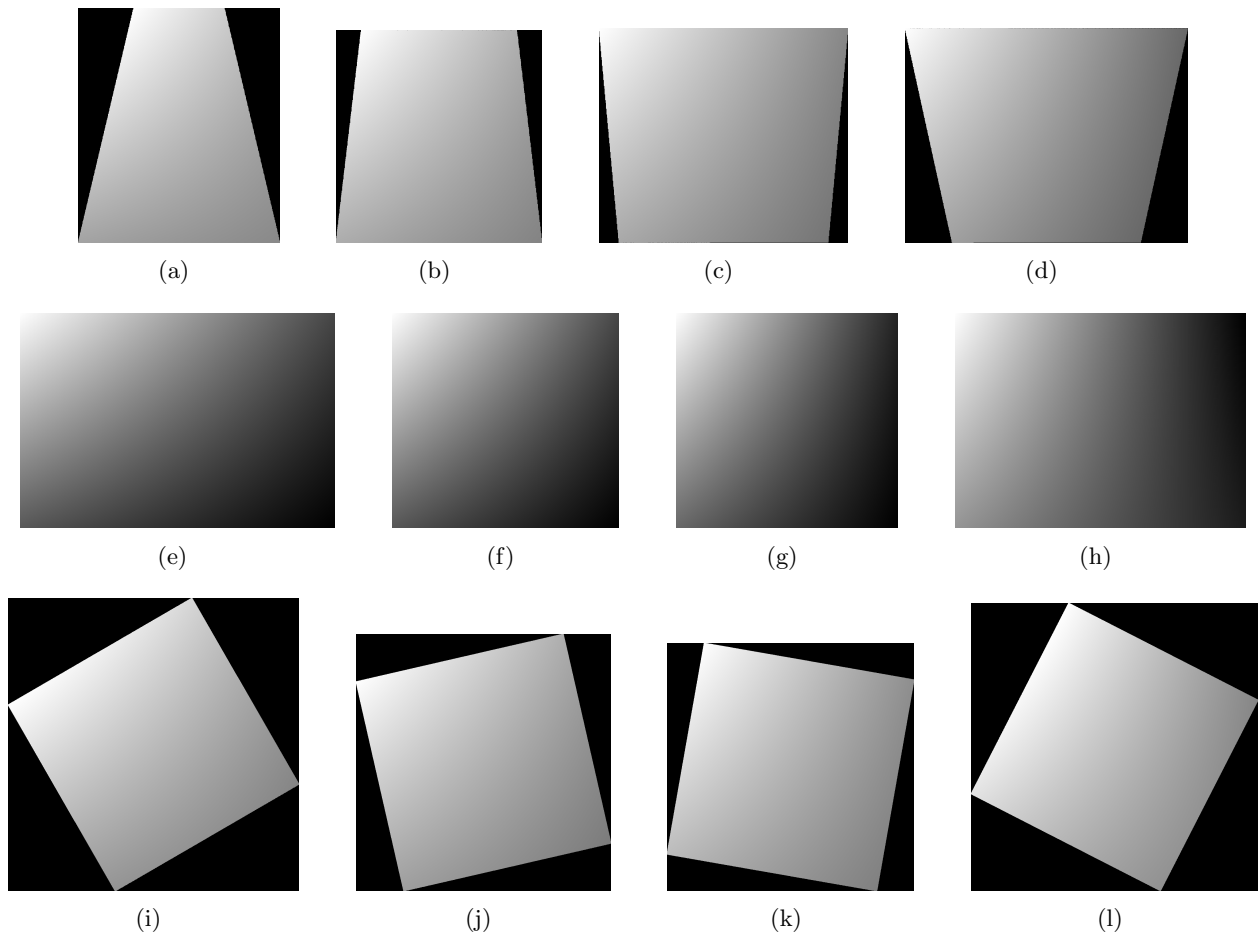


Figura 5.29: Visualización de las imágenes del dataset sintético (formato geocorregido) tras aplicar la operación de dilatación (Banda 0).

(a) Pitch -30° . (b) Pitch -12° . (c) Pitch 12° . (d) Pitch 30° . (e) Roll -30° . (f) Roll -12° . (g) Roll 12° . (h) Roll 30° . (i) Yaw -30° . (j) Yaw -12° . (k) Yaw 12° . (l) Yaw 30° .

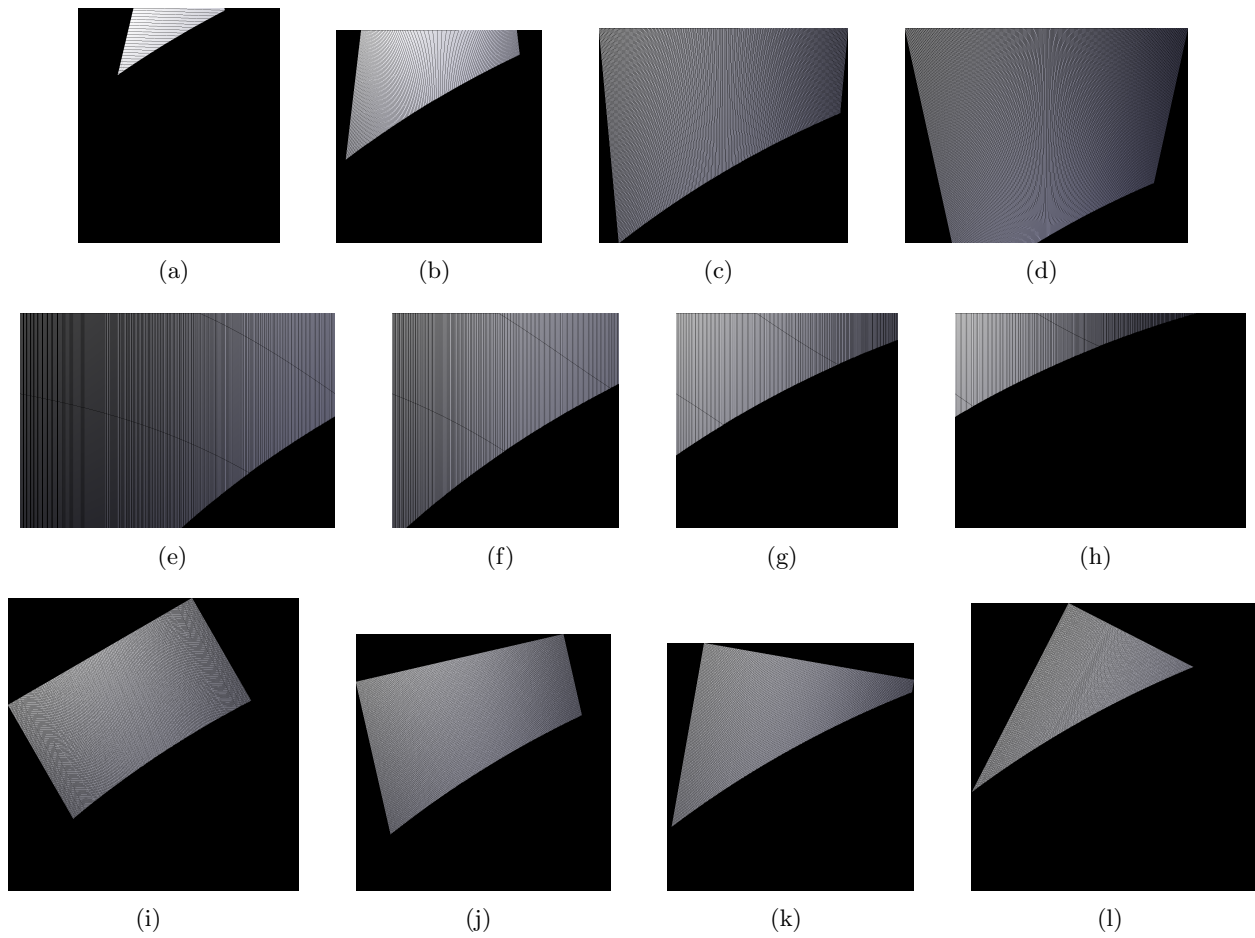


Figura 5.30: Visualización de las imágenes del dataset sintético (formato DMF) tras aplicar la operación de umbralizado (Banda 0).
 (a) Pitch -30° . (b) Pitch -12° . (c) Pitch 12° . (d) Pitch 30° . (e) Roll -30° . (f) Roll -12° . (g) Roll 12° . (h) Roll 30° . (i) Yaw -30° . (j) Yaw -12° . (k) Yaw 12° . (l) Yaw 30° .

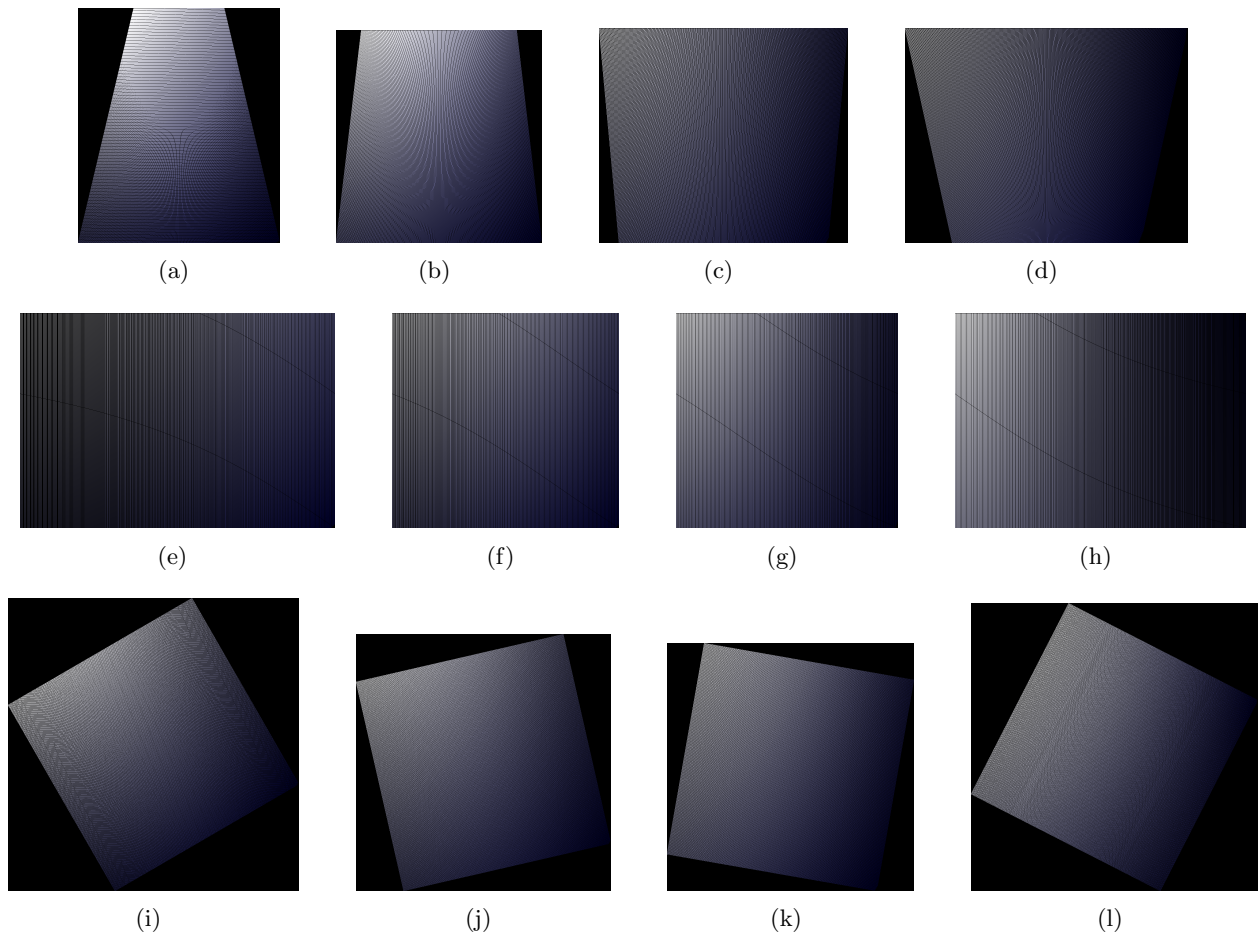
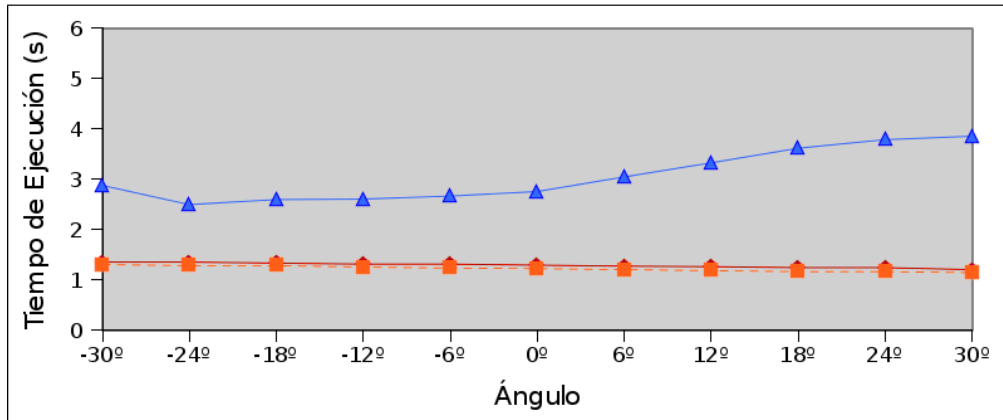


Figura 5.31: Visualización de las imágenes del dataset sintético (formato DMF) tras aplicar la operación de dilatación (Banda 0).

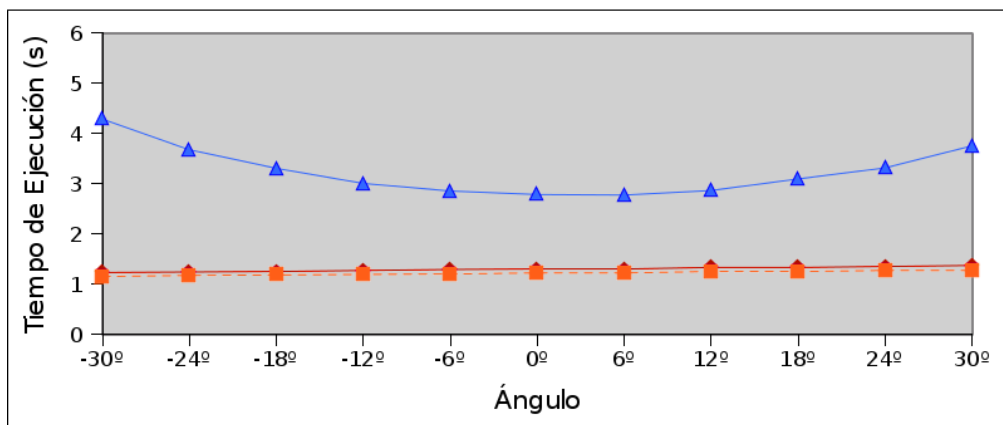
(a) Pitch -30° . (b) Pitch -12° . (c) Pitch 12° . (d) Pitch 30° . (e) Roll -30° . (f) Roll -12° . (g) Roll 12° . (h) Roll 30° . (i) Yaw -30° . (j) Yaw -12° . (k) Yaw 12° . (l) Yaw 30° .

Umbralizado (Pitch)



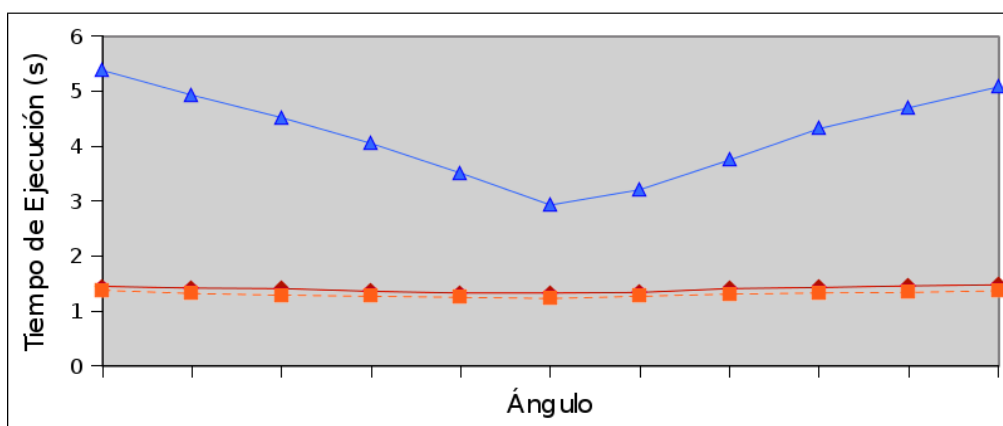
(a)

Umbralizado (Roll)



(b)

Umbralizado (Yaw)



(c)

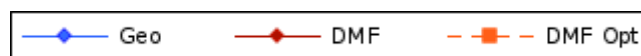


Figura 5.32: Tiempos de ejecución de la operación de umbralizado para cada uno de los movimientos de plataforma del dataset sintético en sus versiones geocorregida, DMF y DMF optimizada.

(a) Pitch. (b) Roll. (c) Yaw.

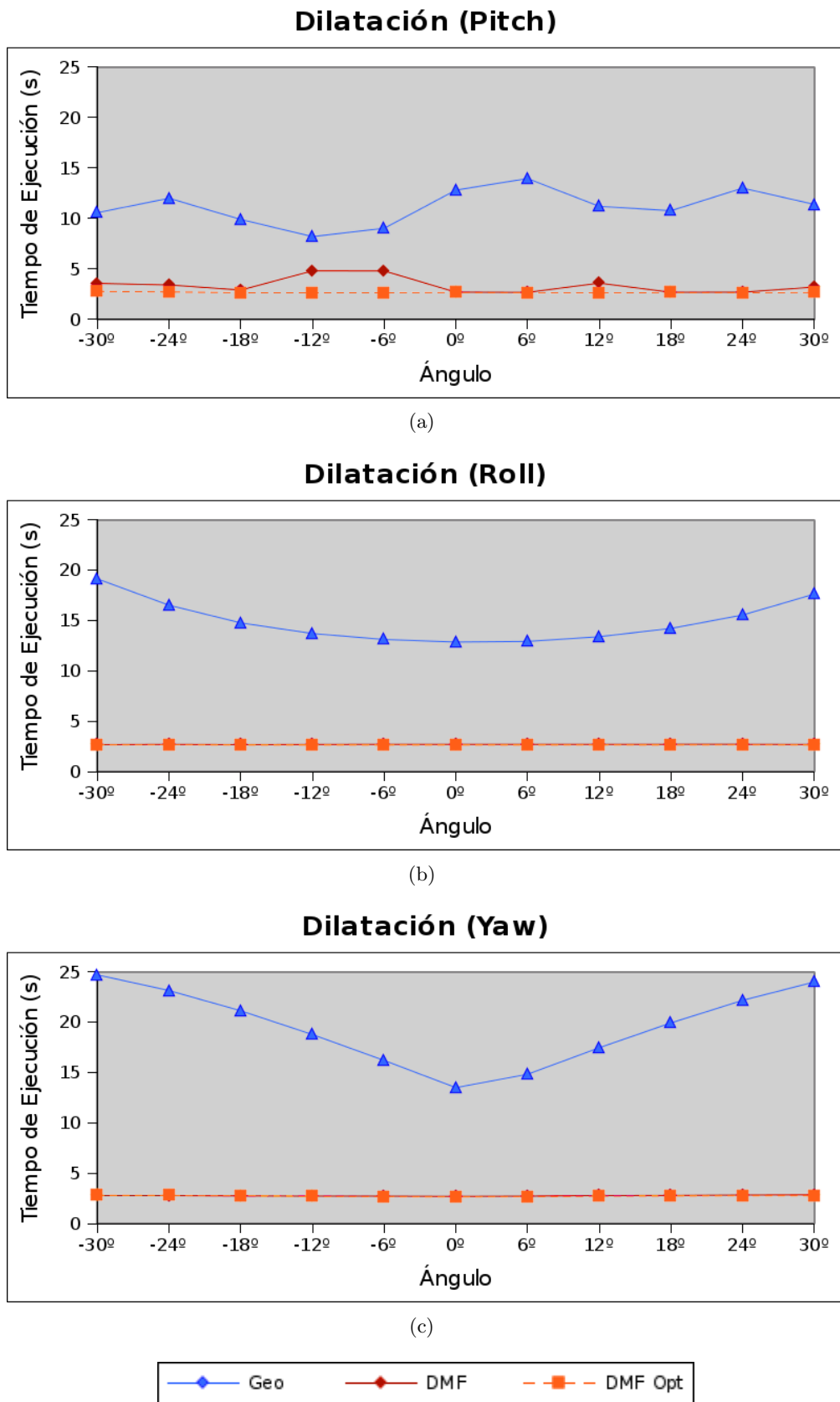


Figura 5.33: Tiempos de ejecución de la operación de dilatación para cada uno de los movimientos de plataforma del dataset sintético en sus versiones geocorregida, DMF y DMF optimizada.

(a) Pitch. (b) Roll. (c) Yaw.

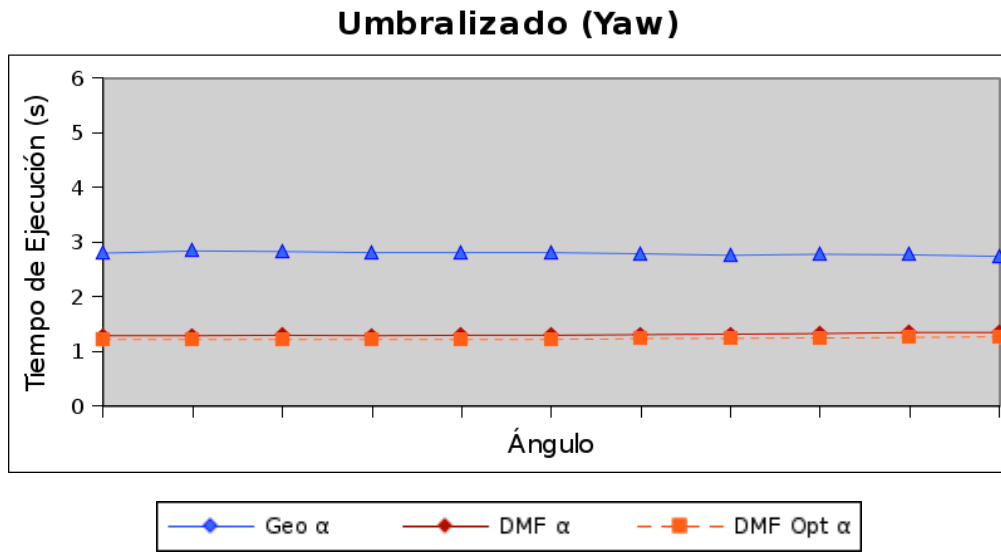


Figura 5.34: *Tiempos de ejecución de la operación de umbralizado para el movimiento de plataforma yaw, considerando ángulo de vuelo y tamaño de celda DMF automático.*

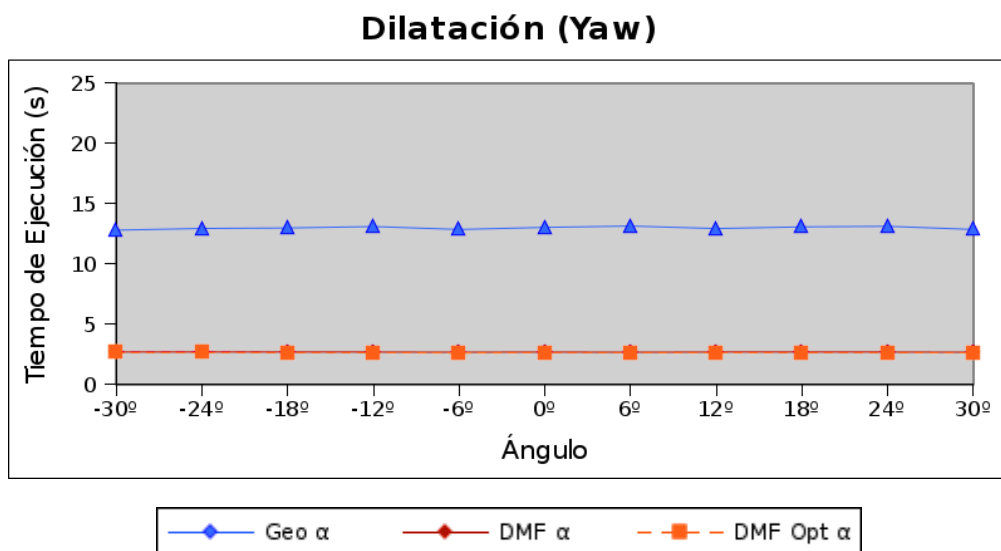


Figura 5.35: *Tiempos de ejecución de la operación de dilatación para el movimiento de plataforma yaw, considerando ángulo de vuelo y tamaño de celda DMF automático.*

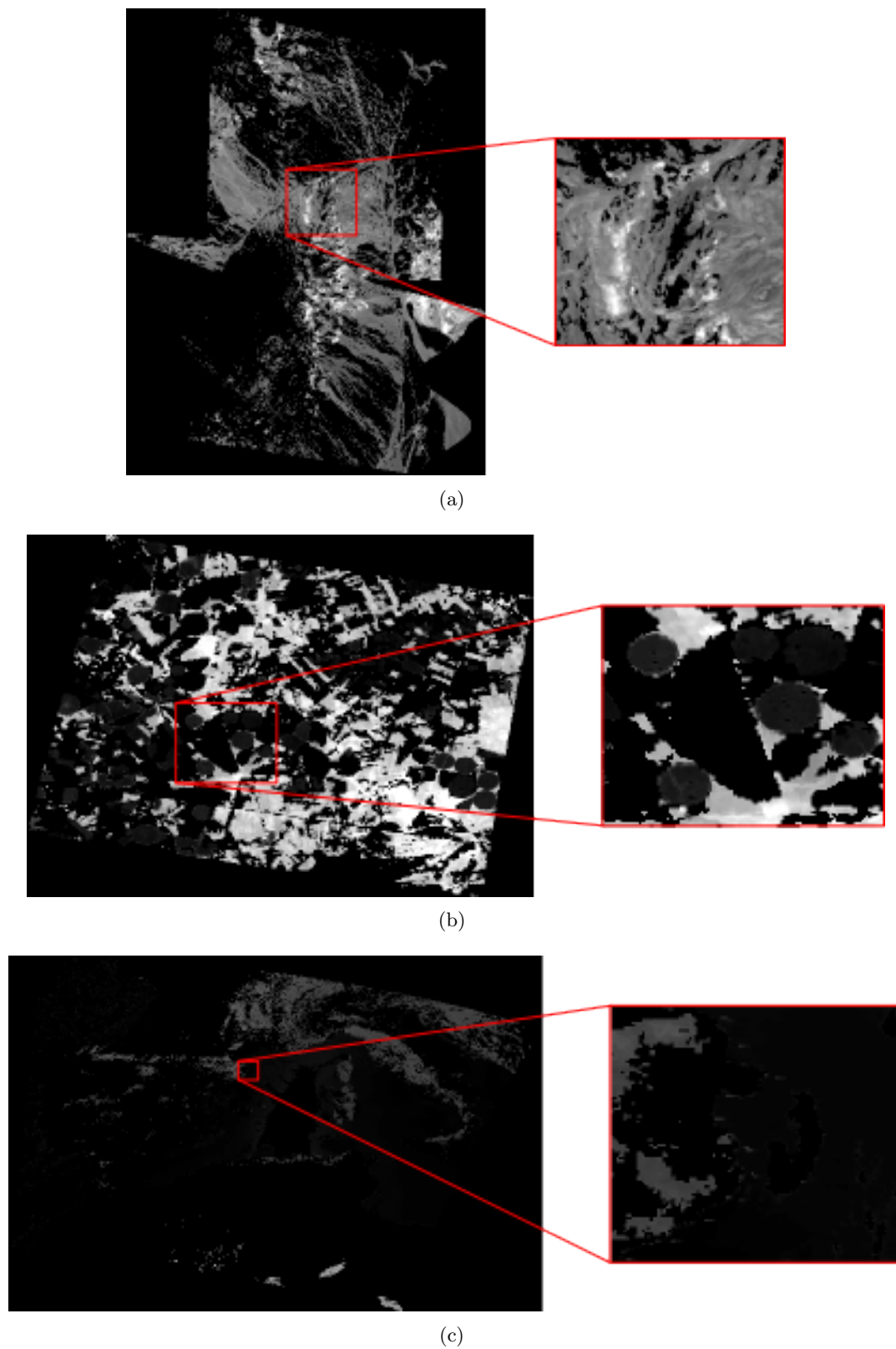
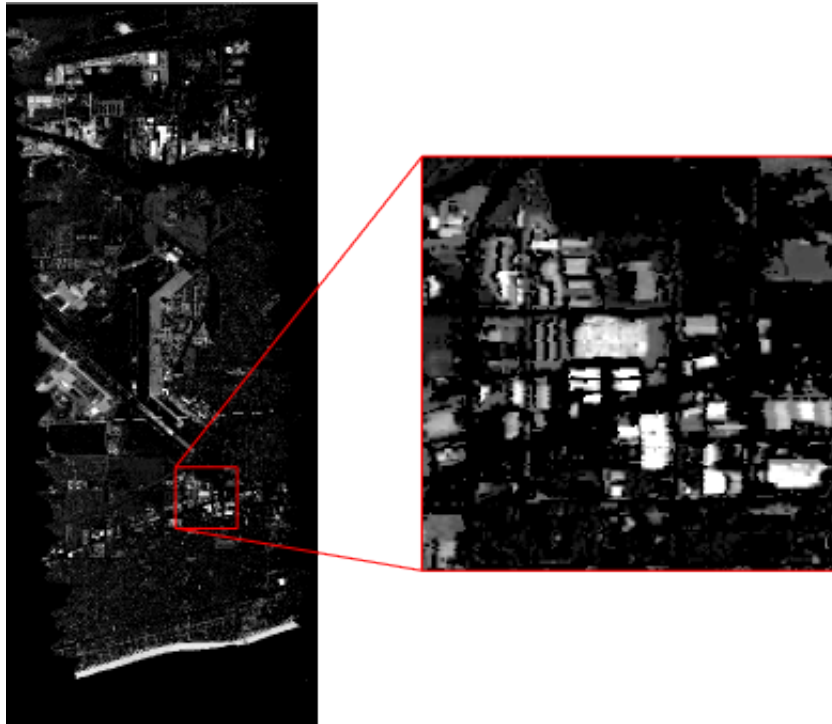
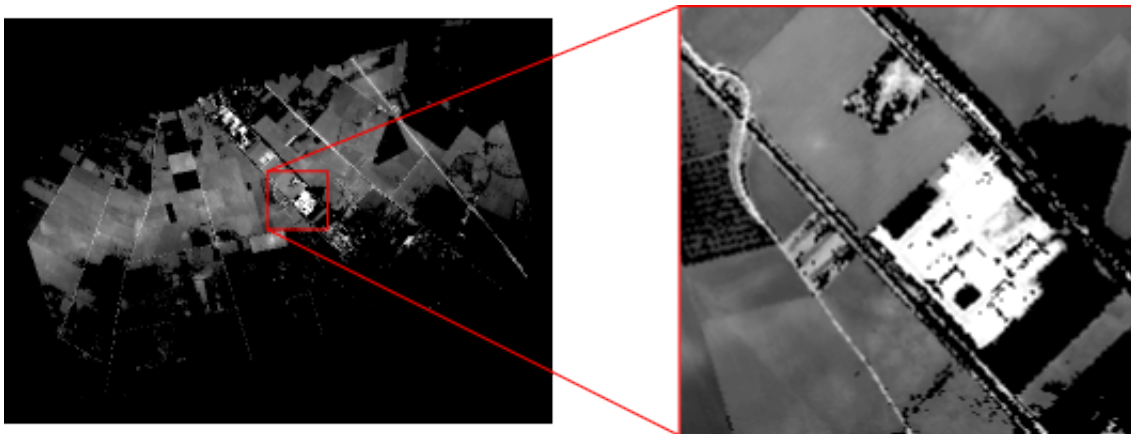


Figura 5.36: Visualización de las imágenes del dataset real (formato DMF) tras aplicar la operación de umbralizado.
(a) Escena Cuprite [Banda 10]. (b) Escena Barrax [Banda 5]. (c) Escena Mediterráneo [Banda 4]. (d) Escena GMéxico [Banda 10]. (e) Escena Cantera [Banda 0].



(d)



(e)

Figura 5.36: (Cont.)

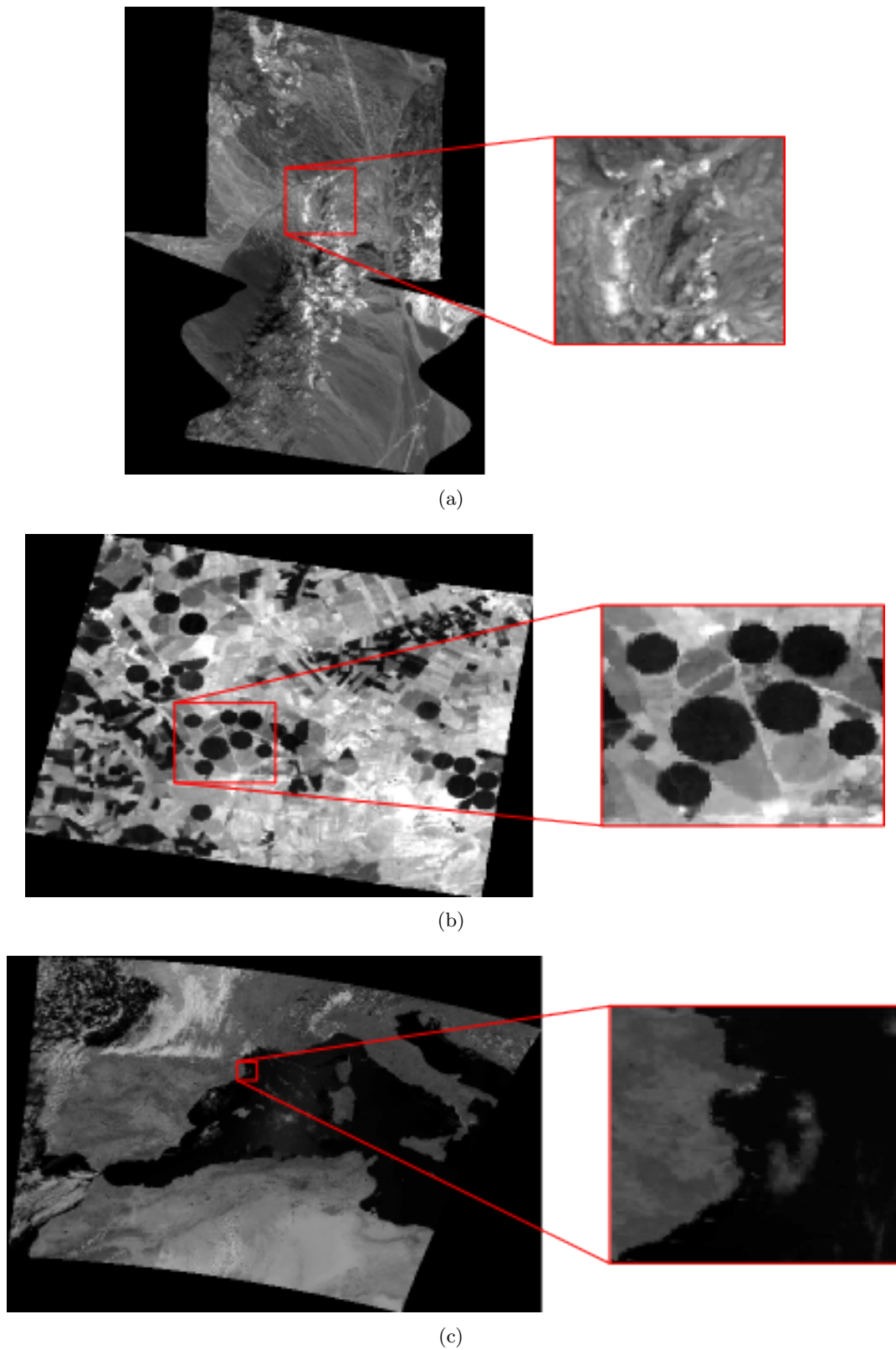
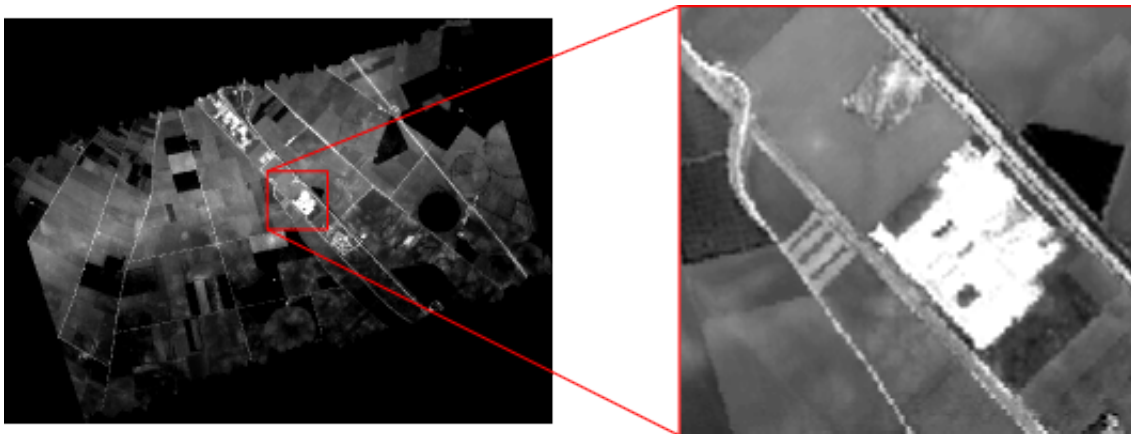


Figura 5.37: Visualización de las imágenes del dataset real (formato geocorregido) tras aplicar la operación de umbralizado.

(a) Escena Cuprite [Banda 10]. (b) Escena Barrax [Banda 5]. (c) Escena Mediterráneo [Banda 4]. (d) Escena GMéxico [Banda 10]. (e) Escena Cantera [Banda 0].



(d)



(e)

Figura 5.37: (Cont.)

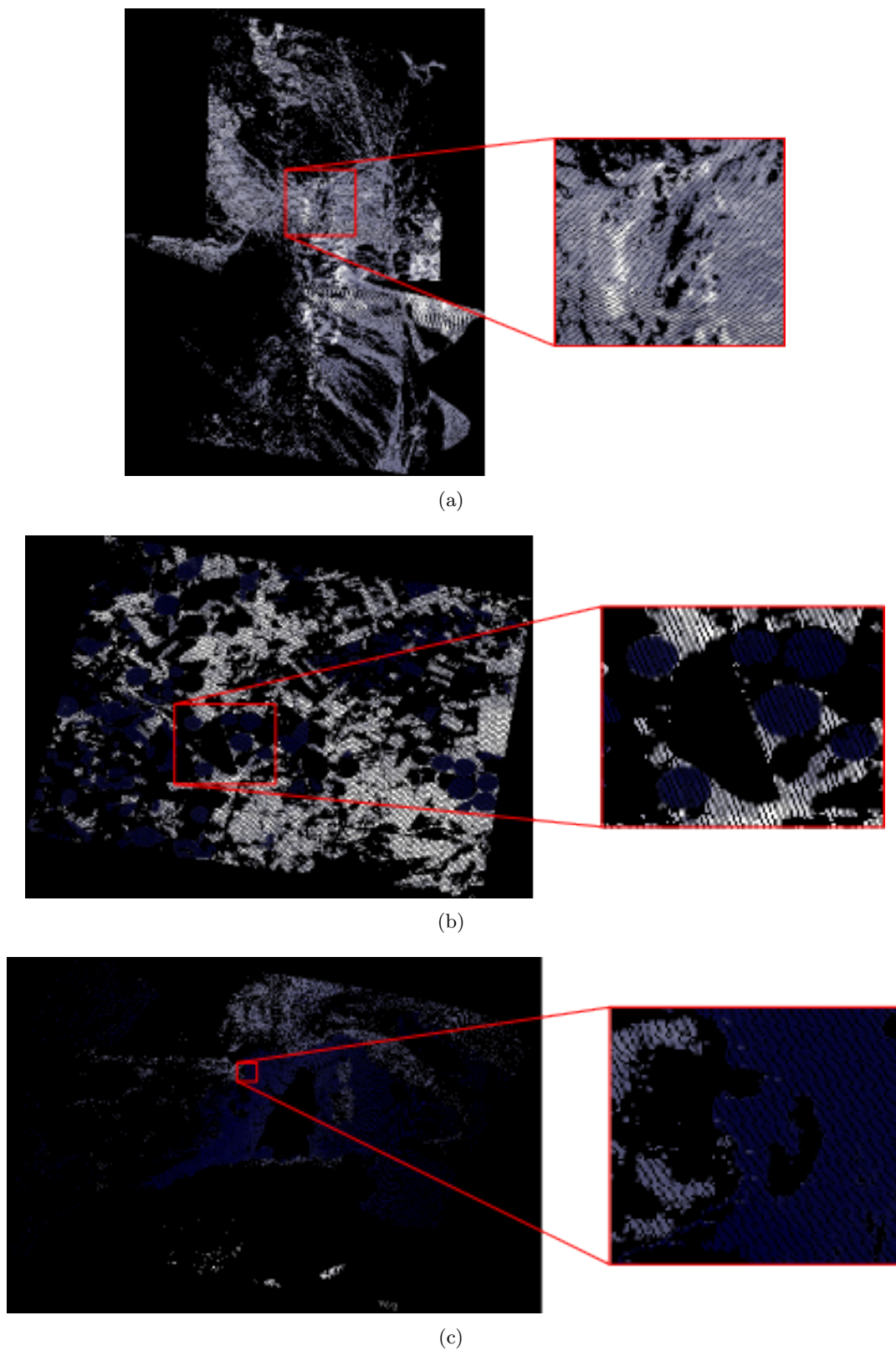
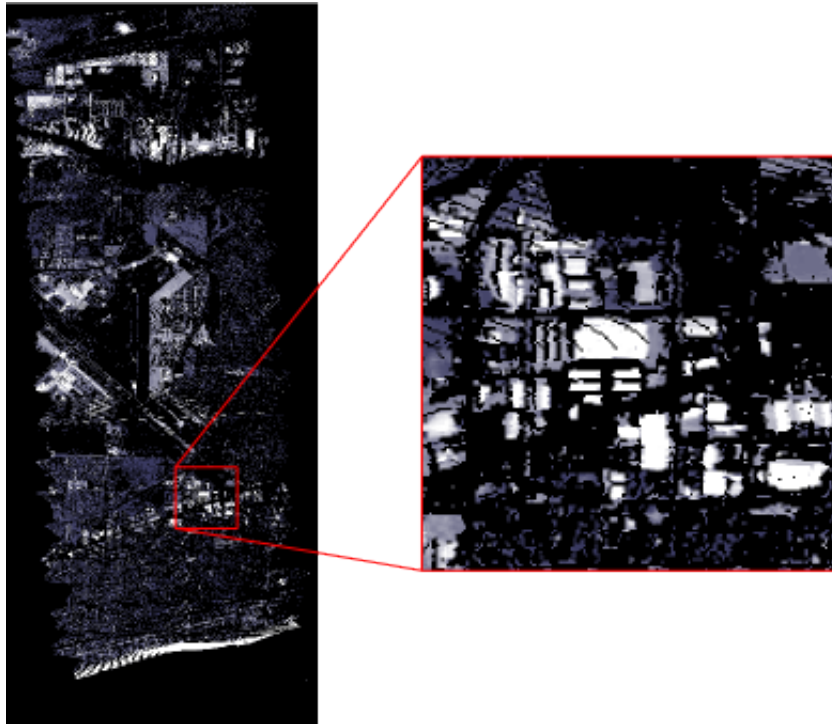
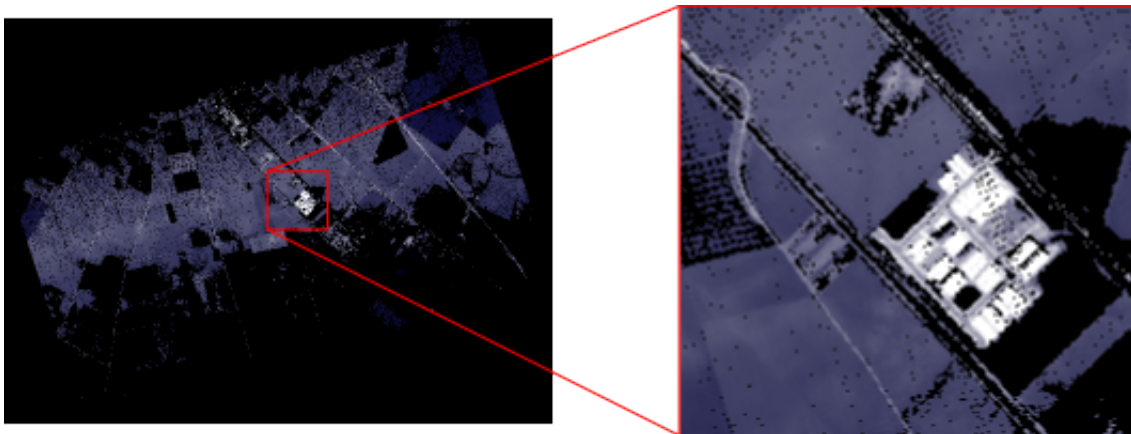


Figura 5.38: Visualización de las imágenes del dataset real (formato geocorregido) tras aplicar la operación de umbralizado.

(a) Escena Cuprite [Banda 10]. (b) Escena Barrax [Banda 5]. (c) Escena Mediterráneo [Banda 4]. (d) Escena GMéxico [Banda 10]. (e) Escena Cantera [Banda 0].



(d)



(e)

Figura 5.38: (Cont.)

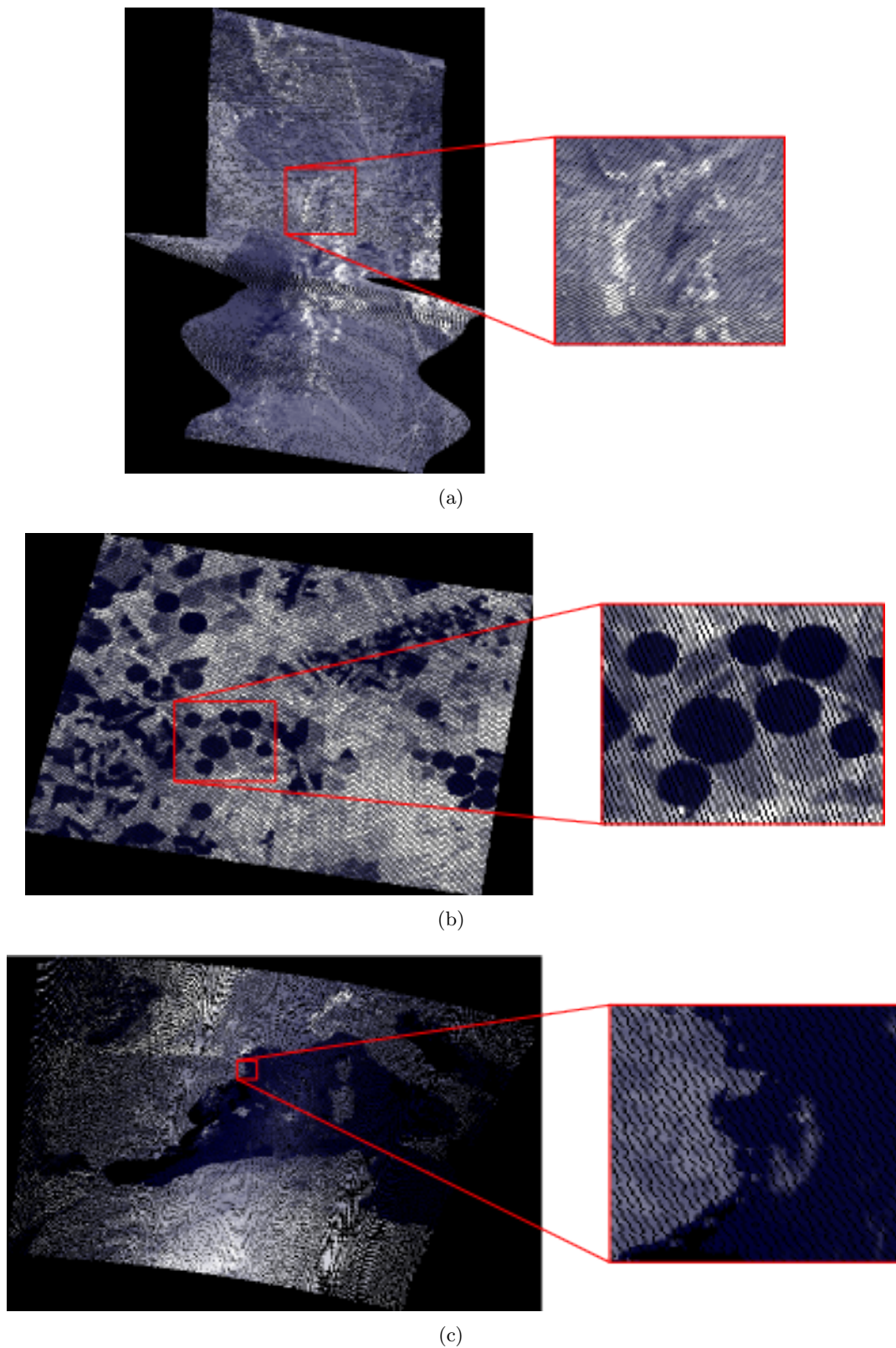
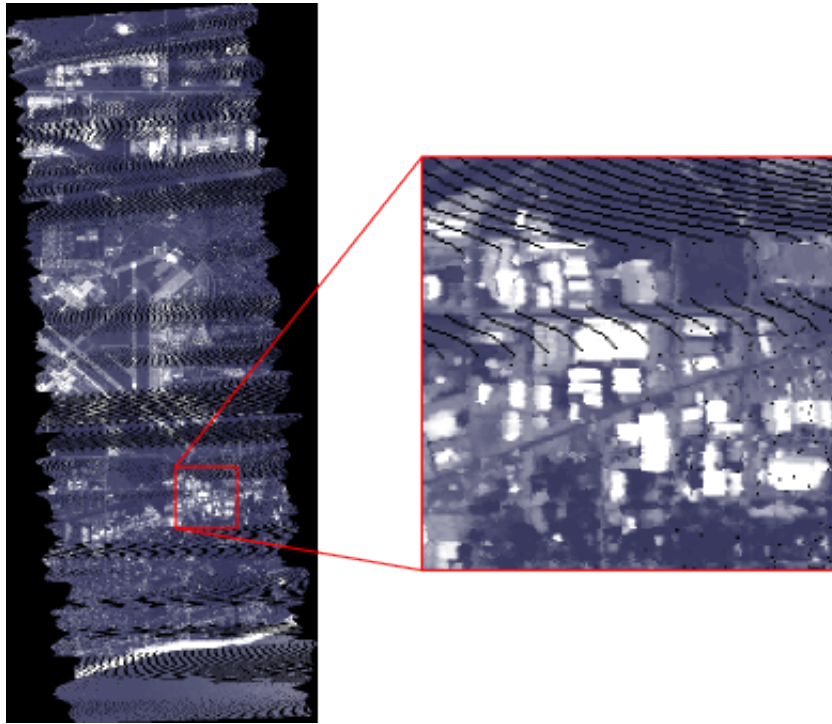
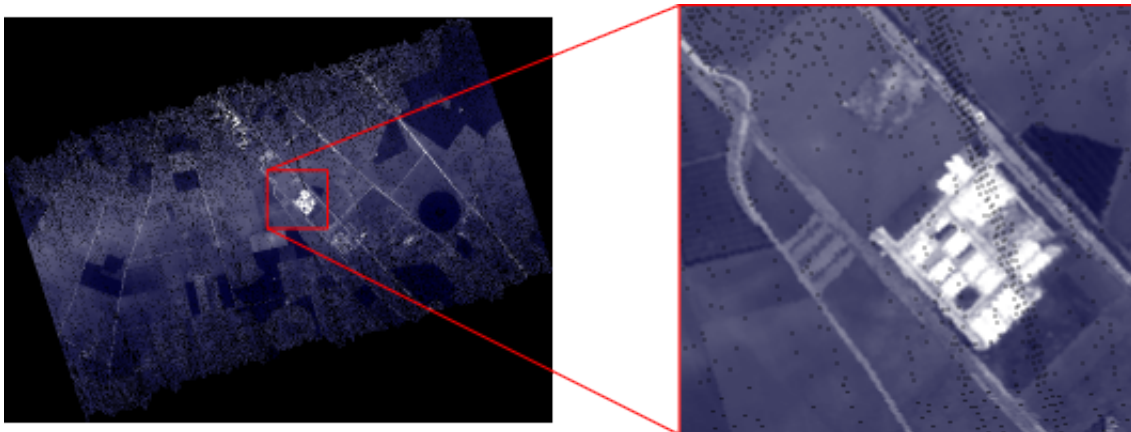


Figura 5.39: Visualización de las imágenes del dataset real (formato DMF) tras aplicar la operación de umbralizado.
(a) Escena Cuprite [Banda 10]. (b) Escena Barrax [Banda 5]. (c) Escena Mediterráneo [Banda 4]. (d) Escena GMéxico [Banda 10]. (e) Escena Cantera [Banda 0].



(d)



(e)

Figura 5.39: (Cont.)

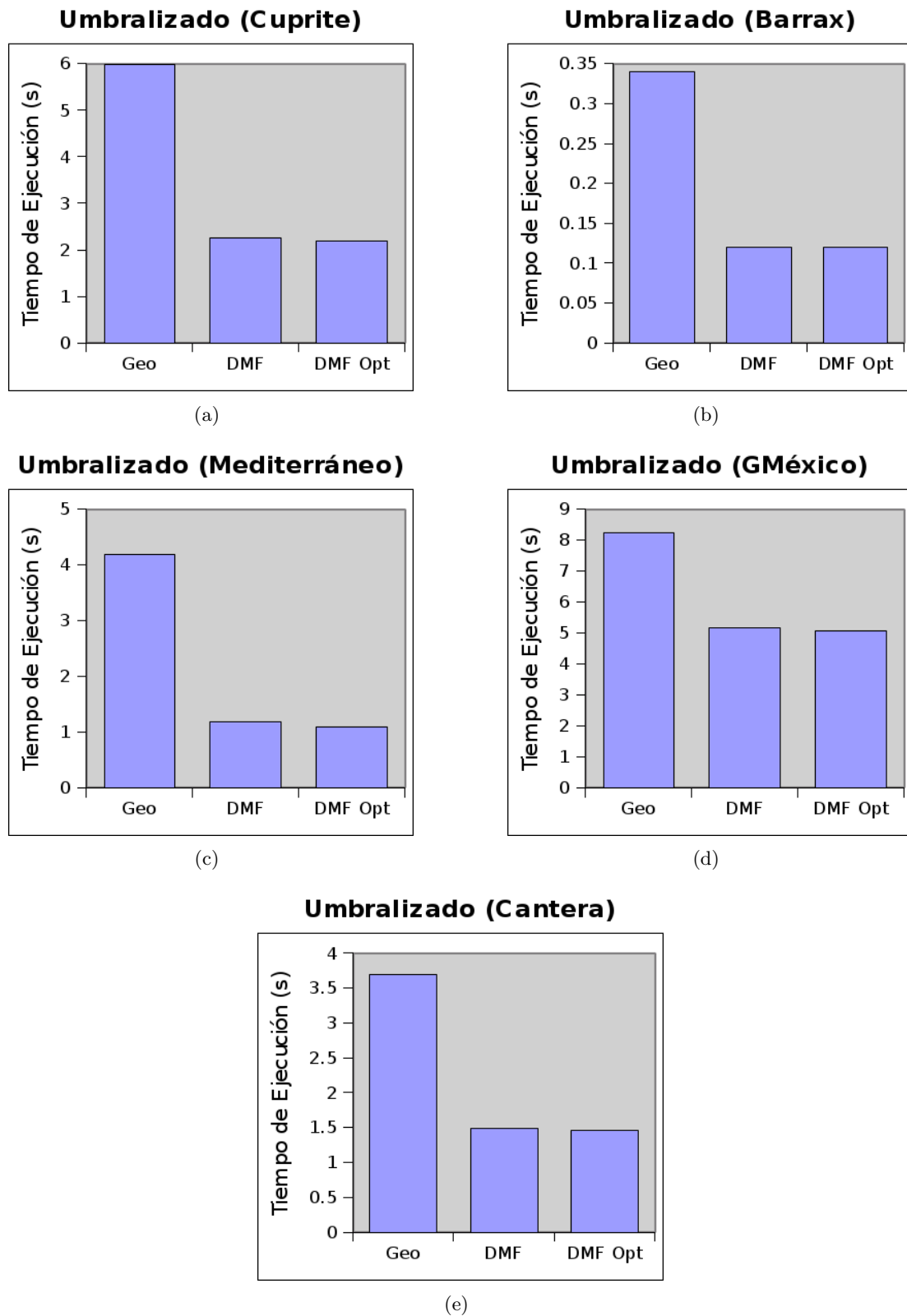


Figura 5.40: *Tiempos de ejecución de la operación de umbralizado para cada imagen del dataset real en sus versiones geocorregida, DMF y DMF optimizada.*
 (a) Escena Cuprite. (b) Escena Barrax. (c) Escena Mediterráneo. (d) Escena GMéxico. (e) Escena Cantera.

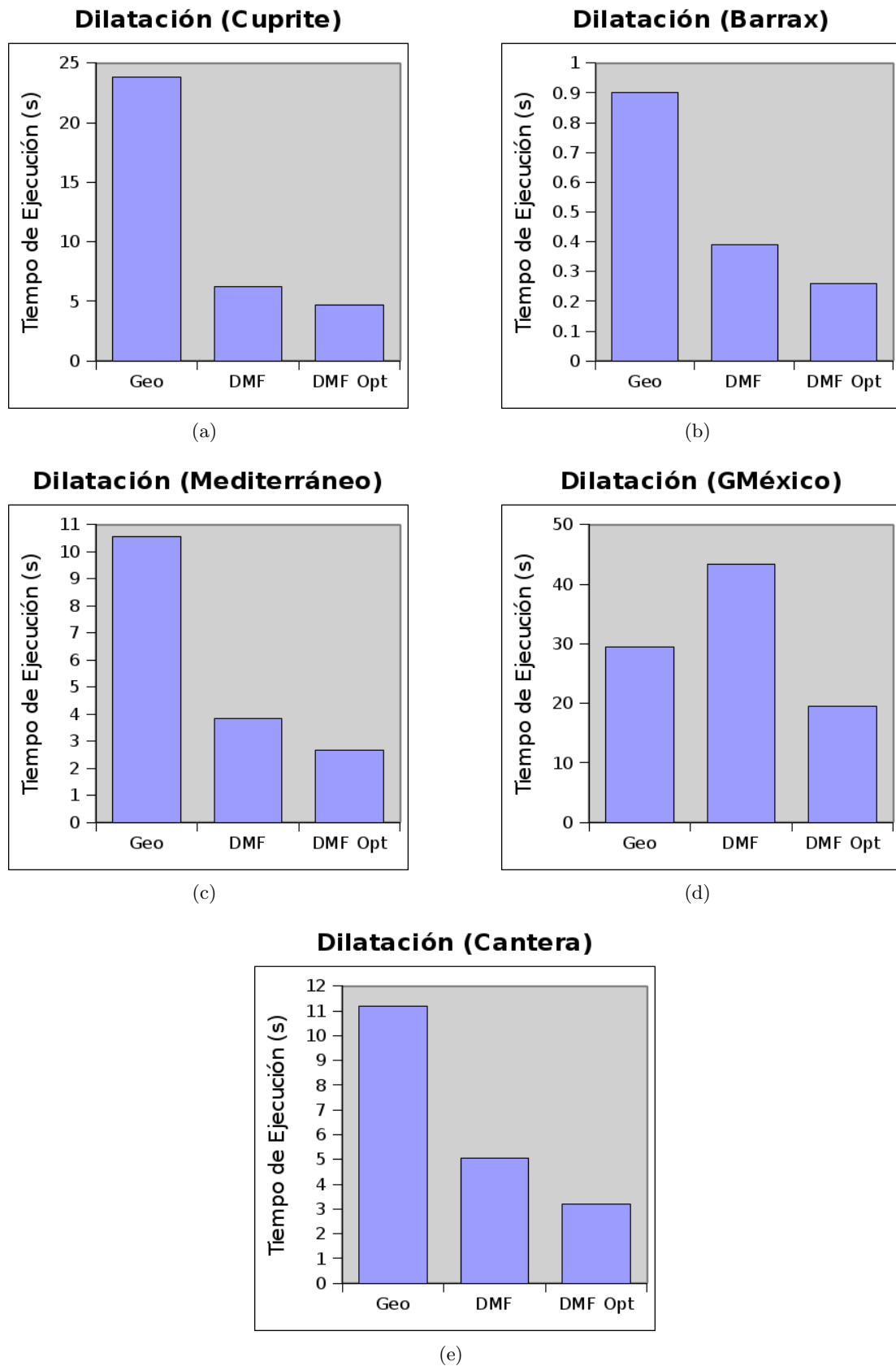


Figura 5.41: Tiempos de ejecución de la operación de dilatación para cada imagen del dataset real en sus versiones geocorregida, DMF y DMF optimizada. (a) Escena Cuprite. (b) Escena Barrax. (c) Escena Mediterráneo. (d) Escena GMéxico. (e) Escena Cantera.

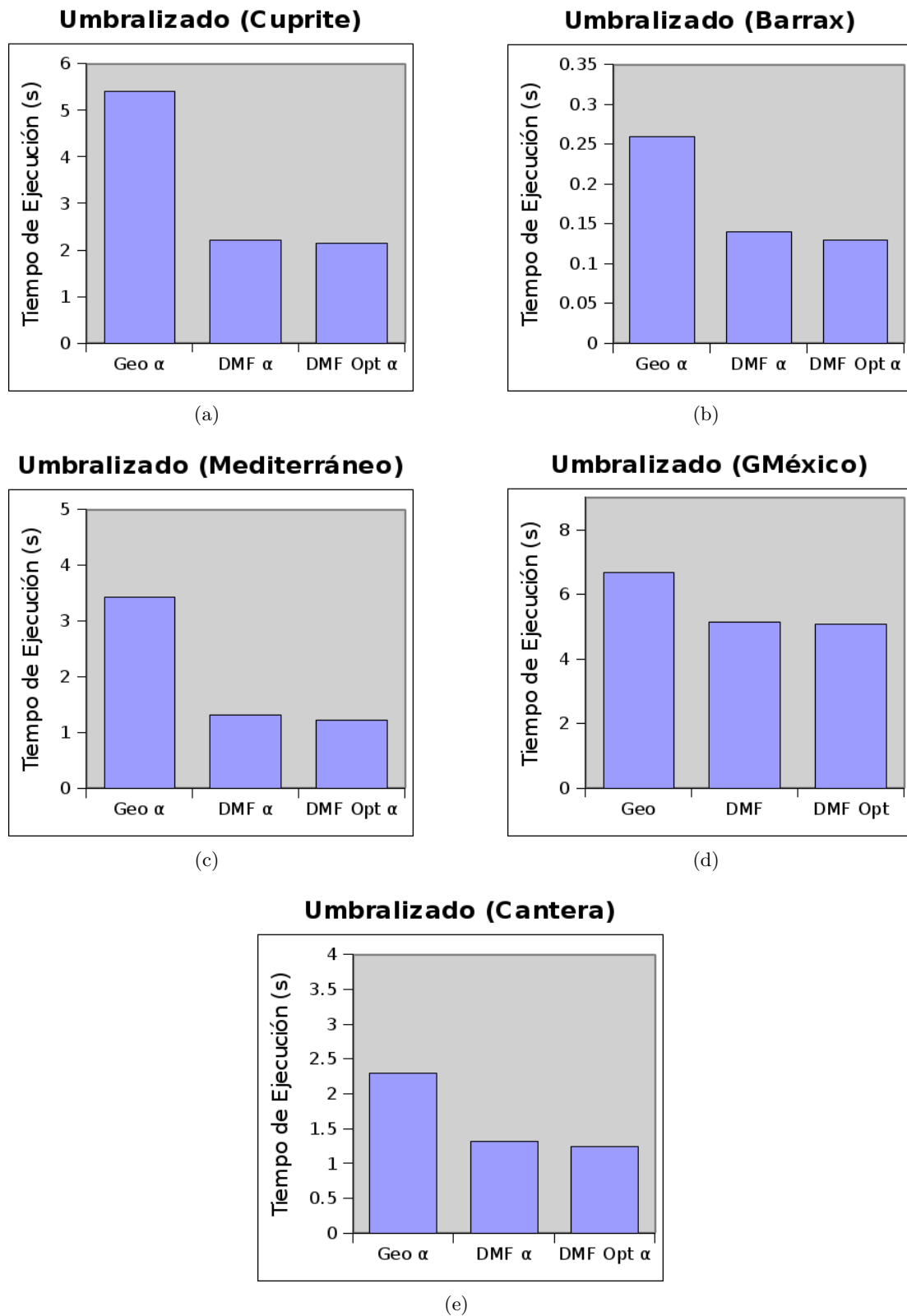


Figura 5.42: Tiempos de ejecución de la operación de umbralizado para cada imagen del dataset real en sus versiones geocorregida, DMF y DMF optimizada, considerando ángulo de vuelo y tamaño de celda DMF automático.

(a) Escena Cuprite. (b) Escena Barrax. (c) Escena Mediterráneo. (d) Escena GMéxico. (e) Escena Cantera.

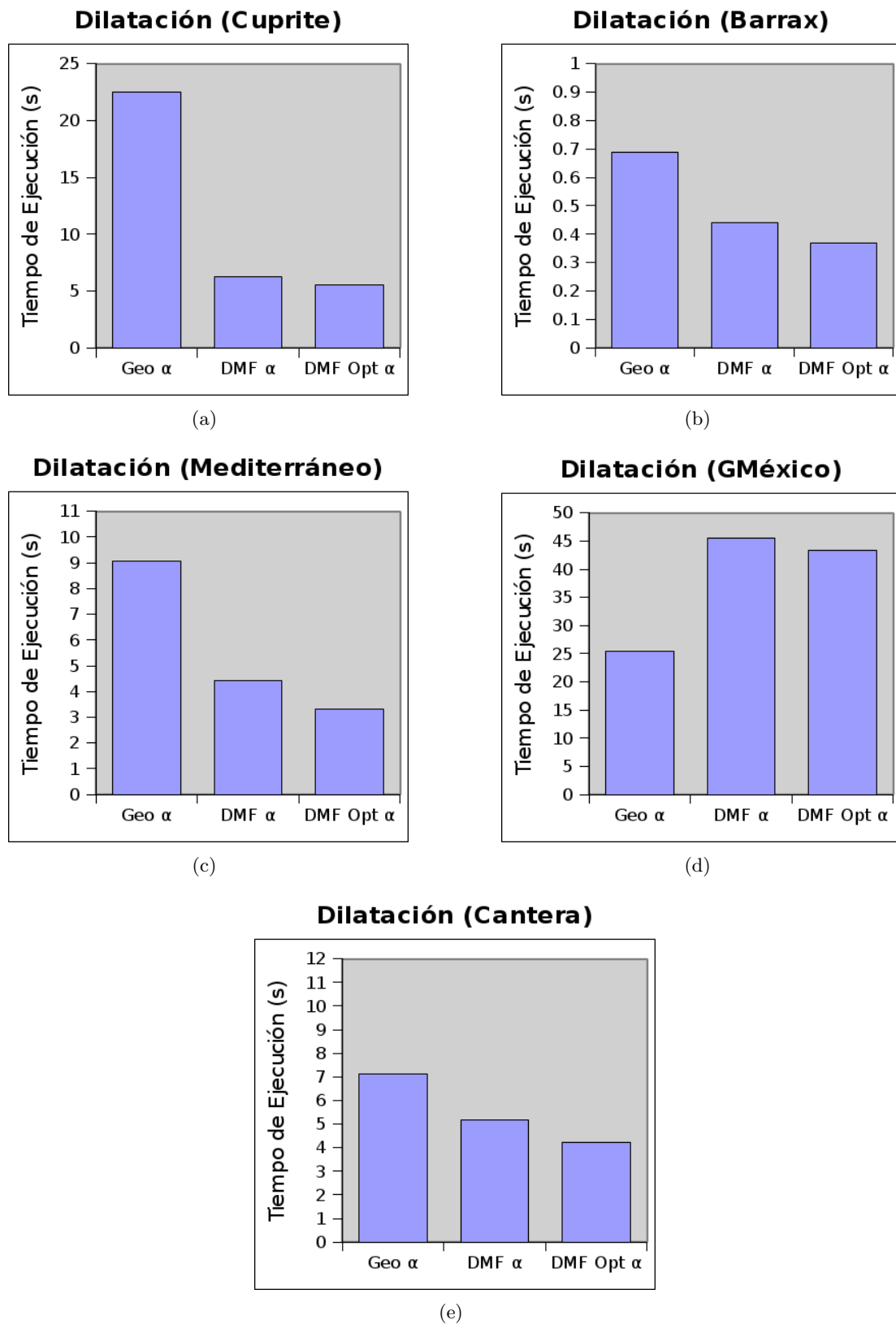


Figura 5.43: Tiempos de ejecución de la operación de dilatación para cada imagen del dataset real en sus versiones geocorregida, DMF y DMF optimizada, considerando ángulo de vuelo y tamaño de celda DMF automático.

(a) Escena Cuprite. (b) Escena Barrax. (c) Escena Mediterráneo. (d) Escena GMéxico. (e) Escena Cantera.

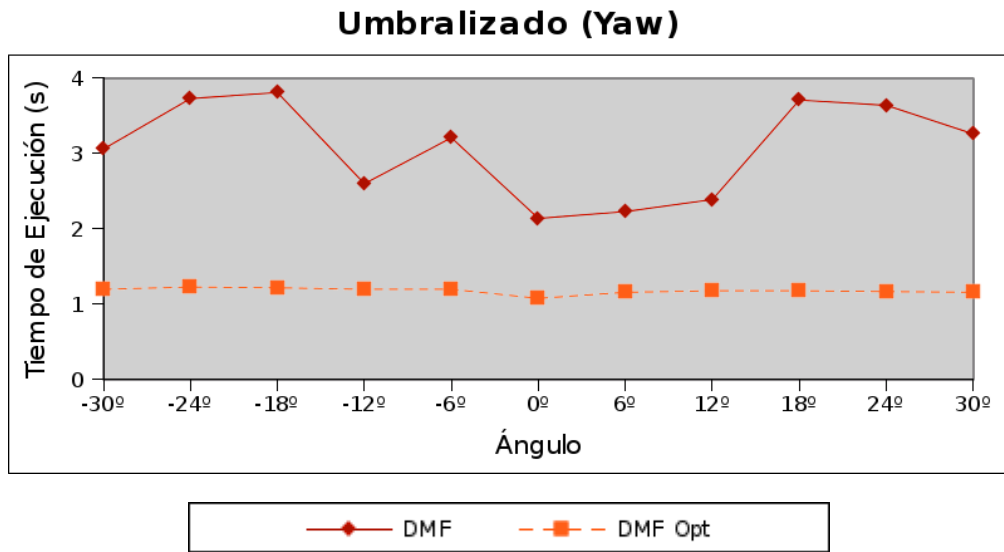


Figura 5.44: Tiempos de ejecución de la operación de umbralizado para el movimiento de plataforma yaw, considerando un tamaño de celda DMF 10 veces menor que el proporcionado por el proveedor.

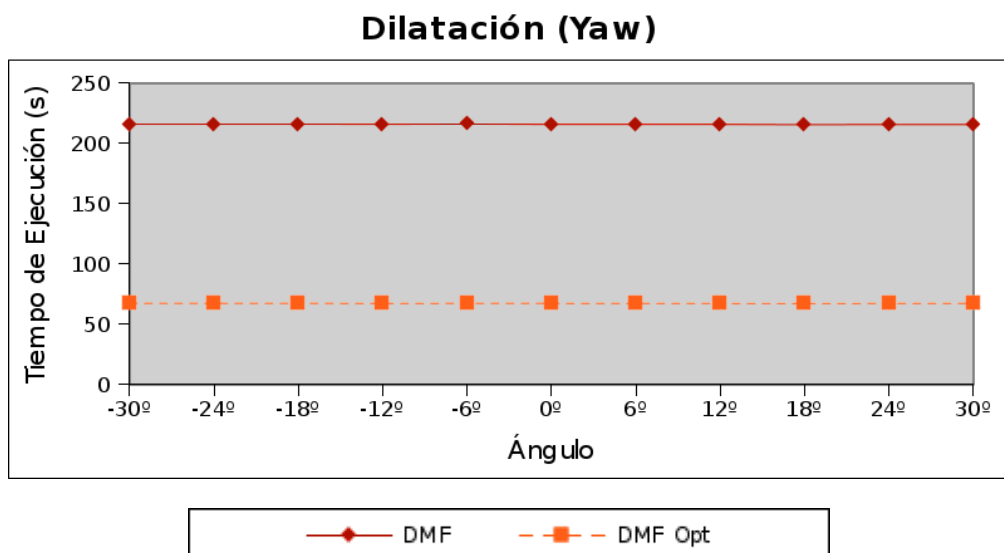


Figura 5.45: Tiempos de ejecución de la operación de dilatación para el movimiento de plataforma yaw, considerando un tamaño de celda DMF 10 veces mayor que el proporcionado por el proveedor.

5.5.4. Análisis de los resultados del experimento.

Para el estudio de los tiempos de ejecución al adaptar las técnicas de procesamiento de imágenes convencionales al formato DMF, se realizaron los siguientes experimentos:

1. Estudio de los tiempos de ejecución para el *dataset sintético*:
 - a) Aspecto visual de las imágenes (formato geocorregido) para la operación de umbralizado (Figura 5.28).
 - b) Aspecto visual de las imágenes (formato geocorregido) para la operación de dilatación (Figura 5.29).
 - c) Aspecto visual de las imágenes (formato DMF) para la operación de umbralizado (Figura 5.30).
 - d) Aspecto visual de las imágenes (formato DMF) para la operación de dilatación (Figura 5.31).
 - e) Cálculo de los tiempos de procesamiento para la operación de umbralizado (medias de 100 iteraciones) (Figura 5.32).
 - f) Cálculo de los tiempos de procesamiento para la operación de dilatación (medias de 100 iteraciones) (Figura 5.33).
 - g) Cálculo de los tiempos de procesamiento para la operación de umbralizado (medias de 100 iteraciones), considerando ángulos de vuelo y tamaños de celda DMF automáticos (Figura 5.34).
 - h) Cálculo de los tiempos de procesamiento para la operación de dilatación (medias de 100 iteraciones), considerando ángulos de vuelo y tamaños de celda DMF automáticos (Figura 5.35).
2. Estudio de los tiempos de ejecución para el *dataset real*:
 - a) Aspecto visual de las imágenes (formato geocorregido) para la operación de umbralizado (Figura 5.36).
 - b) Aspecto visual de las imágenes (formato geocorregido) para la operación de dilatación (Figura 5.37).
 - c) Aspecto visual de las imágenes (formato DMF) para la operación de umbralizado (Figura 5.38).
 - d) Aspecto visual de las imágenes (formato DMF) para la operación de dilatación (Figura 5.39).
 - e) Cálculo de los tiempos de procesamiento para la operación de umbralizado (medias de 100 iteraciones) (Figura 5.40).
 - f) Cálculo de los tiempos de procesamiento para la operación de dilatación (medias de 100 iteraciones) (Figura 5.41).
 - g) Cálculo de los tiempos de procesamiento para la operación de umbralizado (medias de 100 iteraciones), considerando ángulos de vuelo y tamaños de celda DMF automáticos (Figura 5.42).
 - h) Cálculo de los tiempos de procesamiento para la operación de dilatación (medias de 100 iteraciones), considerando ángulos de vuelo y tamaños de celda DMF automáticos (Figura 5.43).

La primera observación que es posible realizar a partir de los resultados obtenidos es el aspecto visual de las imágenes en sus formatos geocorregidos y DMF. Como cabía esperar, para la operación de

umbralizado aproximadamente los espectros de la mitad de los datos han sido asignados un espectro a 0 (en el *dataset sintético* este hecho puede variar ligeramente al utilizar un valor de umbral constante para todos los ángulos de un mismo movimiento de plataforma); para el caso de la dilatación, los objetos más luminosos parecen haberse ensanchado.

Respecto a los tiempos de ejecución, es posible comprobar que las gráficas resultantes son semejantes a las obtenidas en el experimento de tamaño en disco. Esto justifica, obviamente, que los tiempos de ejecución están altamente relacionados con el tamaño de la imagen, tal que tamaños grandes implican mayores tiempos de procesamiento y viceversa. Es posible comprobar más fácilmente este fenómeno en las gráficas correspondientes al *dataset sintético* (Figura 5.32 y Figura 5.33) en el que los valores de las series siguen la misma morfología que las gráficas análogas en el experimento de tamaño en disco (Figura 5.22).

Otro aspecto que sí queda patente en los resultados es que los tiempos de ejecución cuando se utiliza el formato DMF, respecto al formato geocorregido convencional, son siempre bastante menores, implicando un speed-up de hasta aproximadamente 3X. El hecho de considerar celdas a *Null* ahorra un tiempo de procesamiento que sí es empleado en analizar píxeles inválidos (píxeles a los que no les deberían haber sido asignados espectros, pero que almacenan una copia del espectro más cercano) y píxeles fuera de la escena (con valores a 0). La diferencia entre los tiempos de procesamiento para las diferentes formatos se incrementa para la operación de la dilatación, en donde un píxel inválido o fuera de la escena llega a ser analizado y procesado hasta 9 veces (para el caso de aplicación de un *kernel* de 3×3), permaneciendo a *Null* la celda DMF correspondiente. El hecho de utilizar modificación previamente presentada del formato DMF en la que se realiza una indexación de los datos dentro de las celdas DMF mediante árboles Kd-tree, y una lista secuencial de punteros a todos los registros DMR almacenados (*DMF Opt*) no supone un decremento considerable de los tiempos de ejecución respecto al formato DMF original. Esto es debido a dos motivos principales. Para el caso de la operación de dilatación, la densidad de registros DMR por celda DMF es tan baja, que no permite construir árboles lo suficientemente extensos como para suponer una mejoría. Por el otro lado, para el caso de la operación de umbralizado, el número de celdas DMF a *Null* no es lo suficientemente elevado como para experimentar un decremento considerable en los tiempos de procesamiento si éstas no son analizadas. Por tanto, la modificación presentada del formato DMF solamente es realmente útil cuando la densidad de registros DMR por celda DMF es elevada, y cuando el porcentaje de celdas DMF a *Null* sea también alto. Para corroborar este hecho, se analizaron los tiempos de ejecución para el caso del movimiento de plataforma *yaw* del *dataset sintético*, el cual presenta mayores diferencias de tiempo de acuerdo al ángulo. Para cada imagen correspondiente, se consideró un tamaño de píxel 10 veces inferior al proporcionado por el proveedor para la operación de umbralizado (generando una matriz DMF de mayor tamaño con un porcentaje de celdas DMF a *Null* considerablemente mayor); y un tamaño de píxel 10 veces superior para la operación de dilatación (generando una matriz DMF de menor tamaño con una densidad de registros DMR por celda DMF considerablemente mayor). Los resultados obtenidos (representados en la Figura 5.44 y en la Figura 5.45) representan un decremento de los tiempos de procesamiento del orden de segundos para la operación de umbralizado, y del orden de minutos para la operación de dilatación.

Cuando se aplica una rotación de la imagen respecto al ángulo de vuelo, así como un tamaño de celda DMF calculado de manera automática, los tiempos de ejecución se reducen tanto para las versiones DMF como para la versión geocorregida, siendo este incremento mayor cuanto mayor es el ángulo de vuelo (ya que se corrige el hecho de considerar píxeles y celdas DMF fuera de la escena). Los tiempos de procesamiento de la versión DMF modificada (*DMF Opt α*) continúan siendo ligeramente inferior que los de la versión DMF original en este caso.

Sólo existe un único caso en el que los tiempos de ejecución son ligeramente superiores para el formato DMF respecto a la versión geocorregida, y éste es el correspondiente a la escena de GMéxico. La explicación de este incremento es que el tamaño de celda DMF calculado de manera automática ($0.000036^\circ \times 0.00003^\circ$) es lo suficientemente menor que el proporcionado por el proveedor ($0.000036^\circ \times 0.000036^\circ$) como para construir una matriz DMF de mayor tamaño (mayor número de filas y columnas),

implicando un mayor consumo en los tiempos de procesamiento.

5.6. Precisión espacial.

El formato DMF ha sido diseñado para optimizar la manera en la que se almacenan, en las imágenes, los datos recogidos por los sensores, así como para facilitar su acceso, manipulación y distribución a la comunidad científica. Al estar basado en una unidad básica que representa una medición (registro DMR), que permite almacenar, junto con el resto de información, la posición espacial donde fue adquirida esa medición, este formato supone una gran alternativa a la hora de georreferenciar todos los datos obtenidos por los instrumentos a bordo del avión y/o satélite.

5.6.1. *Objetivos del experimento.*

Es importante analizar qué precisión a la hora de localizar y georreferenciar los datos se consigue mediante la utilización del formato DMF, respecto a sus versiones geocorregidas mediante procedimientos convencionales. Al almacenar cada registro DMR la información de posicionamiento de la medición correspondiente, junto con la información recogida por los sensores, la precisión espacial queda optimizada al máximo, siendo únicamente limitada por la precisión del dispositivo GPS/INS. De esta manera, las metodologías de procesamiento de imágenes hiperespectrales adaptadas al formato DMF consiguen unos resultados más precisos y fiables que con los formatos de almacenamiento convencionales.

Con este experimento se pretenden alcanzar los siguientes objetivos:

- Estudiar el porcentaje de píxeles a los que se les asignó originalmente una medición, respecto a aquéllos a los que les fue asignado una medición por vecino más próximo y los que almacenan espectros a 0 por estar situados fuera de la escena tras el proceso de geocorrección.
- Analizar la distancia de cada medición de la imagen geocorregida respecto a su localización original (adquirida por el dispositivo GPS/INS).
- Analizar el número de veces que cada medición del fichero L0 ha sido utilizada para construir la versión geocorregida.
- Estudiar la repercusión al introducir errores espaciales durante el proceso de geocorrección respecto a la identificación y procesamiento espectral de las mediciones respecto a un *ground-truth*.

5.6.2. *Descripción del experimento.*

Este apartado tiene como objetivo estudiar las imprecisiones introducidas en el proceso de geocorrección, para lo cual se realizaron un total de diferentes experimentos que se basan más en descubrir las carencias presentes en estas versiones de almacenamiento y representación de datos, que de mostrar la eficiencia y ventajas de utilizar el formato DMF.

El primer estudio se basa en el análisis de cuántos píxeles de la imagen geocorregida final les han sido realmente asignados su medición correspondiente, a cuántos se les han asignado una medición por vecino más próximo al quedar sin asignar en un principio, y cuántos almacenan un espectro a 0 por estar fuera de la escena. Esto puede realizarse mediante un estudio del fichero GLT proporcionado junto con los datos geocorregidos. Para todo píxel p , el primer caso corresponderá a aquéllos cuyo valor en el fichero GLT es menor a 0 ($GLT_p < 0$); el segundo caso corresponderá a aquéllos cuyo valor

en el fichero GLT es mayor a 0 ($GLT_p > 0$); y el tercer caso corresponderá a aquéllos cuyo valor en el fichero GLT es igual a 0 ($GLT_p = 0$).

El segundo estudio tiene como objetivo analizar los desplazamientos de las mediciones introducidos por el proceso de geocorrección respecto a las coordenadas originales de cada medición. Para cada píxel de cada imagen geocorregida se calcularon las coordenadas de su centro ($Long_p$ y Lat_p) considerando la esquina superior izquierda referencia y el tamaño de píxel, tal y como se realiza en los procedimientos de geocorrección convencionales (Ecuación 3.5). Posteriormente, se recuperaron las coordenadas originales (proporcionadas por el sistema de posicionamiento) de la medición almacenada en dicho píxel ($Long_m$ y Lat_m) determinando la posición de la misma (identificadores del fichero GLT) en el fichero IGM. La distancia entre el posicionamiento geocorregido y el posicionamiento original (d) fue calculado mediante:

$$d = \sqrt{(Long_p - Long_m)^2 + (Lat_p - Lat_m)^2} \quad (5.5)$$

El tercer estudio analiza el aprovechamiento de los datos originales obtenidos por los sensores al construir la imagen geocorregida. Como a la hora de asignar una medición a un píxel siempre se selecciona la más cercana al centro del mismo, existen casos en los que ninguna medición original es candidata a ser almacenada en la imagen geocorregida final, o bien una misma medición es seleccionada para representar varios píxeles. Un análisis simple a los identificadores presentes en el fichero GLT es suficiente para detectar el porcentaje de mediciones originales que no han sido consideradas en la construcción de la imagen geocorregida.

En el cuarto estudio, se analizó cómo los errores espaciales introducidos en el proceso de geocorrección afectan a las imágenes resultantes mediante un caso real. En este experimento en particular, se utilizaron dos imágenes diferentes de la escena de Cantera. La primera de ellas, es la que se ha utilizado en todos los experimentos anteriores y está presentada en la Figura 5.7(i) y en la Figura 5.7(j). Sin embargo, ésta fue recortada a una subescena característica, en la que es posible apreciar varias naves industriales (Figura 5.46(b)). La segunda imagen ($Cantera_{GT}$) (Figura 5.46(a)), se trata de un vuelo cubriendo el mismo área en una segunda pasada instantes después pero a menos altitud. Por lo tanto, al tener mejor resolución y precisión espaciales, se consideró como *ground-truth*.

El objetivo de este cuarto estudio es convertir la subescena de Cantera a sus versiones geocorregida y DMF, y compararlas con la información almacenada en el *ground-truth*, tal y como se detalla en el siguiente procedimiento:

1. A partir de la imagen L0 de la subescena de Cantera ($SubCantera_{L0}$) y del correspondiente fichero IGM, se obtuvo la versión geocorregida ($SubCantera_{Geo}$) (Figura 5.46(c)).
2. A partir de la imagen L0 de la subescena de Cantera ($SubCantera_{L0}$) y del correspondiente fichero IGM, se obtuvo la versión DMF ($SubCantera_{DMF}$) (Figura 5.46(d)).
3. Para cada píxel de $SubCantera_{Geo}$ ($P_SubCantera_{Geo}$), se localizó la medición más cercana en la imagen *ground-truth* ($M_Cantera_{GT}$), y se calculó el ángulo (α) entre el espectro de $P_SubCantera_{Geo}$ (x) y el espectro de $M_Cantera_{GT}$ (y) tal que:

$$\alpha = \arccos \left(\frac{x \cdot y}{|x||y|} \right) \quad (5.6)$$

El ángulo espectral medio de todos los píxeles de $SubCantera_{Geo}$ indicará el error espacial cometido para la versión geocorregida.

4. Para cada medición de $SubCantera_{DMF}$ ($M_SubCantera_{DMF}$), se localizó la medición más cercana en la imagen *ground-truth* ($M_Cantera_{GT}$), y se calculó el ángulo (α) entre el espectro

de $M_{SubCantera_{DMF}}(x)$ y el espectro de $M_{Cantera_{GT}}(y)$ tal y como se indica en Ecuación 5.6. El ángulo espectral medio de todos los píxeles de $SubCantera_{DMF}$ indicará el error espacial cometido para la versión DMF.

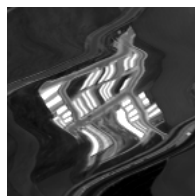
Las características de las imágenes utilizadas en este estudio se encuentran detalladas en la Tabla 5.11.

	[Filas \times Columnas \times Bandas]	TamPíxel
$SubCantera_{L0}$	[143 \times 143 \times 63]	—
$SubCantera_{Geo}$	[186 \times 149 \times 63]	4.4 m
$SubCantera_{DMF}$	[186 \times 149 \times 63]	4.4 m
$SubCantera_{GT}$	[1483 \times 750 \times 63]	—

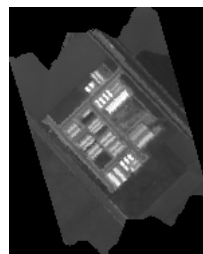
Tabla 5.11: Descripción de las imágenes utilizadas para el estudio de la precisión espacial aplicado a un caso real.



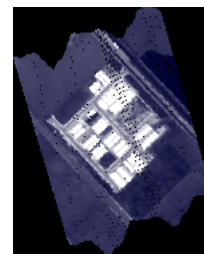
(a)



(b)



(c)



(d)

Figura 5.46: Subimágenes de la escena de Cantera utilizadas para el estudio de la precisión espacial aplicado a un caso real.

(a) Imagen a baja altura (ground-truth) [Banda 0]. (b) Imagen L0 [Banda 0]. (c) Versión geocorregida [Banda 0]. (d) Versión DMF [Banda 0].

5.6.3. Resultados del experimento.

En un primer lugar, los estudios detallados en la sección anterior fueron aplicados al *dataset sintético*, con el objetivo de analizar las imprecisiones introducidas en el proceso de geocorrección a un nivel teórico.

El primer estudio consistió en analizar a cuántos píxeles de la imagen geocorregida final les habían sido realmente asignados su medición correspondiente, a cuántos se les habían asignado una medición por vecino más próximo al quedar sin asignar en un principio, y cuántos almacenaban un espectro a 0 por estar fuera de la escena. Mediante el estudio del fichero GLT proporcionado junto con los datos geocorregidos se consideraron diferentes tipos de píxel $GLT_p < 0$, $GLT_p > 0$ y $GLT_p = 0$. Como análisis visual rápido, se construyó una imagen análoga a la imagen geocorregida original, en donde los píxeles $GLT_p = 0$ han sido pintados de color negro, los píxeles $GLT_p > 0$ de color verde, y los píxeles $GLT_p < 0$ de color rojo. La Figura 5.47 muestra las imágenes coloreadas resultantes para diversos ángulos del *dataset sintético*. Posteriormente, se calculó el porcentaje medio de cada tipo de píxel ($GLT_p = 0$, $GLT_p < 0$ y $GLT_p > 0$) para cada imagen completa, y fue comparada con el porcentaje de celdas DMF ocupadas ($DMF_c \neq Null$) y a $Null$ ($DMF_c = Null$) de las versiones DMF análogas. La Figura 5.48 muestra dichos porcentajes, en donde las series azules representan los resultados obtenidos para las versiones geocorregidas, de tal manera que la línea azul clara corresponde al caso $GLT_p = 0$, la azul oscura punteada al caso $GLT_p > 0$, la azul oscura segmentada al caso $GLT_p < 0$ y la azul oscura continua al total de píxeles con medición asignada ($GLT_p < 0 + GLT_p > 0$). De igual modo, las series rojas representan los resultados obtenidos para las versiones DMF análogas, de tal manera que la línea roja clara corresponde al porcentaje de celdas DMF a $Null$ y la roja oscura al porcentaje de celdas DMF con una o más mediciones. El mismo procedimiento fue realizado para las imágenes considerando el ángulo de vuelo (movimiento de plataforma *yaw*), y para la versión DMF, también el tamaño de celda DMF calculado de manera automática. La Figura 5.49 muestra las imágenes coloreadas resultantes para diversos ángulos del movimiento de plataforma *yaw* de acuerdo a si se tratan de píxeles almacenando su medición original correspondiente (verde), o una medición asignada por vecino más próximo (rojo). La Figura 5.50 muestra los porcentajes de cada tipo de píxel de acuerdo al tipo de asignación de medición ($GLT_p = 0$, $GLT_p < 0$ y $GLT_p > 0$ para el caso de la versión geocorregida, y $DMF_c = Null$ y $DMF_c \neq Null$ para la versión DMF).

El segundo estudio se basó en el análisis de los desplazamientos de las mediciones introducidos por el proceso de geocorrección respecto a las coordenadas originales de cada medición. El primer resultado obtenido lo proporciona la Figura 5.51, en la que se muestran imágenes análogas a las originales, y donde el color de cada píxel representa la distancia de la posición geocorregida de la medición almacenada respecto a la posición proporcionada por el sistema de posicionamiento (d), de tal manera que cuanto más blanca sea la representación, mayor es la distancia entre ambas posiciones. La distancia media global también fue calculada para cada imagen y representada en la Figura 5.52 en función del ángulo de la plataforma, en donde la serie segmentada indica la distancia media global para los píxeles que almacenan mediciones asociadas por vecino más próximo ($GLT_p < 0$); la serie punteada la distancia media global para los píxeles que almacenan su medición correspondiente ($GLT_p > 0$); y la serie continua la distancia media global de todos los píxeles de la imagen con medición asignada ($GLT_p < 0 + GLT_p > 0$). El mismo procedimiento fue realizado para las imágenes considerando el ángulo de vuelo (movimiento de plataforma *yaw*), tal que la Figura 5.53 representa de manera visual la distancia para cada píxel de la imagen, y la Figura 5.54 la distancia media global para los casos $GLT_p < 0$, $GLT_p > 0$ y promedio total ($GLT_p < 0 + GLT_p > 0$).

El tercer estudio consistió en el análisis del aprovechamiento de los datos originales obtenidos por los sensores al construir la imagen geocorregida. El primer análisis se basa en la representación de imágenes análogas a las matrices L0, en donde cada píxel representa el número de veces que la medición ha sido seleccionada para ser almacenada en diferentes píxeles de la imagen geocorregida correspondiente. La Figura 5.55 muestra dicha representación, de tal manera que cuanto más blanco sea el píxel, más veces ha sido seleccionada la medición para la construcción de la imagen geocorregida.

Píxeles en negro indican que la medición nunca ha sido seleccionada. Posteriormente, el porcentaje global de mediciones originales que no fueron seleccionadas para la construcción de la imagen geocorregida correspondiente fue calculado y representado en la Figura 5.56. Del mismo modo, se calculó el número de veces que una medición ha sido seleccionada para ser almacenada en la imagen geocorregida, siendo este hecho representado en la Figura 5.57, en donde la línea continua corresponde al promedio de veces, y la línea discontinua al máximo de veces. El mismo procedimiento fue realizado para las imágenes considerando el ángulo de vuelo (movimiento de plataforma *yaw*), tal que la Figura 5.58 representa de manera visual el número de veces que cada medición ha sido seleccionada para la construcción de la imagen geocorregida correspondiente; la Figura 5.59 el porcentaje medio global de mediciones originales no seleccionadas; y la Figura 5.60 el número de veces máximo que una misma medición ha sido seleccionada.

Tras aplicar los experimentos a las imágenes correspondientes al *dataset sintético*, se repitieron para las imágenes correspondientes al *dataset real*.

Respecto al primer estudio, en el que se analizaron a cuántos píxeles de la imagen geocorregida final les habían sido realmente asignados su medición correspondiente ($GLT_p > 0$), a cuántos se les habían asignado una medición por vecino más próximo al quedar sin asignar en un principio ($GLT_p < 0$), y cuántos almacenan un espectro a 0 por estar fuera de la escena ($GLT_p = 0$), se representó de manera visual en la Figura 5.61, cada tipo de píxel mediante imágenes análogas a las geocorregidas y aplicando el esquema de color previamente presentado. Posteriormente, para cada imagen se calculó el porcentaje global de cada tipo de píxel, y se comparó con el porcentaje de celdas DMF a *Null* ($DMF_c = Null$) con el porcentaje de celdas DMF con medición asignada ($DMF_c \neq Null$) para las versiones DMF análogas. Todos estos parámetros fueron representados en la Figura 5.63, en donde las barras azules corresponden a las versiones geocorregidas, mientras que las barras rojas corresponden a las versiones DMF. Respecto a las versiones geocorregidas, la barra azul clara representa el porcentaje para el caso $GLT_p = 0$; la barra azul oscura rayada el caso $GLT_p > 0$; la barra azul oscura punteada el caso $GLT_p < 0$; y la barra azul oscura el total de píxeles con medición asignada ($GLT_p < 0 + GLT_p > 0$). Respecto a las versiones en DMF, la barra roja clara representa el porcentaje de celdas DMF a *Null* ($DMF_c = Null$), mientras que la barra roja oscura representa el porcentaje de celda con medición asignada ($DMF_c \neq Null$). El mismo procedimiento fue realizado para las imágenes considerando el ángulo de vuelo, y para la versión DMF, también el tamaño de celda DMF calculado de manera automática. La Figura 5.62 muestra las imágenes coloreadas de acuerdo a si se tratan de píxeles almacenando su medición original correspondiente (verde), o una medición asignada por vecino más próximo (rojo). La Figura 5.50 muestra los porcentajes de cada tipo de píxel de acuerdo al tipo de asignación de medición ($GLT_p = 0$, $GLT_p < 0$ y $GLT_p > 0$ para el caso de la versión geocorregida, y $DMF_c = Null$ y $DMF_c \neq Null$ para la versión DMF).

En cuanto al segundo estudio, en el que se realizó un análisis de los desplazamientos de las mediciones introducidos por el proceso de geocorrección respecto a las coordenadas originales de cada medición, se representaron, de manera visual en la Figura 5.65, las distancias de la medición asociada a cada centro de píxel respecto a su localización original (proporcionada por el sistema de posicionamiento), de tal manera que cuanto más blanco es el píxel, mayor es la distancia. Posteriormente, para cada imagen se calculó la distancia media global y se representaron en la Figura 5.66. El mismo procedimiento fue realizado para las imágenes considerando el ángulo de vuelo, tal que la Figura 5.67 representa la distancia de la medición asignada a cada píxel respecto a su localización original (proporcionada por el sistema de posicionamiento); y tal que la Figura 5.68 representa las distancias medias globales.

Respecto al tercer estudio, en el que se realizó un análisis del porcentaje de mediciones originales utilizadas para construir la imagen geocorregida, se representó, de manera visual en la Figura 5.69, análoga a las matrices L0 e IGM, el número de veces que cada medición ha sido seleccionada para dicho propósito, tal que cuanto más blanco sea el píxel, más veces ha sido seleccionada la medición correspondiente. Tras este análisis, se determinaron los porcentajes globales de mediciones originales que no fueron seleccionadas para la construcción de las imágenes geocorregidas correspondiente,

las cuales se muestran en la Figura 5.70; así como el número de veces que cada medición ha sido seleccionada para ser almacenada en un determinado píxel de la imagen geocorregida, representado en la Figura 5.71, en donde la barra azul clara corresponde al promedio global, y la barra azul oscura punteada al valor máximo. El mismo procedimiento fue realizado para las imágenes considerando el ángulo de vuelo, tal que la Figura 5.72 representa el número de veces que la medición ha sido seleccionada y almacenada en la imagen geocorregida correspondiente; la Figura 5.73 representa el porcentaje global de mediciones originales que no fueron seleccionadas; y la Figura 5.74 representa la media global y el máximo de veces que una medición ha sido seleccionada.

Finalmente, para el cuarto estudio, el cual tenía como objetivo analizar cómo los errores espaciales introducidos en el proceso de geocorrección afectan a las imágenes resultantes mediante un caso real, se aplicó la metodología expuesta en la sección anterior, y se obtuvieron dos imágenes como resultado. La primera de ellas, representada en la Figura 5.75(a), es el ángulo espectral (α) obtenido para cada píxel de la versión geocorregida, respecto a su análogo en la imagen *ground-truth*. Cuanto más blanco es el píxel, mayor es el ángulo espectral (α) entre ambos y por tanto mayor es la diferencia entre los espectros. De la misma manera, la segunda imagen corresponde al ángulo espectral (α) para todas las mediciones almacenadas en la representación DMF, también respecto a sus análogos en la imagen *ground-truth*, siendo representado en la Figura 5.75(b). En este caso en concreto, al ser el resultado una imagen en formato DMF, se muestra el ángulo espectral (α) de la medición más cercana al centro de cada celda DMF.

El ángulo espectral medio ($\bar{\alpha}$) obtenido para las versiones geocorregidas y DMF, así como el número de mediciones consideradas en la comparación, se muestran en la Tabla 5.12.

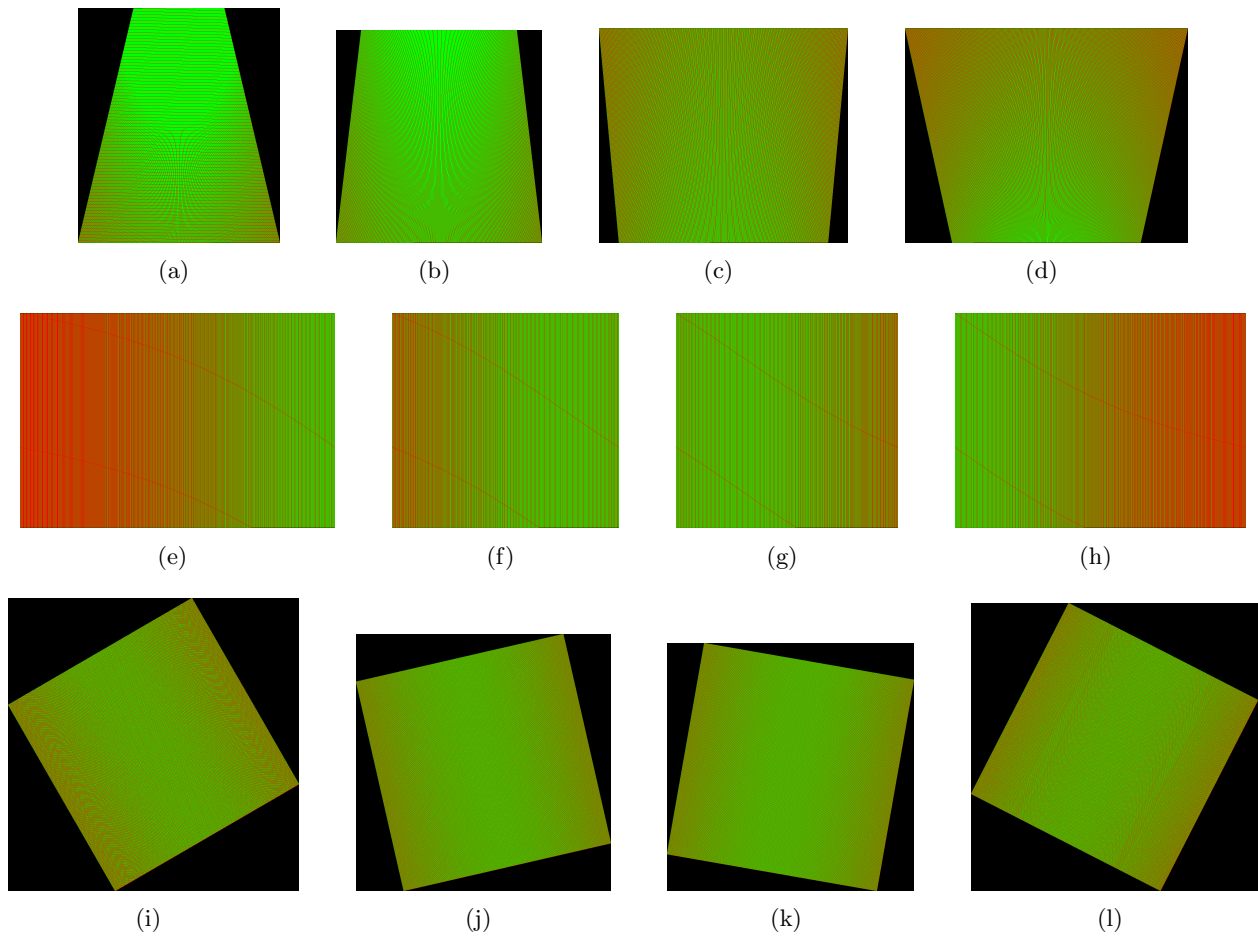


Figura 5.47: Visualización de la localización de los píxeles de las imágenes geocorregidas de acuerdo a la manera en la fueron asignadas sus mediciones correspondientes (dataset sintético).
 (a) Pitch -30° . (b) Pitch -12° . (c) Pitch 12° . (d) Pitch 30° . (e) Roll -30° . (f) Roll -12° . (g) Roll 12° . (h) Roll 30° . (i) Yaw -30° . (j) Yaw -12° . (k) Yaw 12° . (l) Yaw 30° .

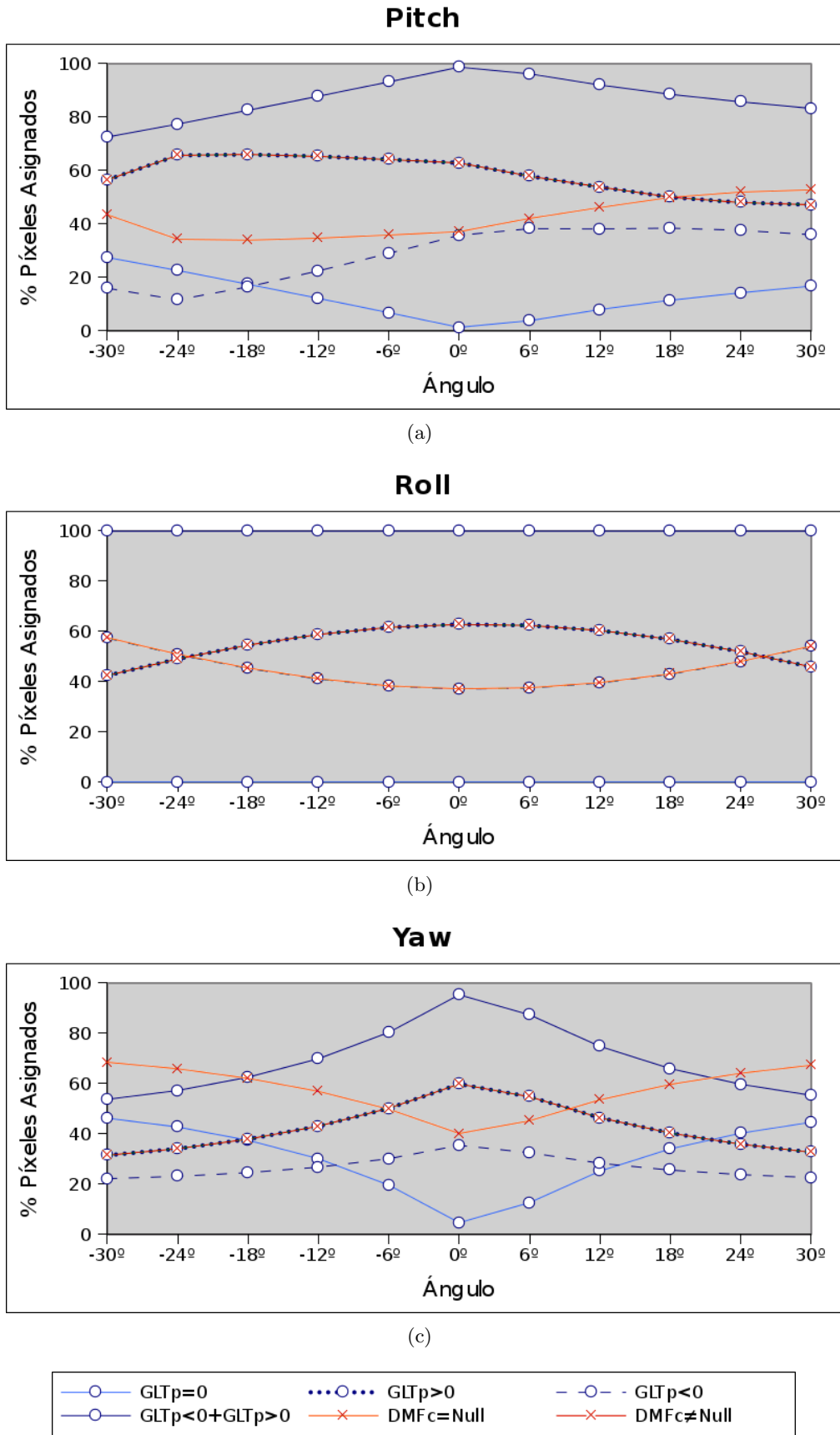


Figura 5.48: Porcentaje de tipo de asignación de mediciones presente en las imágenes geocorregidas y en sus versiones DMF análogas (dataset sintético).
 (a) Pitch. (b) Roll. (c) Yaw.

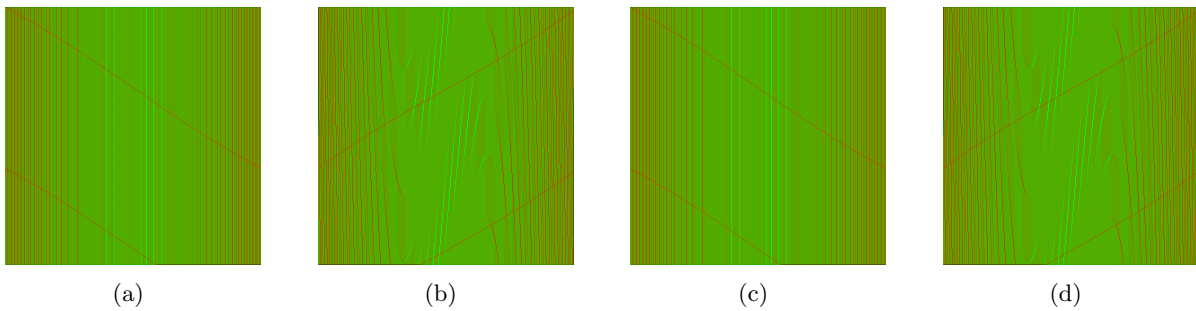


Figura 5.49: Visualización de la localización de los píxeles de las imágenes geocorregidas de acuerdo a la manera en la fueron asignadas sus mediciones correspondientes, teniendo en cuenta el ángulo de vuelo y el tamaño de celda DMF calculado de manera automática (movimiento de plataforma yaw).

(a) Yaw -30° . (b) Yaw -12° . (c) Yaw 12° . (d) Yaw 30° .

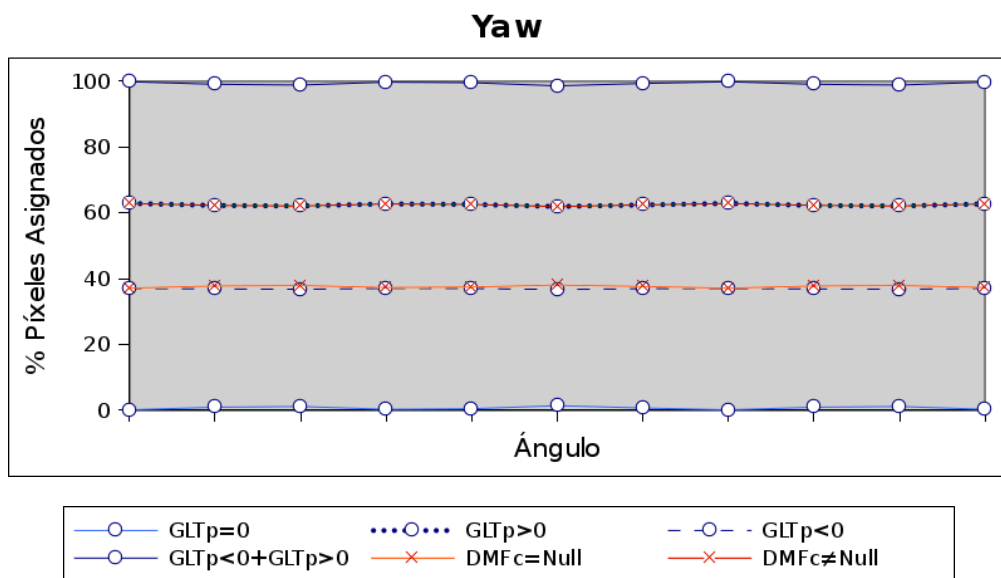


Figura 5.50: Porcentaje de tipo de asignación de mediciones presente en las imágenes geocorregidas y en sus versiones DMF análogas (dataset sintético), teniendo en cuenta el ángulo de vuelo y el tamaño de celda DMF calculado de manera automática (movimiento de plataforma yaw).

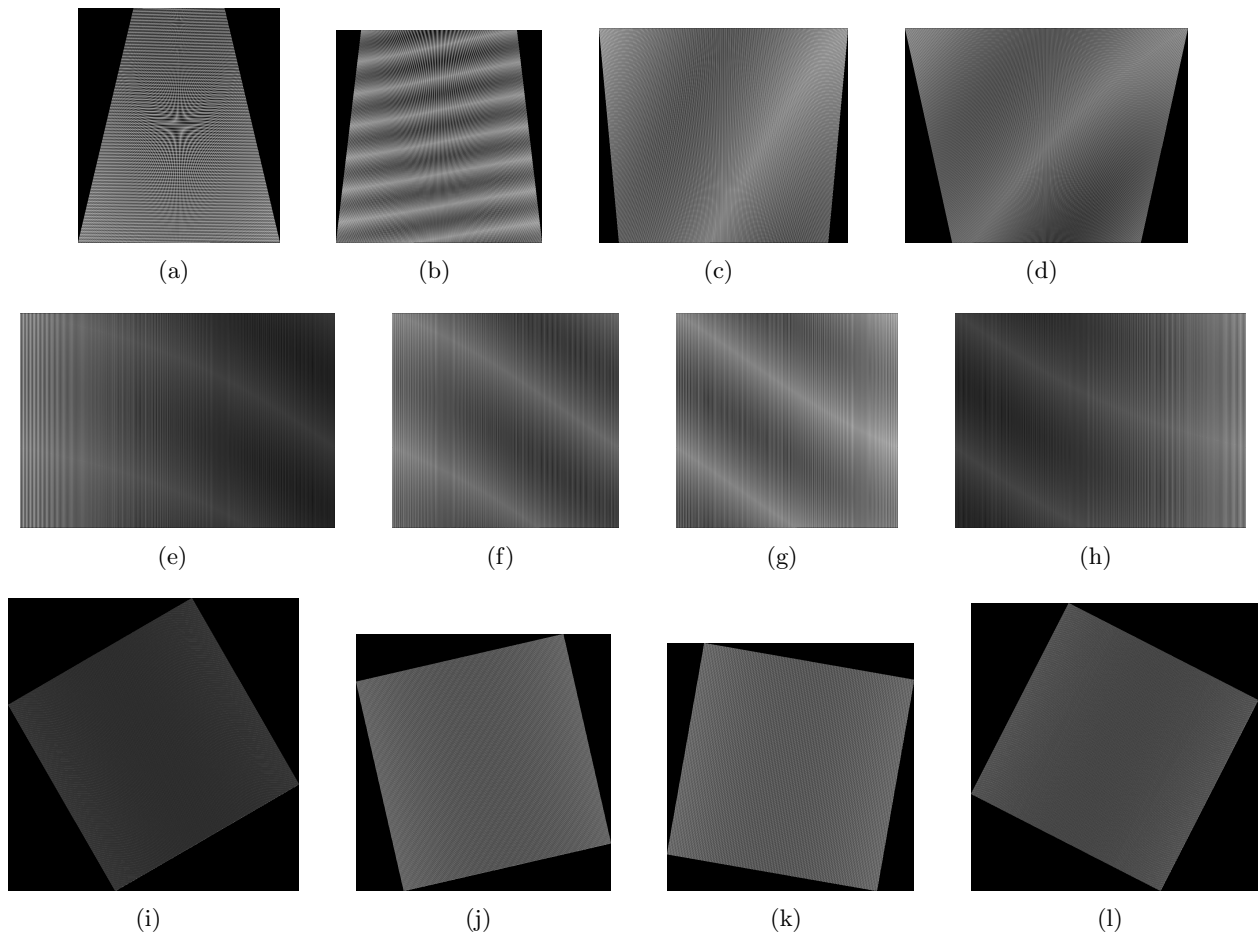


Figura 5.51: Visualización de las distancias entre el posicionamiento geocorregido de cada medición y el proporcionado por el sistema de posicionamiento (dataset sintético).
 (a) Pitch -30° . (b) Pitch -12° . (c) Pitch 12° . (d) Pitch 30° . (e) Roll -30° . (f) Roll -12° . (g) Roll 12° . (h) Roll 30° . (i) Yaw -30° . (j) Yaw -12° . (k) Yaw 12° . (l) Yaw 30° .

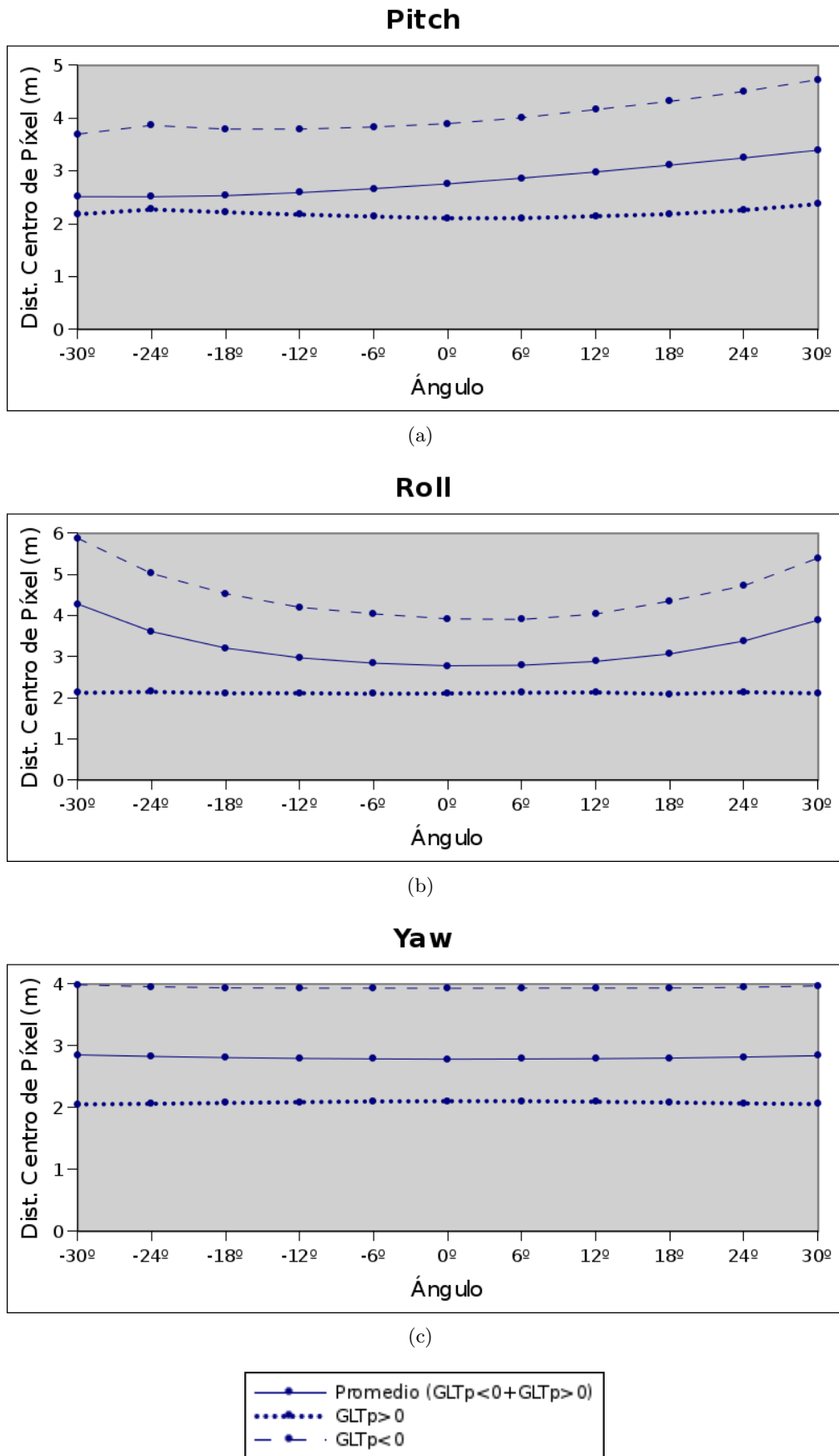


Figura 5.52: Distancias medias globales de las posiciones geocorregidas para cada píxel respecto a la posición original de la medición correspondiente (dataset sintético).

(a) Pitch. (b) Roll. (c) Yaw.

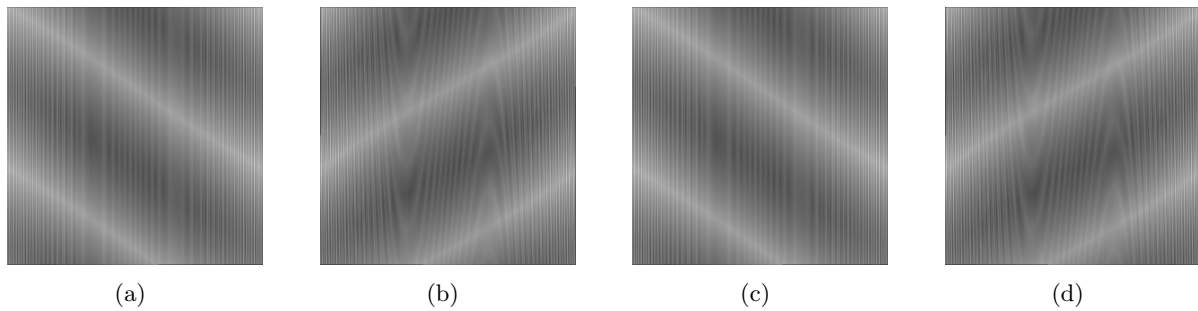


Figura 5.53: Visualización de las distancias entre el posicionamiento geocorregido de cada medición y el proporcionado por el sistema de posicionamiento (dataset sintético), teniendo en cuenta el ángulo de vuelo (movimiento de plataforma yaw).

(a) Yaw -30° . (b) Yaw -12° . (c) Yaw 12° . (d) Yaw 30° .

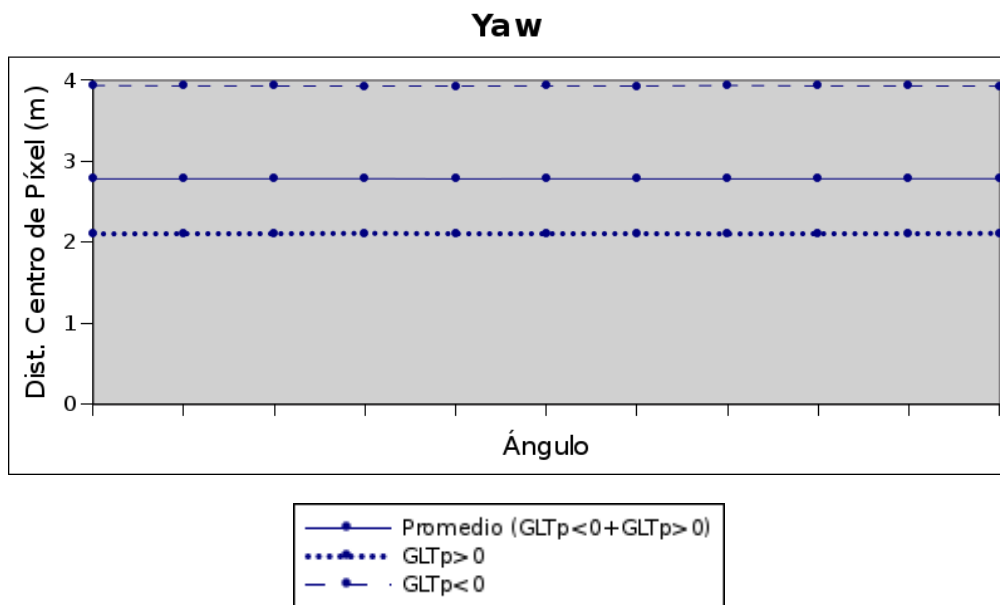


Figura 5.54: Distancias medias globales de las posiciones geocorregidas para cada píxel respecto a la posición original de la medición correspondiente (dataset sintético), teniendo en cuenta el ángulo de vuelo (movimiento de plataforma yaw).

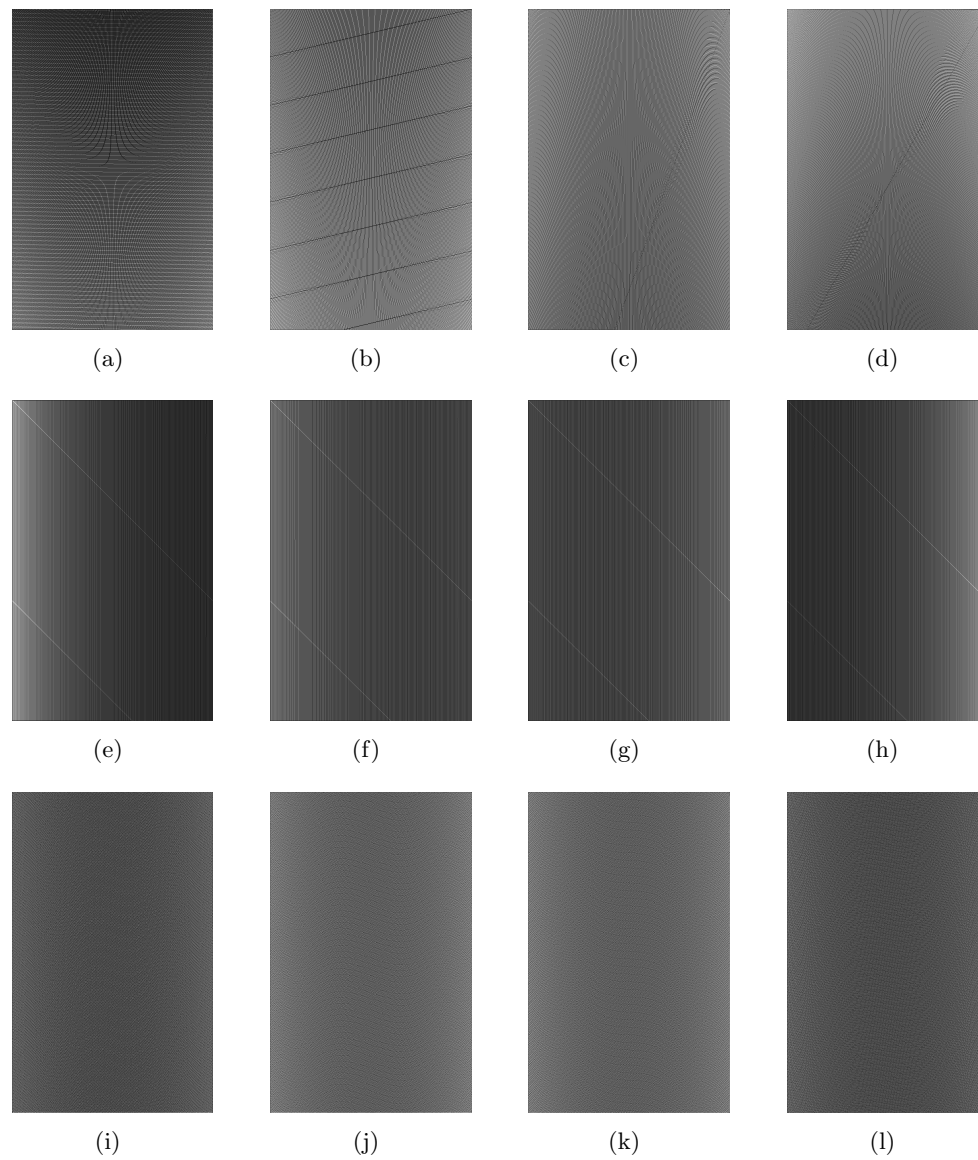


Figura 5.55: Visualización de la frecuencia de selección de las mediciones originales para la construcción de la imagen geocorregida correspondiente (dataset sintético).
 (a) Pitch -30° . (b) Pitch -12° . (c) Pitch 12° . (d) Pitch 30° . (e) Roll -30° . (f) Roll -12° . (g) Roll 12° . (h) Roll 30° . (i) Yaw -30° . (j) Yaw -12° . (k) Yaw 12° . (l) Yaw 30° .

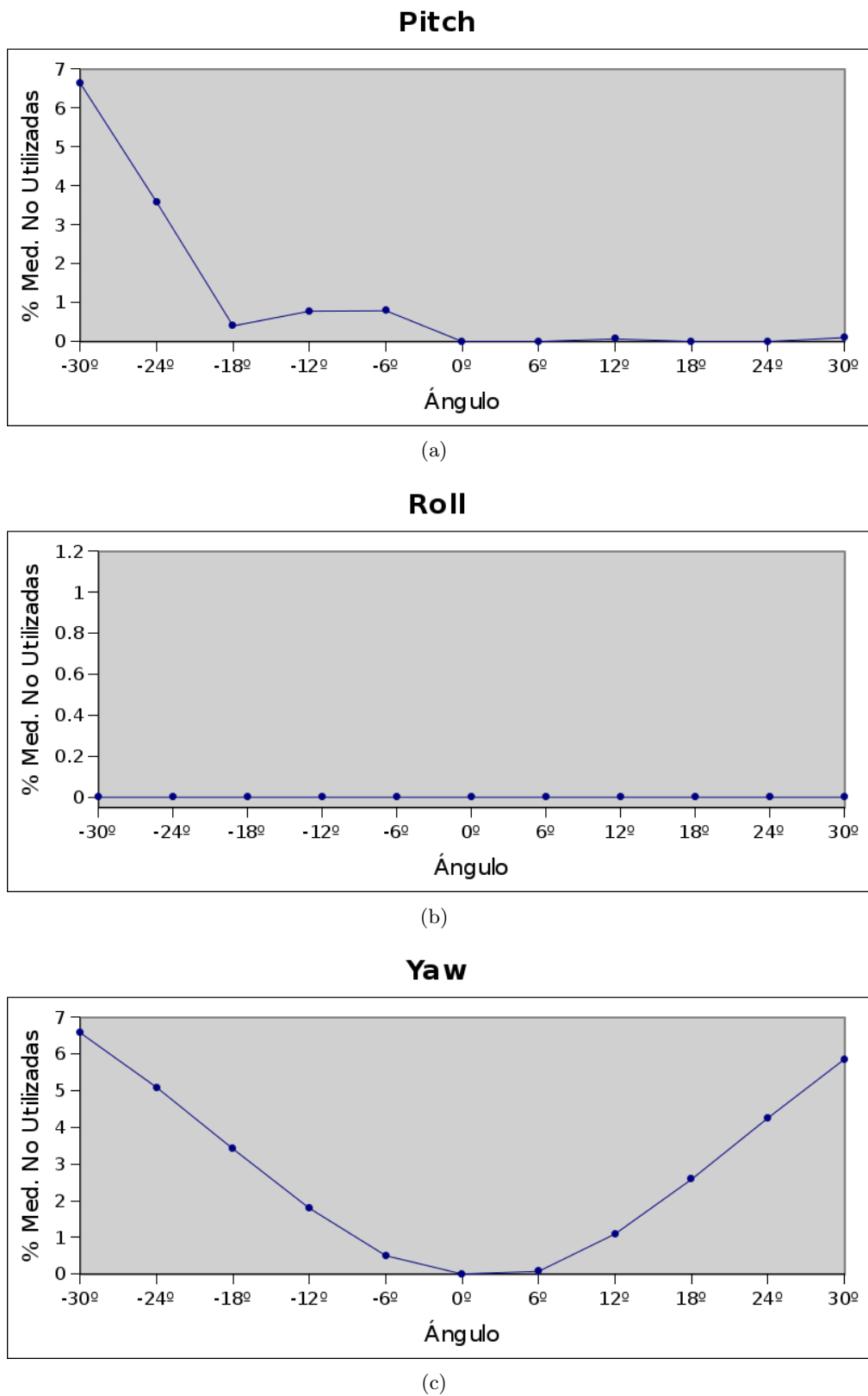
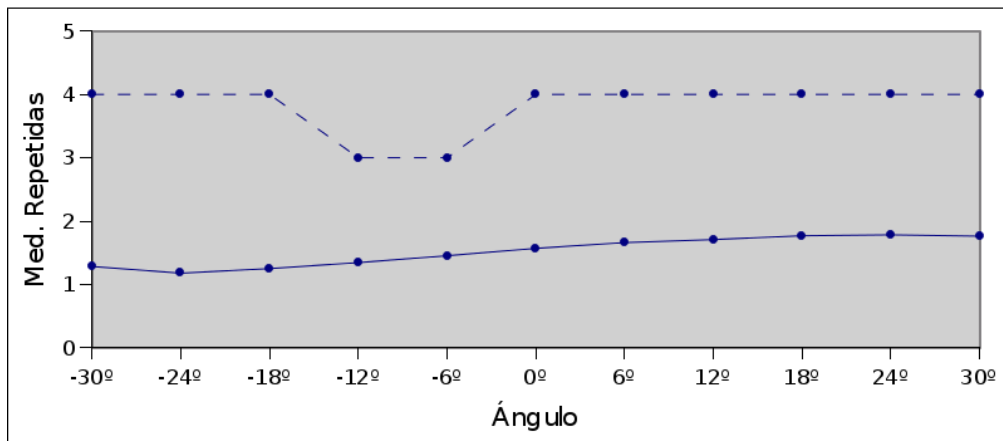
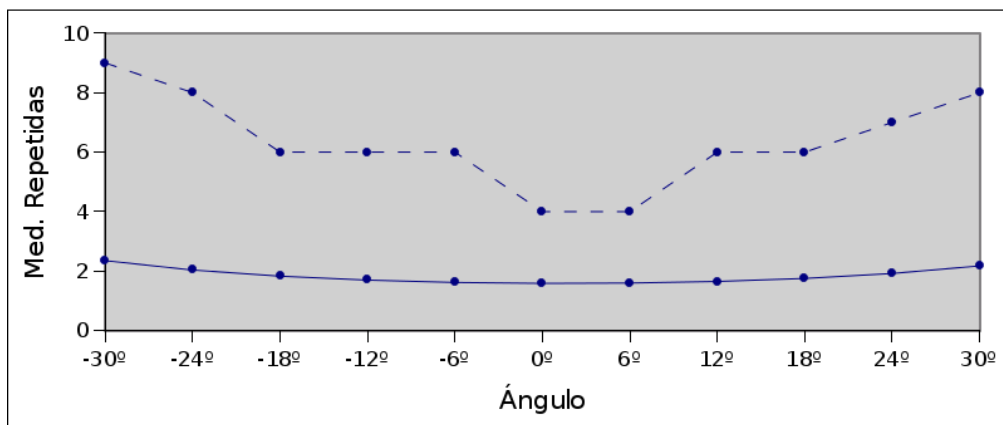


Figura 5.56: Porcentaje de mediciones originales no seleccionadas para la construcción de la imagen geocorregida correspondiente (dataset sintético).

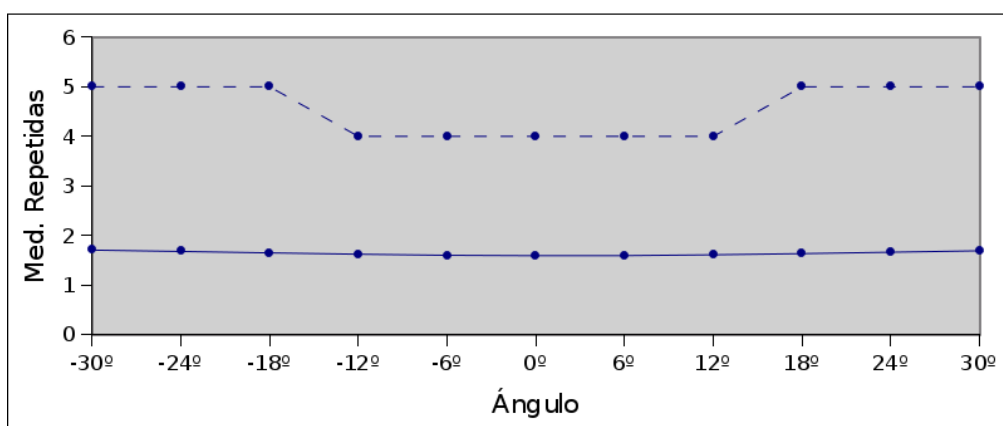
(a) Pitch. (b) Roll. (c) Yaw.

Pitch

(a)

Roll

(b)

Yaw

(c)

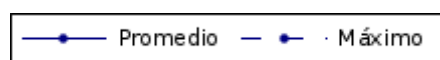


Figura 5.57: Número de veces que una medición ha sido seleccionada para para la construcción de la imagen geocorregida correspondiente (dataset sintético).
(a) Pitch. (b) Roll. (c) Yaw.

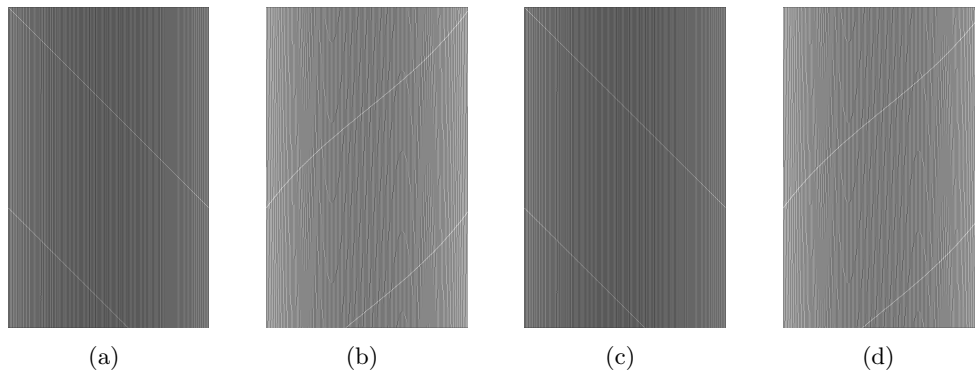


Figura 5.58: Visualización de la frecuencia de selección de las mediciones originales para la construcción de la imagen geocorregida correspondiente (dataset sintético), teniendo en cuenta el ángulo de vuelo (movimiento de plataforma yaw).
 (a) Yaw -30° . (b) Yaw -12° . (c) Yaw 12° . (d) Yaw 30° .

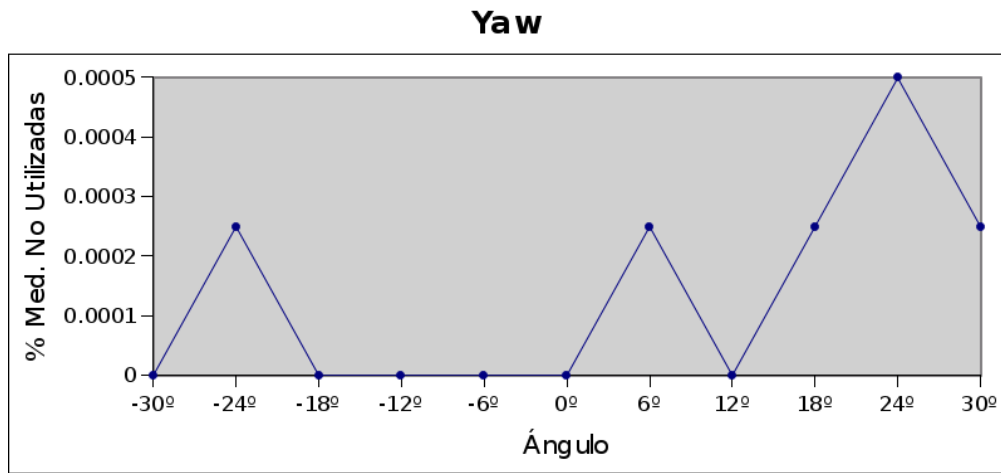


Figura 5.59: Porcentaje de mediciones originales no seleccionadas para la construcción de la imagen geocorregida correspondiente (dataset sintético), teniendo en cuenta el ángulo de vuelo (movimiento de plataforma yaw).

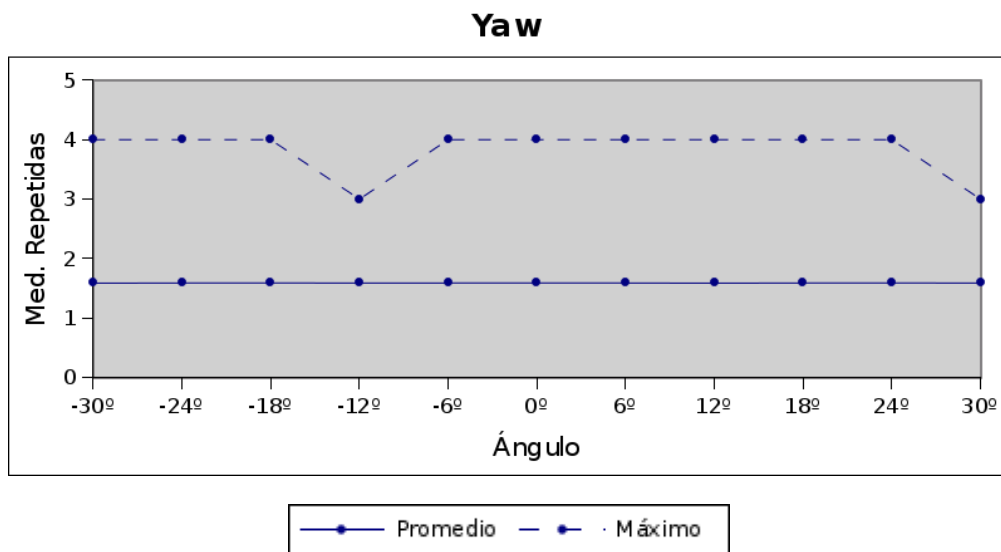
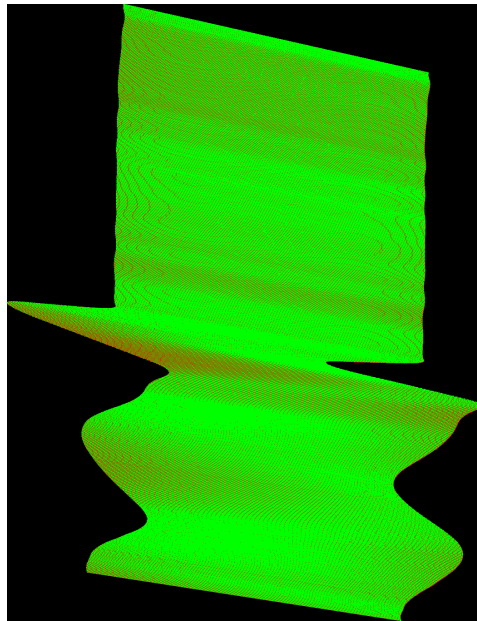
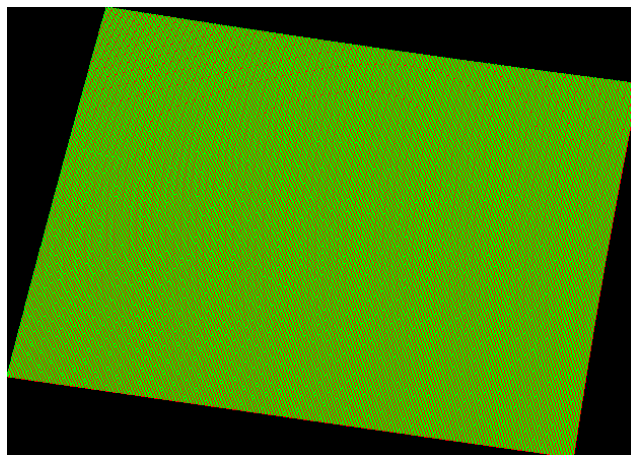


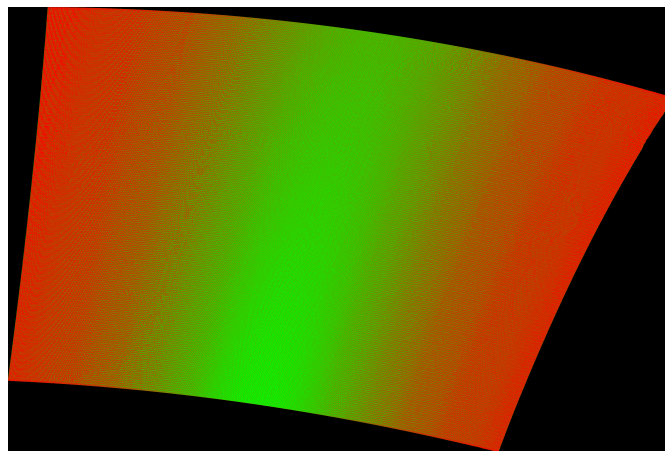
Figura 5.60: Número de veces que una medición ha sido seleccionada para para la construcción de la imagen geocorregida correspondiente (dataset sintético), teniendo en cuenta el ángulo de vuelo (movimiento de plataforma yaw).



(a)

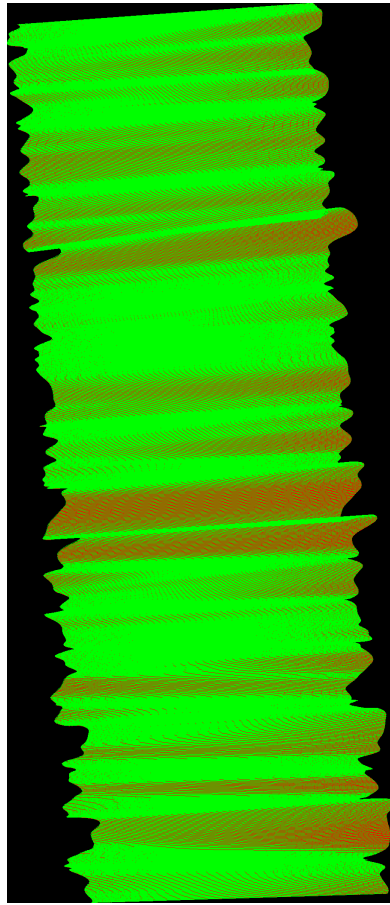


(b)

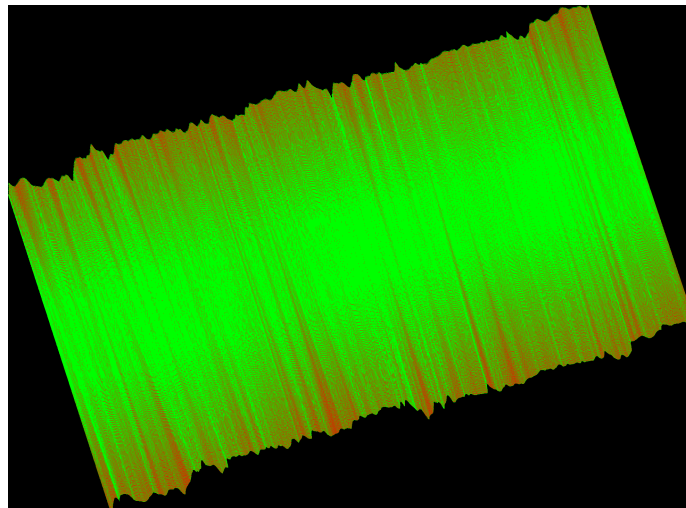


(c)

Figura 5.61: Visualización de la localización de los píxeles de las imágenes geocorregidas de acuerdo a la manera en la fueron asignadas sus mediciones correspondientes (dataset real).
(a) Escena Cuprite. (b) Escena Barrax. (c) Escena Mediterráneo. (g) Escena GMéxico. (h) Escena Cantera.

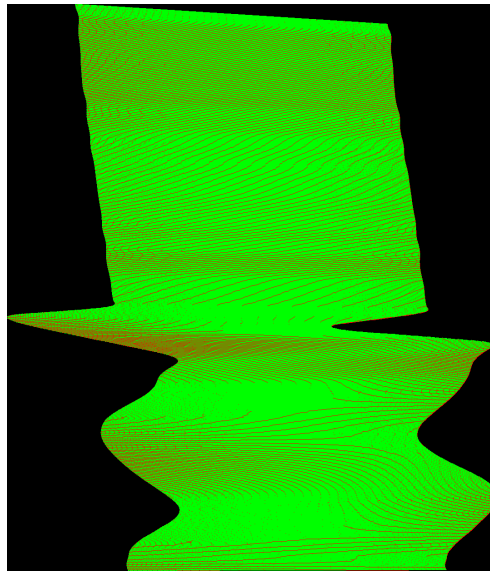


(g)

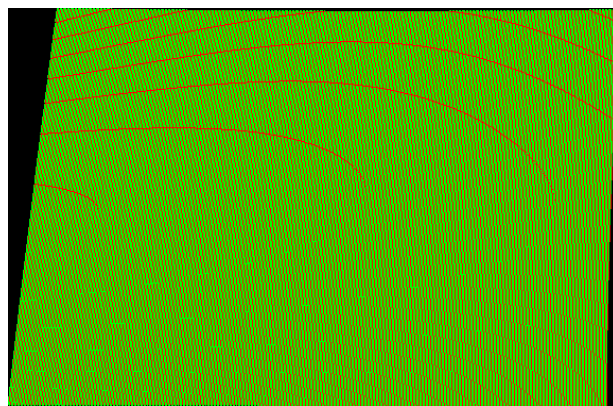


(h)

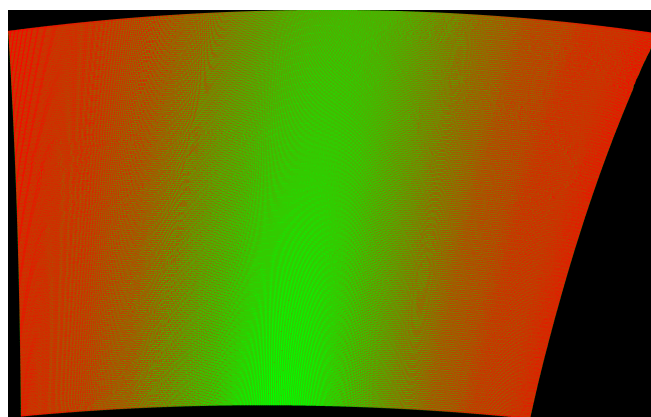
Figura 5.61: (Cont.)



(a)



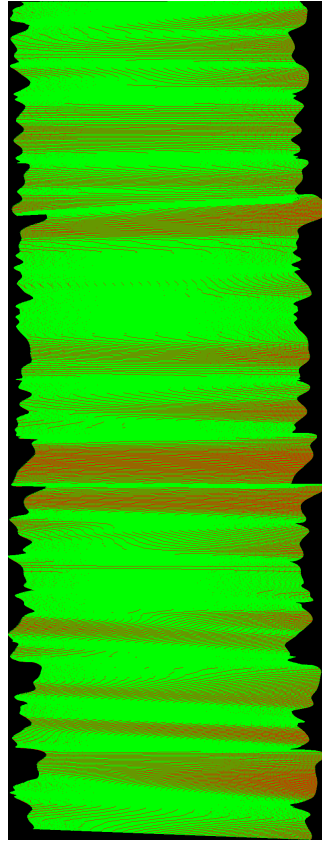
(b)



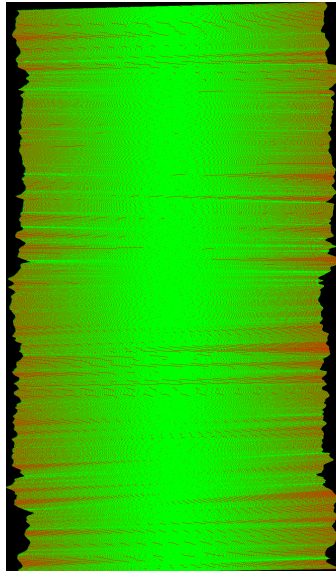
(c)

Figura 5.62: Porcentaje de tipo de asignación de mediciones presente en las imágenes geocorregidas y en sus versiones DMF análogas (dataset real), teniendo en cuenta el ángulo de vuelo y el tamaño de celda DMF calculado de manera automática.

(a) Escena Cuprite. (b) Escena Barrax. (c) Escena Mediterráneo. (g) Escena GMéxico. (h) Escena Cantera.



(g)



(h)

Figura 5.62: (Cont.)

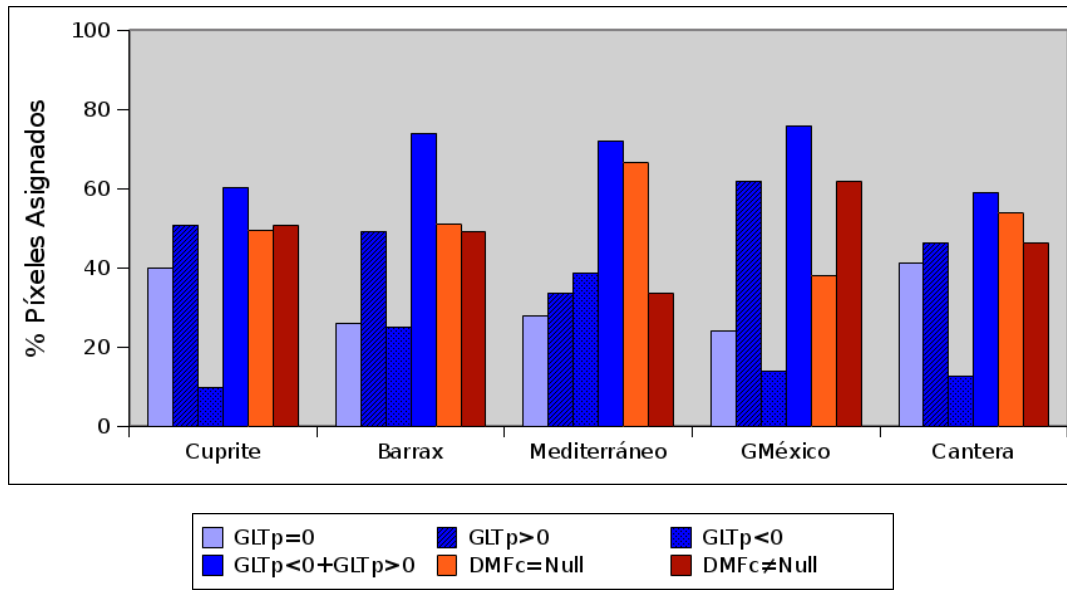


Figura 5.63: Porcentaje de tipo de asignación de mediciones presente en las imágenes geocorregidas y en sus versiones DMF análogas (dataset real).

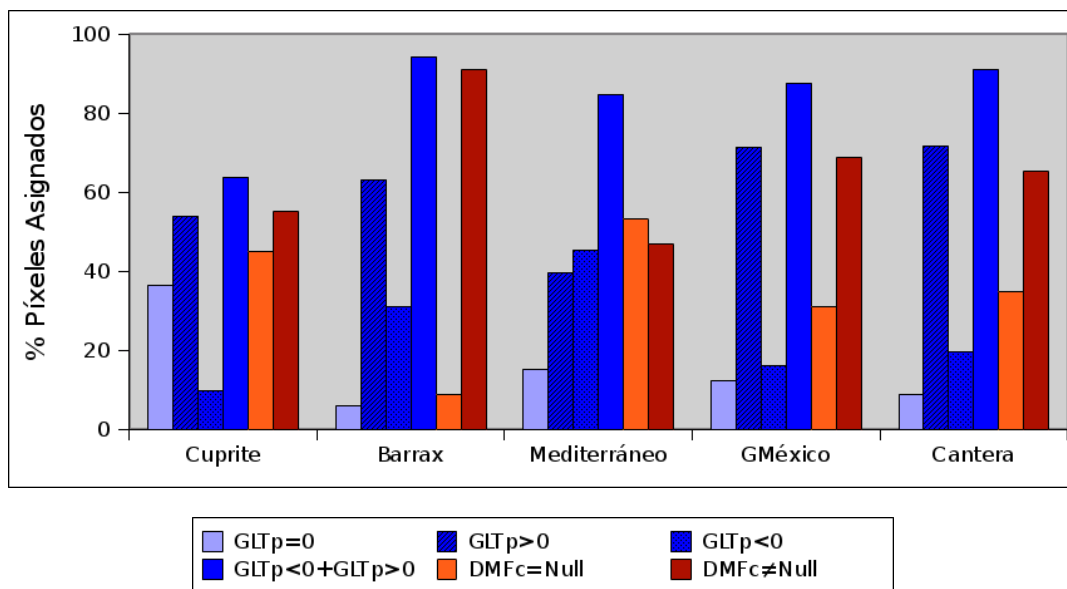
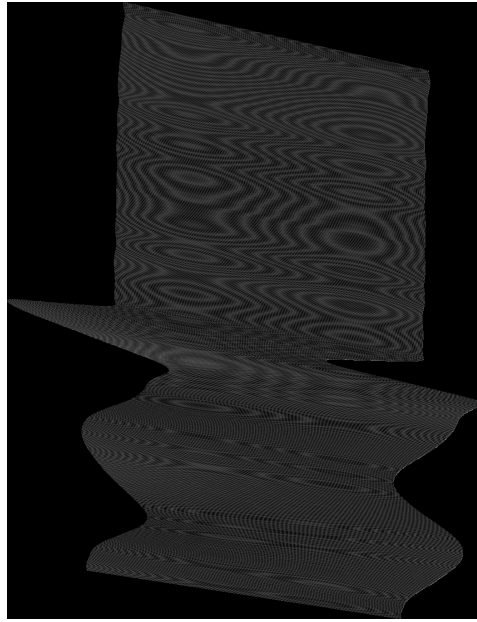
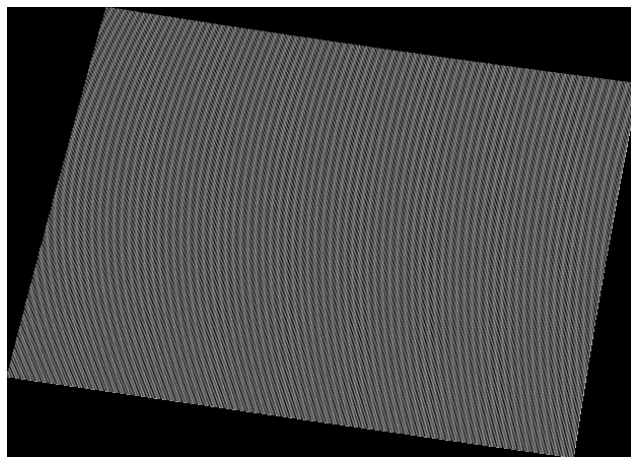


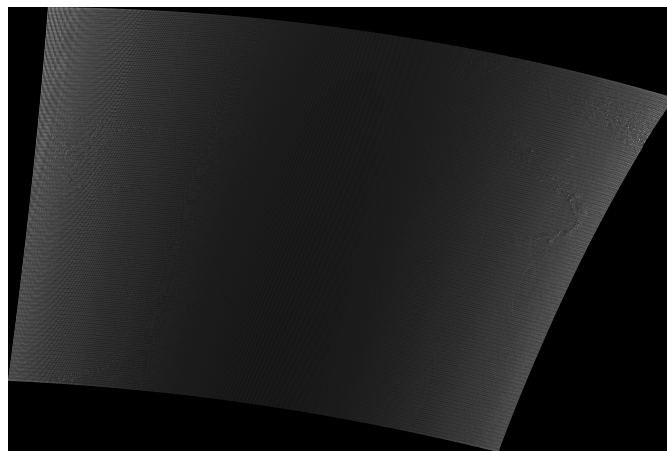
Figura 5.64: Porcentaje de tipo de asignación de mediciones presente en las imágenes geocorregidas y en sus versiones DMF análogas (dataset real), teniendo en cuenta el ángulo de vuelo y el tamaño de celda DMF calculado de manera automática.



(a)

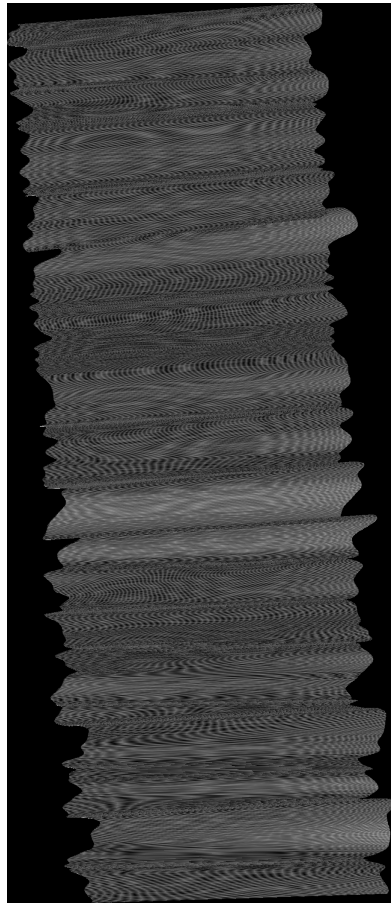


(b)

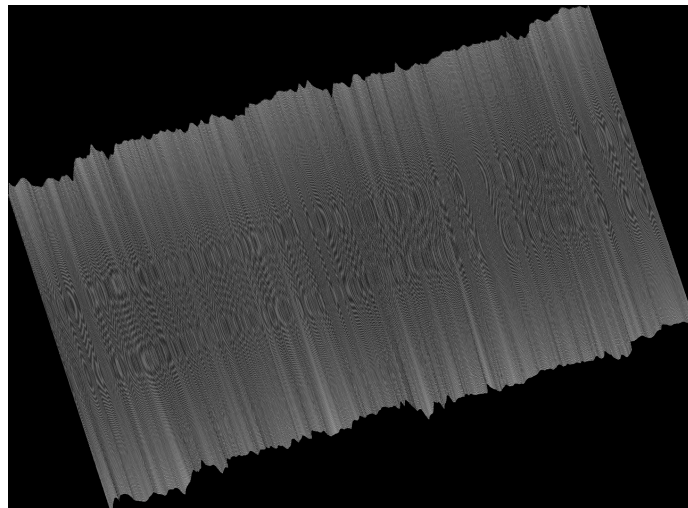


(c)

Figura 5.65: Visualización de la localización de los píxeles de las imágenes geocorregidas de acuerdo a la manera en la fueron asignadas sus mediciones correspondientes (dataset real).
(a) Escena Cuprite. (b) Escena Barrax. (c) Escena Mediterráneo. (g) Escena GMéxico. (h) Escena Cantera.



(g)



(h)

Figura 5.65: (Cont.)

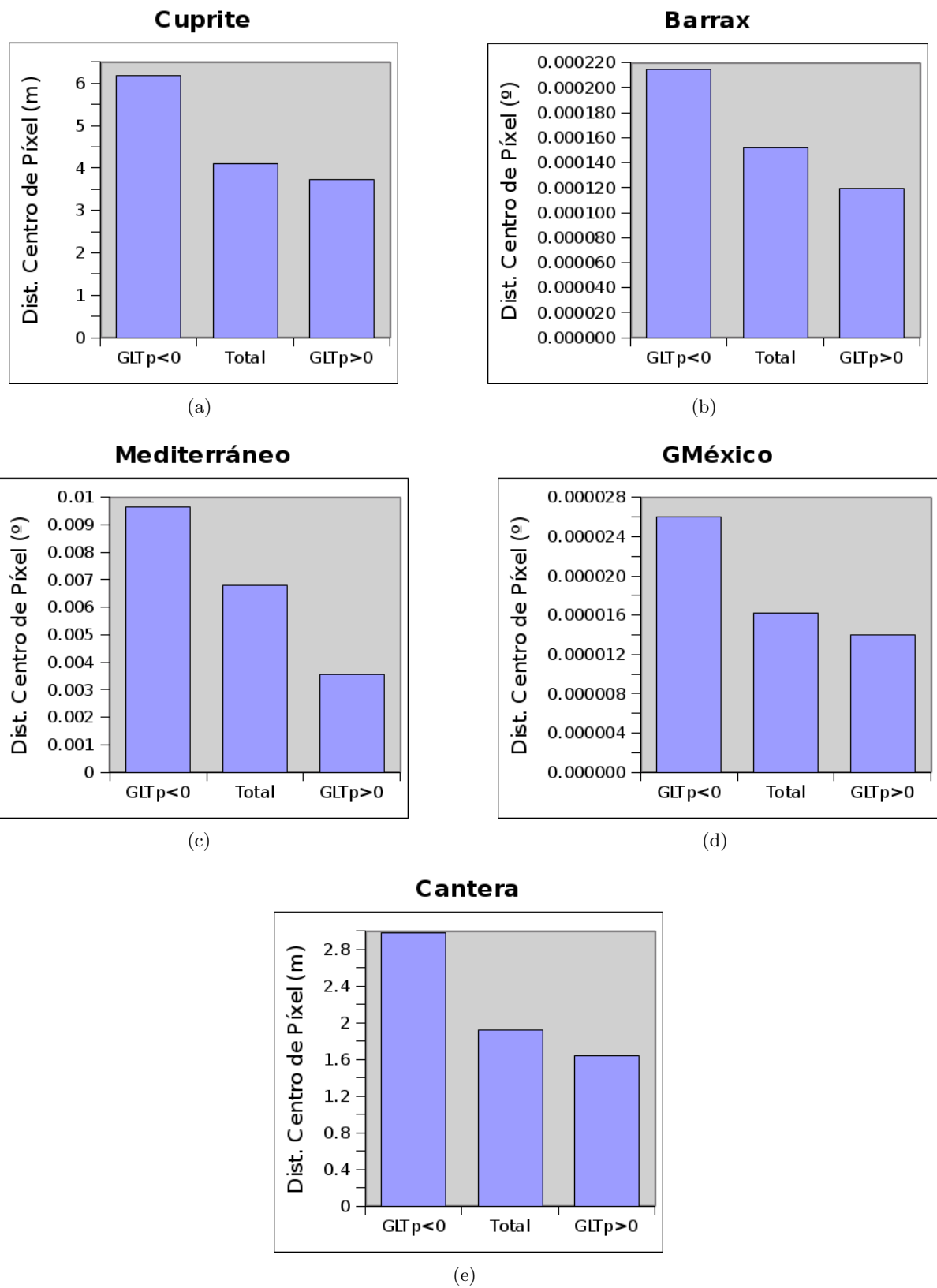
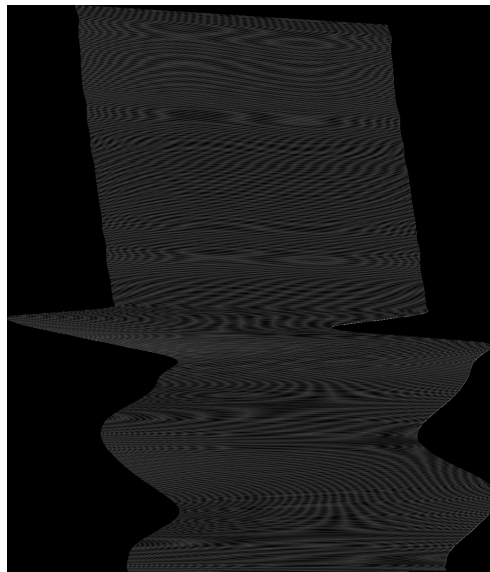
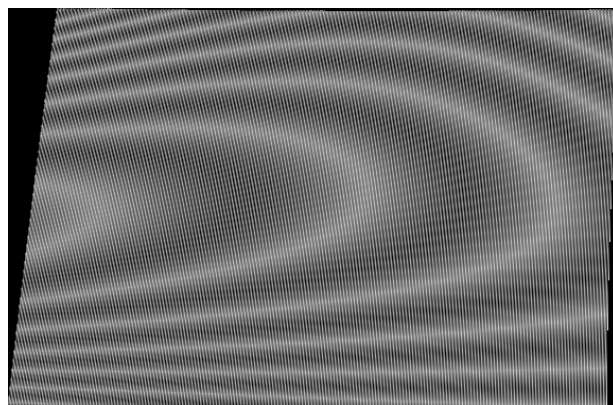


Figura 5.66: Distancias medias de las posiciones geocorregidas para cada píxel respecto a la posición original de la medición correspondiente (dataset real).

(a) Escena Cuprite. (b) Escena Barrax. (c) Escena Mediterráneo. (d) Escena GMéxico. (e) Escena Cantera.



(a)

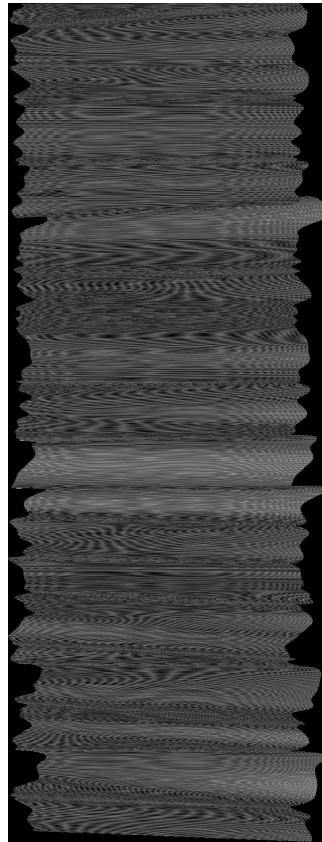


(b)

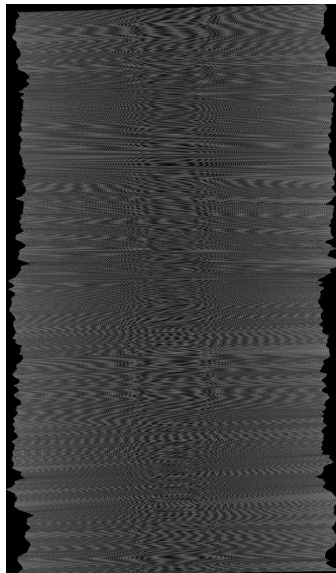


(c)

Figura 5.67: Visualización de las distancias entre el posicionamiento geocorregido de cada medición y el proporcionado por el sistema de posicionamiento (dataset real), teniendo en cuenta el ángulo de vuelo. (a) Escena Cuprite. (b) Escena Barrax. (c) Escena Mediterráneo. (g) Escena GMéxico. (h) Escena Cantera.



(g)



(h)

Figura 5.67: (Cont.)

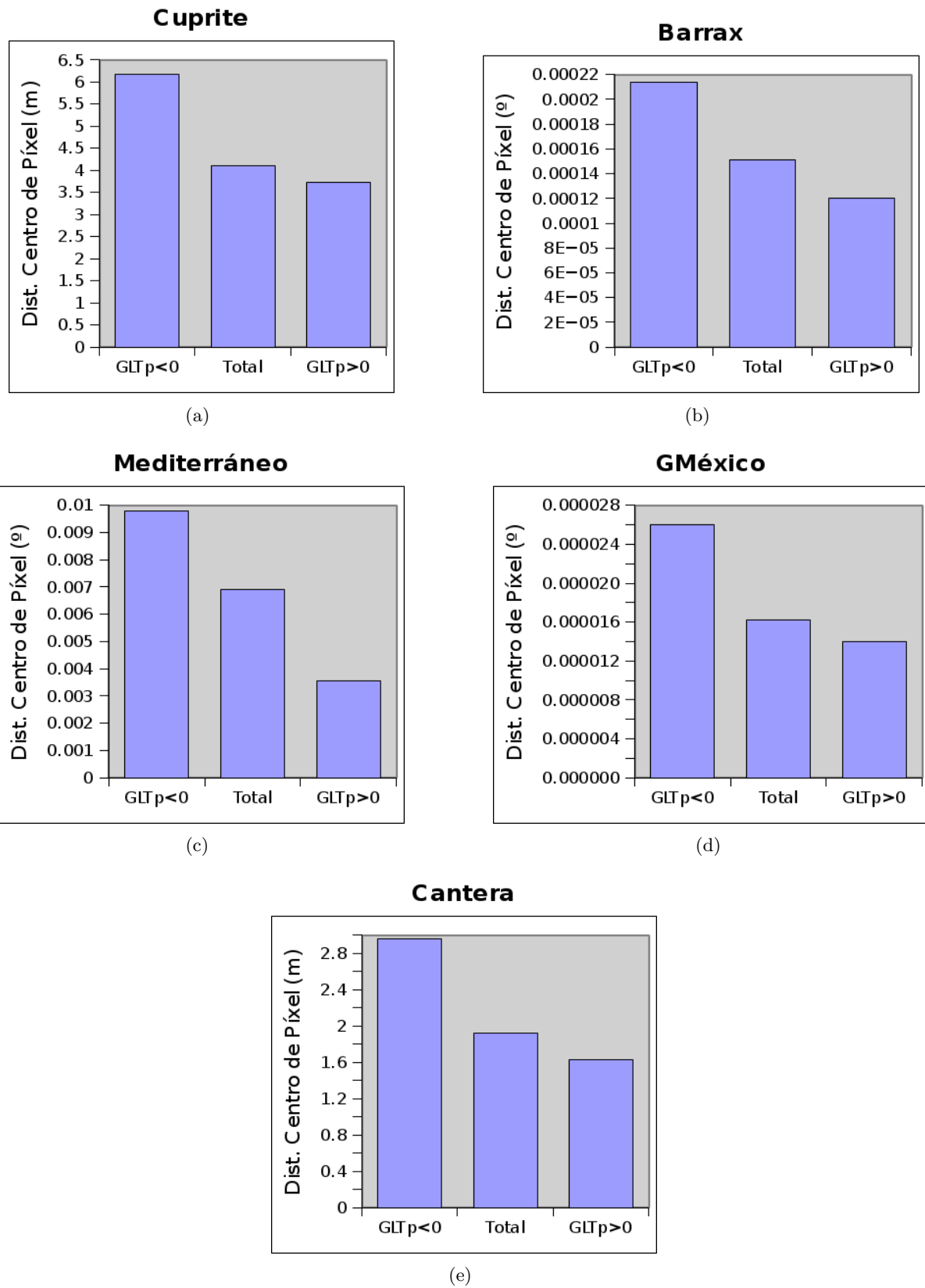
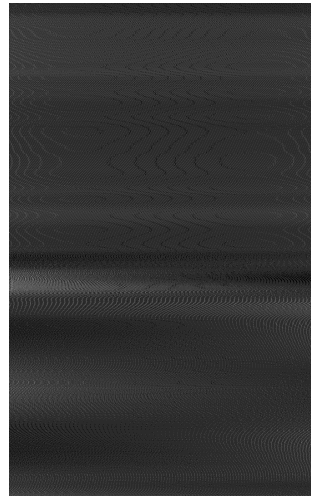
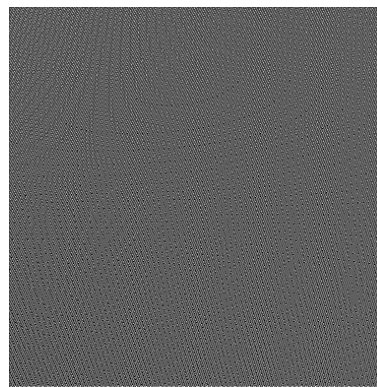


Figura 5.68: Distancias medias de las posiciones geocorregidas para cada píxel respecto a la posición original de la medición correspondiente (dataset real), teniendo en cuenta el ángulo de vuelo.

(a) Escena Cuprite. (b) Escena Barrax. (c) Escena Mediterráneo. (d) Escena GMéxico. (e) Escena Cantera.



(a)



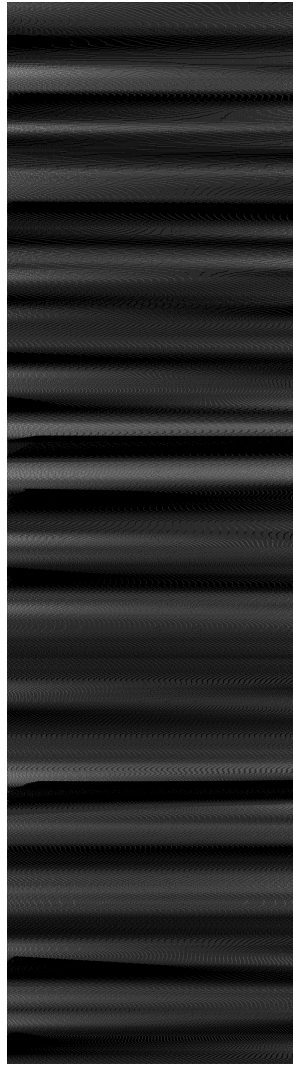
(b)



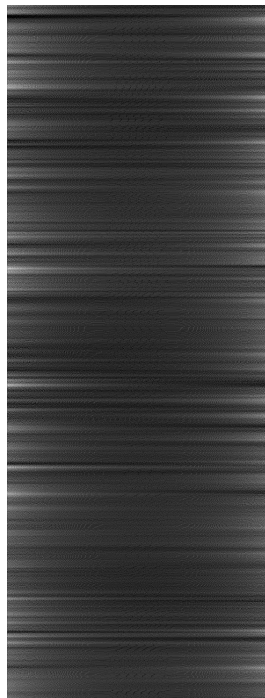
(c)

Figura 5.69: Visualización de la frecuencia de selección de las mediciones originales para la construcción de la imagen geocorregida correspondiente (dataset real).

(a) Escena Cuprite. (b) Escena Barrax. (c) Escena Mediterráneo. (g) Escena GMéxico. (h) Escena Cantera.



(g)



(h)

Figura 5.69: (Cont.)

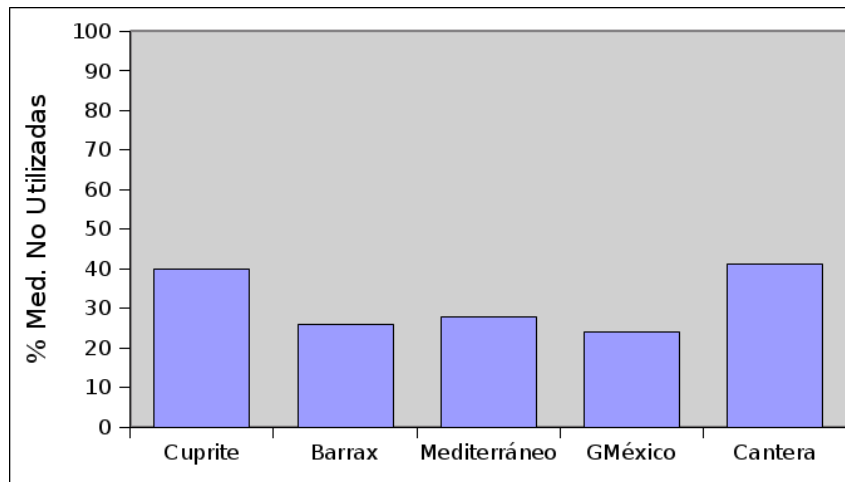


Figura 5.70: *Porcentaje de mediciones originales no seleccionadas para la construcción de la imagen geocorregida correspondiente (dataset real).*

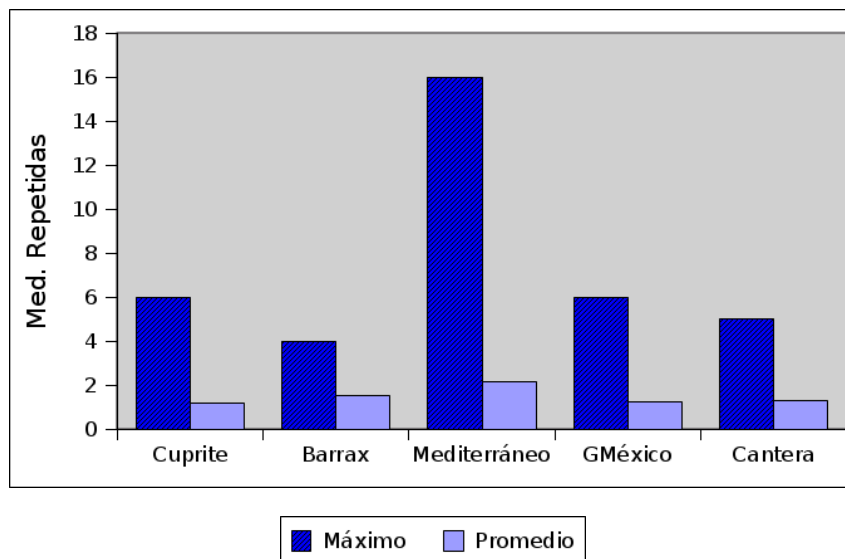
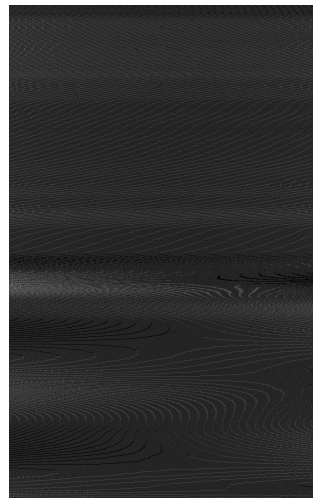
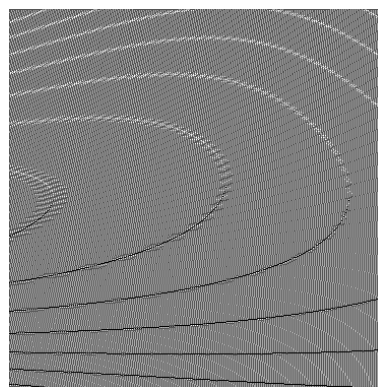


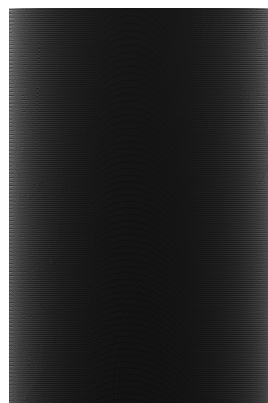
Figura 5.71: *Número de veces que una medición ha sido seleccionada para para la construcción de la imagen geocorregida correspondiente (dataset real).*



(a)



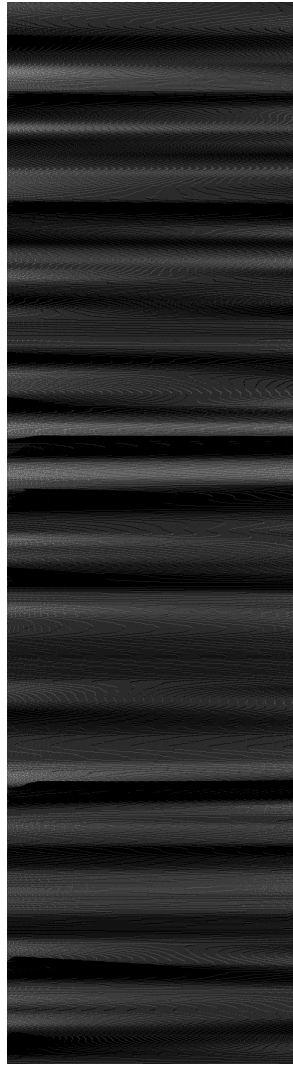
(b)



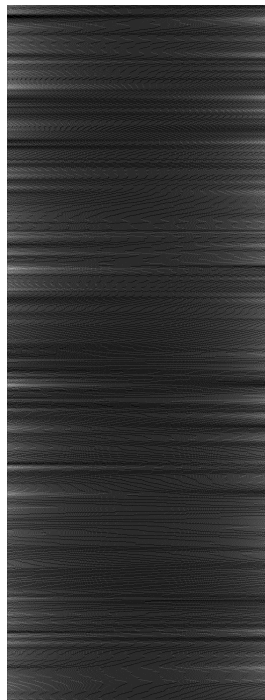
(c)

Figura 5.72: Visualización de la frecuencia de selección de las mediciones originales para la construcción de la imagen geocorregida correspondiente (dataset real), teniendo en cuenta el ángulo de vuelo.

(a) Escena Cuprite. (b) Escena Barrax. (c) Escena Mediterráneo. (g) Escena GMéxico. (h) Escena Cantera.



(g)



(h)

Figura 5.72: (Cont.)

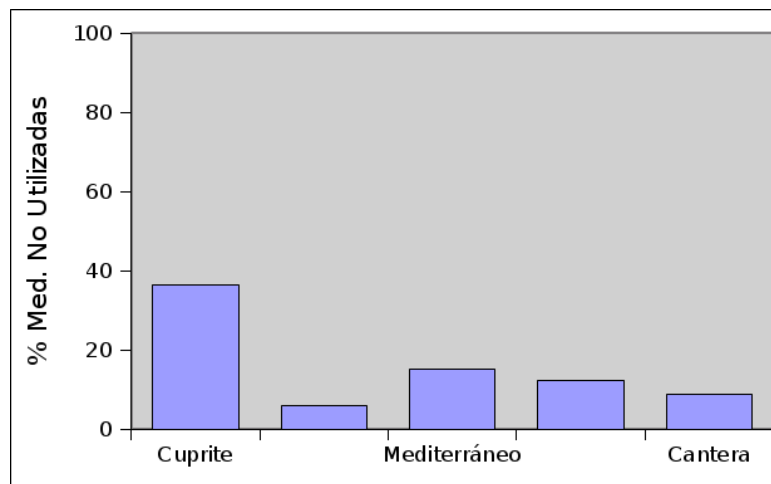


Figura 5.73: Porcentaje de mediciones originales no seleccionadas para la construcción de la imagen geocorregida correspondiente (dataset real), teniendo en cuenta el ángulo de vuelo.

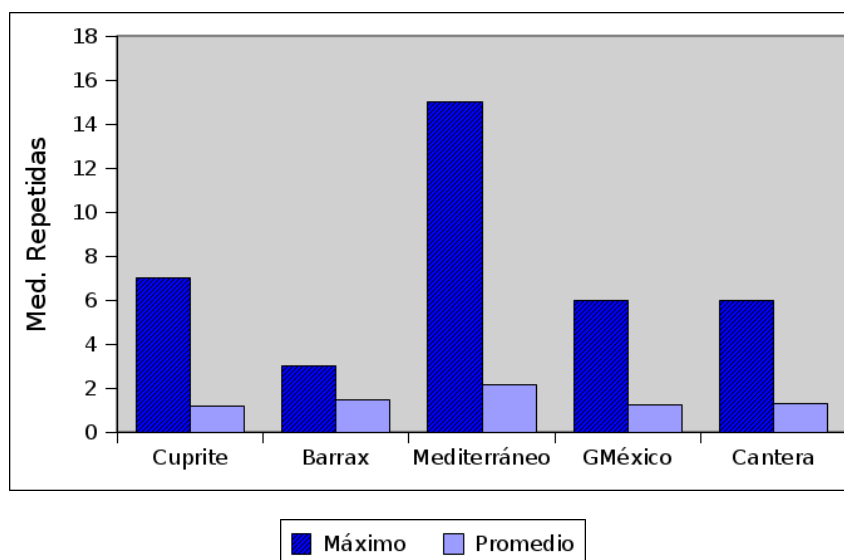


Figura 5.74: Número de veces que una medición ha sido seleccionada para para la construcción de la imagen geocorregida correspondiente (dataset real), teniendo en cuenta el ángulo de vuelo.

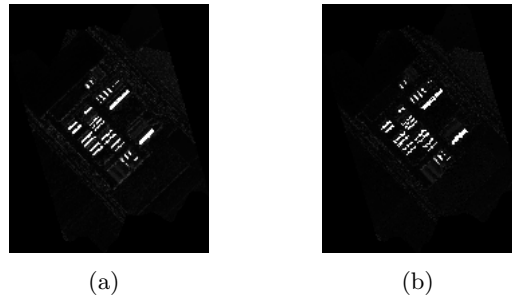


Figura 5.75: Visualización de los resultados de la precisión espacial para las subescenas de Cantera.
 (a) Comparación de la versión geocorregida con el ground-truth. (b) Comparación de la versión DMF con el ground-truth.

	Nº Mediciones	$\bar{\alpha}$
Geocorregida	16787 med	3.33°
DMF	20449 med	2.98°

Tabla 5.12: Ángulo espectral medio de las mediciones almacenadas en las imágenes geocorregidas y DMF, respecto a sus análogas en la imagen ground-truth.

5.6.4. Análisis de los resultados del experimento.

Para el estudio de precisión espacial a la hora de georreferenciar los datos, tanto para las versiones geocorregidas como para las versiones DMF, se realizaron los siguientes experimentos:

1. Estudio de la precisión espacial para el *dataset sintético*:
 - a) Análisis de la asignación de mediciones a los píxeles de la imagen geocorregida:
 - 1) Análisis visual de los tipos de píxeles en la imagen geocorregida final (sin medida asignada, con medida correspondiente asignada, y con medida asignada por vecino más próximo) (Figura 5.47).
 - 2) Cálculo del porcentaje de cada tipo de píxel en la imagen geocorregida final (sin medida asignada, con medida correspondiente asignada, y con medida asignada por vecino más próximo), comparado con el número de celdas ocupadas y a *Null* en la versión DMF análoga (Figura 5.48).
 - 3) Análisis visual de los tipos de píxeles en la imagen geocorregida final (sin medida asignada, con medida correspondiente asignada, y con medida asignada por vecino más próximo), considerando el ángulo de vuelo (Figura 5.49).
 - 4) Cálculo del porcentaje de cada tipo de píxel en la imagen geocorregida final (sin medida asignada, con medida correspondiente asignada, y con medida asignada por vecino más próximo), comparado con el número de celdas ocupadas y a *Null* en la versión DMF análoga, considerando el ángulo de vuelo y un tamaño de celda DMF calculado de forma automática (Figura 5.50).
 - b) Análisis de los desplazamientos de las mediciones respecto a su localización en el proceso de geocorrección:
 - 1) Análisis visual de las distancias de cada medición situada en un píxel de la imagen geocorregida respecto a la localización original proporcionada por el sistema de posicionamiento (Figura 5.51).
 - 2) Cálculo de la distancia media de cada medición respecto a la localización original proporcionada por el sistema de posicionamiento, de acuerdo al tipo de píxel (sin medida asignada, con medida correspondiente asignada, y con medida asignada por vecino más próximo) (Figura 5.52).

- 3) Análisis visual de las distancias de cada medición situada en un píxel de la imagen geocorregida respecto a la localización original proporcionada por el sistema de posicionamiento, considerando el ángulo de vuelo (Figura 5.53).
 - 4) Cálculo de la distancia media de cada medición respecto a la localización original proporcionada por el sistema de posicionamiento, de acuerdo al tipo de píxel (sin medida asignada, con medida correspondiente asignada, y con medida asignada por vecino más próximo), considerando el ángulo de vuelo (Figura 5.54).
- c) Análisis del porcentaje de utilización de medidas originales para la construcción de la imagen geocorregida:
- 1) Análisis visual del número de veces que cada medida ha sido considerada para ser almacenada en un píxel (Figura 5.55).
 - 2) Cálculo del porcentaje global de mediciones que no han sido consideradas para ser almacenadas en un píxel (Figura 5.56).
 - 3) Cálculo del número medio y máximo de veces que una mediciones es considerada para ser almacenada en un píxel (Figura 5.57).
 - 4) Análisis visual del número de veces que cada medida ha sido considerada para ser almacenada en un píxel, considerando el ángulo de vuelo (Figura 5.58).
 - 5) Cálculo del porcentaje global de mediciones que no han sido consideradas para ser almacenadas en un píxel, considerando el ángulo de vuelo (Figura 5.59).
 - 6) Cálculo del número medio y máximo de veces que una mediciones es considerada para ser almacenada en un píxel, considerando el ángulo de vuelo (Figura 5.60).

2. Estudio de la precisión espacial para el *dataset real*:

- a) Análisis de la asignación de mediciones a los píxeles de la imagen geocorregida:
- 1) Análisis visual de los tipos de píxeles en la imagen geocorregida final (sin medida asignada, con medida correspondiente asignada, y con medida asignada por vecino más próximo) (Figura 5.61).
 - 2) Cálculo del porcentaje de cada tipo de píxel en la imagen geocorregida final (sin medida asignada, con medida correspondiente asignada, y con medida asignada por vecino más próximo), comparado con el número de celdas ocupadas y a *Null* en la versión DMF análoga (Figura 5.63).
 - 3) Análisis visual de los tipos de píxeles en la imagen geocorregida final (sin medida asignada, con medida correspondiente asignada, y con medida asignada por vecino más próximo), considerando el ángulo de vuelo (Figura 5.62).
 - 4) Cálculo del porcentaje de cada tipo de píxel en la imagen geocorregida final (sin medida asignada, con medida correspondiente asignada, y con medida asignada por vecino más próximo), comparado con el número de celdas ocupadas y a *Null* en la versión DMF análoga, considerando el ángulo de vuelo y un tamaño de celda DMF calculado de forma automática (Figura 5.64).
- b) Análisis de los desplazamientos de las mediciones respecto a su localización en el proceso de geocorrección:
- 1) Análisis visual de las distancias de cada medición situada en un píxel de la imagen geocorregida respecto a la localización original proporcionada por el sistema de posicionamiento (Figura 5.65).
 - 2) Cálculo de la distancia media de cada medición respecto a la localización original proporcionada por el sistema de posicionamiento, de acuerdo al tipo de píxel (sin medida asignada, con medida correspondiente asignada, y con medida asignada por vecino más próximo) (Figura 5.66).
 - 3) Análisis visual de las distancias de cada medición situada en un píxel de la imagen geocorregida respecto a la localización original proporcionada por el sistema de posicionamiento, considerando el ángulo de vuelo (Figura 5.67).
-

- 4) Cálculo de la distancia media de cada medición respecto a la localización original proporcionada por el sistema de posicionamiento, de acuerdo al tipo de píxel (sin medida asignada, con medida correspondiente asignada, y con medida asignada por vecino más próximo), considerando el ángulo de vuelo (Figura 5.68).
- c) Análisis del porcentaje de utilización de medidas originales para la construcción de la imagen geocorregida:
- 1) Análisis visual del número de veces que cada medida ha sido considerada para ser almacenada en un píxel (Figura 5.69).
 - 2) Cálculo del porcentaje global de mediciones que no han sido consideradas para ser almacenadas en un píxel (Figura 5.70).
 - 3) Cálculo del número medio y máximo de veces que una mediciones es considerada para ser almacenada en un píxel (Figura 5.71).
 - 4) Análisis visual del número de veces que cada medida ha sido considerada para ser almacenada en un píxel, considerando el ángulo de vuelo (Figura 5.72).
 - 5) Cálculo del porcentaje global de mediciones que no han sido consideradas para ser almacenadas en un píxel, considerando el ángulo de vuelo (Figura 5.73).
 - 6) Cálculo del número medio y máximo de veces que una mediciones es considerada para ser almacenada en un píxel, considerando el ángulo de vuelo (Figura 5.74).
- d) Comparación de las imprecisiones espaciales introducidas por las versiones geocorregidas y DMF, aplicado a un caso real (Figura 5.75 y Tabla 5.12).

Respecto al primer experimento, en el cual se analizaron los píxeles a los que se les habían sido realmente asignados su medición correspondiente ($GLT_p > 0$), los que se les habían asignado una medición por vecino más próximo al quedar sin asignar en un principio ($GLT_p < 0$), y los que almacenan un espectro a 0 por estar fuera de la escena ($GLT_p = 0$), es posible comprobar por las representaciones visuales, que los píxeles $GLT_p < 0$ suelen encontrarse en las zonas de mayor turbulencia, o en donde la adquisición de mediciones no sigue el patrón regular. Las gráficas demuestran que, por normal general, el porcentaje de píxeles $GLT_p > 0$ es superior al de $GLT_p < 0$, aunque la diferencia entre ambas, así como el porcentaje de píxeles $GLT_p = 0$, depende de cada imagen. Para el *dataset sintético*, el incremento del ángulo de la plataforma aumenta también el porcentaje de mediciones por vecino más próximo ($GLT_p < 0$), por lo que supone, de igual manera, un incremento de las imprecisiones. El tipo de movimiento de plataforma que supone una mayor asignación de más mediciones por vecino más próximo ($GLT_p < 0$) es *roll*, pues el espacio entre mediciones es mayor cuanto más lejos están del Nadir, y ese espacio es rellenado por mediciones cercanas. Sin embargo, el tipo de movimiento de plataforma que supone una menor asignación de más mediciones por vecino más próximo ($GLT_p < 0$) es *yaw*, pues la escena permanece estática aunque girada. Este último caso sí que presenta, obviamente, un mayor número de píxeles con espectros a 0 ($GLT_p = 0$), al ser el número de píxeles fuera de la escena mayor estando ésta rotada. Por el contrario, es el tipo de movimiento *roll* el que presenta un menor porcentaje de píxeles ($GLT_p = 0$) pues la escena siempre se encuentra orientada en el sentido del vuelo. La aplicación del mismo experimento al *dataset real* corroboran estas conclusiones. El porcentaje de píxeles $GLT_p < 0$ y $GLT_p = 0$ varían de acuerdo a las condiciones de la imagen. El porcentaje de píxeles $GLT_p < 0$ depende directamente del nivel de turbulencias, mientras que el porcentaje de píxeles $GLT_p = 0$ depende de la dirección del vuelo. Igual que ocurría para el *dataset real*, el porcentaje de píxeles $GLT_p < 0$ es generalmente menor que el porcentaje de píxeles $GLT_p > 0$. El hecho de considerar el ángulo de vuelo, en contrario a lo que cabía esperar, no supone una mejora considerable mediante el decremento del porcentaje de píxeles $GLT_p < 0$. Esto es debido a que la asignación de mediciones se realiza siempre sobre una matriz (imagen geocorregida) en la que ya la latitud y longitud del centro de cada píxel tiene implícitos la rotación de la misma, y en la que el único error cometido es el de la precisión del ángulo de vuelo, el cual es demasiado pequeño como para introducir imprecisiones notables. Los errores espaciales al considerar el ángulo de vuelo sí que serían mayores siempre que se rotara la imagen después de haber sido geocorregida, pero como es considerado durante el proceso de geocorrección, siempre se seleccionará la medición más cercana al centro del píxel

correspondiente, minimizando el error. En lo que sí influye la consideración del ángulo de vuelo es en el porcentaje de píxeles $GLT_p = 0$, pues al situar la escena en sentido del vuelo, el porcentaje de píxeles fuera de la escena disminuye considerablemente. Por otro lado, y en comparación con las versiones en formato DMF análogas, el porcentaje de píxeles $GLT_p > 0$ es siempre muy similar al porcentaje de celdas DMF ocupadas ($DMF_c \neq Null$), mientras que el porcentaje de celdas a $Null$ ($DMF_c = Null$) suele coincidir con las posiciones en donde a los píxeles les han sido asignados mediciones por vecino más próximo, o bien espectros a 0 al estar fuera de la escena ($GLT_p < 0 + GLT_p = 0$). Es por este motivo por el cual el formato DMF presenta siempre menos imprecisiones espaciales que las versiones análogas geocorregidas por métodos convencionales, al no asignar mediciones por vecino más próximo en localizaciones en las que no ha sido adquirido dato. Esto además supone un decremento en consumo de memoria, tal y como se comprobó en la Sección 5.4, al no tener que almacenar espectros de n bandas a 0 para los píxeles situados fuera de la escena.

En cuanto al segundo experimento, en el que se realizó un análisis de los desplazamientos de las mediciones introducidos por el proceso de geocorrección respecto a las coordenadas originales de cada medición, es posible comprobar por las representaciones visuales correspondientes, que las mayores imprecisiones espaciales (distancias respecto al origen) se localizan principalmente en áreas de turbulencias y en zonas donde la distancia en el *sampleado* es mayor. Las mediciones almacenadas en dichos píxeles corresponden a mediciones asignadas por vecino más próximo ($GLT_p < 0$), por ello presentan mayores distancias respecto a su localización original proporcionada por el sistema de posicionamiento. Para el *dataset sintético*, el incremento del ángulo de la plataforma influye directamente en el promedio de distancias de las mediciones, siendo esta influencia mayor para el tipo de movimiento *roll*, al existir un mayor número de píxeles $GLT_p < 0$ debido a la variabilidad del *sampleado* a medida que se alejan del Nadir. Por el contrario, el tipo de movimiento *yaw* no es influenciado por este hecho de acuerdo al ángulo, al estar constituido por una escena estática pero girada. Para el caso del *dataset real*, las conclusiones que pueden extraerse son las mismas. Las mediciones que presentan un mayor desplazamiento respecto a su localización original son aquellas que están presentes en áreas de turbulencias o con un *sampleado* variable. El hecho de considerar el ángulo de vuelo, y tal como ocurría en el experimento anterior, no influyen en gran medida en las imprecisiones espaciales. El motivo es el mismo, en el momento de realizar el proceso de geocorrección este ángulo ya va implícito en la posición de cada píxel, por lo que se asigna la medición que más se aproxima a dicha localización. Comparado con el formato DMF, el desplazamiento ($d = [\Delta x, \Delta y]$) producido al asignar una medición no existe en las versiones DMF análogas, ya que éstas, aunque estén almacenadas en una celda DMF determinada, siempre contendrán la localización proporcionada por el sistema de posicionamiento. De esta manera, con el formato DMF el error espacial cometido siempre estará limitado al error de posicionamiento del dispositivo ($d_0 = [\Delta x_0, \Delta y_0]$). Sin embargo, este error procedente del sistema de posicionamiento también está presente en las versiones geocorregidas, por lo que ha de ser añadido al error ya cometido durante el proceso de geocorrección ($d + d_0 = [\Delta x, \Delta y] + [\Delta x_0, \Delta y_0]$).

Respecto al tercer experimento, en el que se realizó un análisis del porcentaje de mediciones originales utilizadas para construir la imagen geocorregida, los resultados visuales muestran que el número de veces que una medición es seleccionada no es constante, y existen algunas de ellas que son consideradas varias veces en el proceso de geocorrección. Estas mediciones responden a zonas de turbulencias o de *sampleado* variable, en las que los datos son adquiridos bajo circunstancias extremas. El hecho de seleccionar varias veces una medición para la construcción de la imagen geocorregida final genera una repetición y una inserción de información falsa que puede producir imprecisiones a la hora del procesamiento. Para el caso del *dataset sintético*, el movimiento de plataforma que produce una mayor repetición de los datos es *roll*, que como se demostró en los dos experimentos anteriores, es el tipo de movimiento que produce mayor porcentaje de imprecisiones espaciales, llegando a extremos en el que una misma medición puede llegar a ser seleccionada hasta 10 veces. Los movimientos de plataforma *pitch* y *yaw* presentan un mejor comportamiento, y el promedio de número de veces que una medición es seleccionada no llega a 2. Por otro lado, al igual que existen casos en los que una misma medición es seleccionada varias veces, también existen casos en los que una medición no es

seleccionadas ni una vez, generando una pérdida de información válida al no estar presentes en la imagen geocorregida final. El movimiento de plataforma más afectado por este último hecho es *yaw*, en el cual el porcentaje de mediciones no utilizadas alcanza el 7 % en los ángulos extremos. Por tanto, el hecho de no considerar el ángulo de rotación en el proceso de geocorrección no influye en gran medida en la inserción de imprecisiones espaciales, tal y como se demostró en los dos experimentos anteriores, pero sí que provoca un peor aprovechamiento de los datos recogidos por los sensores. El movimiento de plataforma que mejor aprovecha la asignación de los datos originales es *roll*, ya que, como se demostró en los anteriores experimentos, es el que presenta mayores imprecisiones y mayor repetición en la selección de mediciones, y por ello el porcentaje de mediciones no seleccionadas disminuye considerablemente a casi el 0 %. Un caso peculiar es el presentado por el tipo de movimiento *pitch* (Figura 5.56(a)) en el que presenta un porcentaje de mediciones originales no asignadas de cerca del 7 % para el ángulo extremo -30° , y cerca del 0 % para el ángulo extremo 30° . Esto es debido a que si se consideran ángulos negativos, la trayectoria del avión es descendente, por lo que la distancia de *sampleado*, tanto en longitud como en latitud, es mayor cuando se considera un tamaño de píxel y una velocidad del avión constantes y un ángulo de inclinación mayor. Dichas distancias se hacen variables, no tan grandes, cuando el avión parte del mismo punto y con una misma altura inicial, pero con una trayectoria ascendente (con un ángulo a partir de 0°). Es por ello por lo que en este caso, el porcentaje de mediciones no asignadas decrece hasta cerca del 0 %. El hecho de considerar el ángulo de vuelo en el proceso de geocorrección no influye en gran medida en la media global del número de veces que una medición es seleccionada para asignación, pero sí que permite corregir el problema presentado con anterioridad referente al alto porcentaje de mediciones que quedan sin ser seleccionadas. Al encontrarse la imagen geocorregida final rotada respecto a la dirección en la que fue adquirida la escena, el aprovechamiento de los datos adquiridos por los sensores se optimiza, alcanzando porcentajes máximos de mediciones no asignadas cercanos al 0.0005 %. La aplicación del experimento al *dataset práctica* corrobora las conclusiones extraídas. El porcentaje de mediciones utilizadas para la construcción de la imagen geocorregida depende del nivel de turbulencias a las que fue sometida la plataforma, así como de la variabilidad de las distancias del *sampleado*. Igualmente, el hecho de considerar el ángulo de vuelo en el proceso de geocorrección no implica mejoras considerables en dicho porcentaje, aunque sí que supone la optimización del aprovechamiento de los datos originales. Comparando los resultados con el formato DMF, es necesario denotar que este formato no supone la repetición de ninguna medición (se elimina la redundancia de información), y todas las mediciones originales son consideradas para ser almacenadas en la matriz DMF, aprovechando los datos adquiridos por los sensores al 100 %. Esto implica una optimización en la georreferenciación de la información que no se consigue con los procedimientos de geocorrección convencionales.

El cuarto experimento se analizó cómo los errores espaciales introducidos en el proceso de geocorrección afectaban a las imágenes resultantes aplicado a un caso real, en el que una subimagen de *Cantera* fue geocorregida por métodos convencionales y, por otro lado, convertida a formato DMF, para una posterior comparación con un *ground-truth*. En los resultados obtenidos, es posible comprobar que en las imágenes resultantes (donde se representa el ángulo espectral para cada medición) las mayores diferencias se sitúan en las zonas de bordes, acentuándose concretamente en el lugar que ocupan las naves industriales. Esto es debido a que la diferencia espectral entre el material de los tejados es mucho más alto que la diferencia entre diferentes tipos de vegetación, por lo que el desplazamiento de una medición desde o hasta dichas naves supondrá un mayor nivel de imprecisión.

Respecto al ángulo espectral medio, la versión DMF consigue ligeramente mejores resultados que la versión geocorregida, pues las diferencias encontradas respecto al *ground-truth* son menores. Sin embargo, es necesario considerar que, aunque la precisión espacial es la principal causante de esta diferencia entre ángulos espectrales medios, este resultado también se ve influenciado por otros parámetros:

- La escena está compuesta por grandes área con la misma información espectral (por ejemplo, las zonas que representan un mismo componente, como el tipo de vegetación en el campo), por lo que el desplazamiento de una medición puede implicar la comparación con otra medición

referencia en el *ground-truth*, situada varios metros, pero con el mismo tipo de espectro. De esta manera, el error espacial existe, pero no queda bien reflejado.

- Las diferencias espectrales respecto al *ground-truth*, por lo general, no son muy elevadas. Por lo que la media de los ángulos espectrales es cercano a 0, camuflando la existencia de zonas más pequeñas con mayor diferencia espectral o donde el formato DMF consigue un mejor y notable rendimiento.
- El hecho de comparar una imagen adquirida en un vuelo a alta altura respecto a un *ground-truth* consistente en un vuelo a baja altura supone considerar un *footprint* mayor en la primera respecto a la segunda. Es decir, el área de integración es mayor en el primer caso, por lo que una medición situada exactamente en la misma posición que su análoga en el *ground-truth* puede estar constituida por la mezcla de los espectros circundantes situados dentro del mismo área de integración. Esto generará un determinado error espectral que no debería ser considerado al ser el desplazamiento de la medición nulo. Por supuesto, al ser un problema generado en el proceso de adquisición que afecta a un nivel de medición, este fenómeno aparece tanto en la versión geocorregida como en la versión DMF.

Como conclusión general, y tras realizar un exhaustivo análisis de todos los resultados, es posible afirmar que los errores espaciales de las imágenes geocorregidas son producidos por varias circunstancias, siendo acentuadas en las zonas de turbulencias de la imagen, y en áreas en las que la distancia de *sampleado* es variable. Estas imprecisiones tienen diferentes orígenes:

- No utilizar toda la información recogida por los sensores, quedando un cierto porcentaje variable (dependiendo de las características de la imagen) sin considerar en el proceso de geocorrección.
- Utilizar mediciones ya asignadas en sus respectivos píxeles ($GLT_p > 0$) para asignar, por vecino más próximo, a aquéllos que han quedado sin asignar ($GLT_p < 0$), generando redundancia de información.
- Considerar la localización de la medición almacenada en un determinado píxel como las coordenadas del centro del mismo, en lugar de la localización original proporcionado por el sistema de posicionamiento. Esto supone un desplazamiento de la información que se acentúa si las mediciones han sido asignadas por vecino más próximo ($GLT_p < 0$).

Estos errores espaciales, tal y como se han demostrado en los resultados, son inexistentes para el formato DMF, por lo que el error espacial de las imágenes representadas en dicho formato queda reducido al error provocado por el dispositivo de posicionamiento.

5.7. Adaptación del formato DMF para campañas de campo.

Las campañas de campo son actividades muy importantes en el campo de la teledetección. Consisten en la adquisición de mediciones por un experto sobre un área de interés con la ayuda de un espectrómetro. Las mediciones recogidas, que abarcan diferentes materiales previamente identificados, suponen una referencia a la hora de realizar cualquier tipo de procesamiento (como la detección de objetos, análisis subpíxel, clasificaciones espectrales, etc.) a partir de imágenes de la misma escena (u otras similares situadas en diferentes zonas del planeta) adquirida por otros sensores o instrumentos a bordo de aviones o satélites. Ya que los espectros adquiridos *in situ* pueden ser considerados generalmente como *espectros puros*, y que cada espectro es típico de un determinado tipo de material bajo una serie de condiciones, es posible construir *librerías espectrales* [Bal09] que pueden estar de manera pública o privada a la comunidad científica para la elaboración de sus estudios y experimentos. Para una mejor precisión en la generación de resultados, en dichas librerías espectrales es común

almacenar las condiciones en las que fueron adquiridos los espectros puros, como por ejemplo la intensidad y color de la luz, ángulo de incidencia, o estado de actividad si el espectro adquirido pertenece a un determinado tipo de planta.

5.7.1. *Objetivos del experimento.*

Las mediciones recogidas durante una campaña de campo suelen estar almacenadas en un formato propietario de acuerdo al tipo de espectrómetro utilizado, generalmente de tipo texto con una sintaxis determinada. Al no estar basado en ficheros independientes para las coordenadas (IGM) y los valores de espectro (L0), la adaptación al formato DMF no es directa.

Por tanto, los objetivos que se pretenden cubrir con la elaboración de este experimento son los siguientes:

- Adaptación de las adquisiciones realizadas durante una campaña de campo al formato DMF.
- Estimación del tamaño de celda DMF y visualización de la información recogida.
- Identificación de los tipos de materiales recogidos y visualización de los mismos a través del formato DMF.

5.7.2. *Descripción del experimento.*

El resultado obtenido tras una campaña de campo es un fichero en un determinado formato, generalmente propietario, el cual es difícil de georreferenciar mediante metodologías convencionales. El formato DMF juega aquí un gran papel, ya que sería posible representar las mediciones adquiridas sobre la escena o región de interés, e incluso realizar cualquier tipo de procesamiento previo.

Sin embargo, es necesario adaptar el formato DMF para realizar una conversión desde el formato propietario. Para este experimento, se ha utilizado un caso particular de una campaña de campo desarrollada durante el proyecto Datasat, en el área de Barrax. Los datos han sido presentados con anterioridad e identificados como *Campaña*, y consta de un fichero de texto en el que se presenta, para cada medición, sus coordenadas espaciales, su espectro (144 bandas), y su identificador de tipo de material, siendo éstos festuca (1), agua (2), suelo descubierto (3), maíz (4) y trigo (5).

Debido a que el formato DMF reconoce únicamente los campos de *Longitud*, *Latitud* y *Datos Instrumento* (espectro) en su diseño original (Figura 4.4), es interesante considerar también el tipo de material al que corresponde cada medición. De esta manera, se ampliará el registro DMR con un nuevo campo (*ID_Material*), tal y como se detalló en la Sección 4.1.2.1, para almacenar este parámetro (Figura 5.76). La distribución de las mediciones sobre la matriz DMF se realizará siguiendo el procedimiento convencional.

Una de las principales dificultades que aparecen a la hora de georreferenciar los datos es el desconocimiento del tamaño de píxel. Sin embargo, es posible determinarlo mediante la utilización de la metodología de cálculo automático de celda DMF presentado en la Sección 5.3. Pero es necesario tener en cuenta que las distancias de los datos adquiridos por sensores a bordo de aviones y satélites, siempre que se haya mantenido una dirección N/S, son determinados por la geometría del sensor (en longitud) y por la velocidad del vuelo (en latitud), que pueden ser más o menos constantes dependiendo de las turbulencias a las que ha sido sometida la plataforma. Para el caso de las campañas de campo, la forma en la que se distribuyen los datos puede ser totalmente aleatoria, sin seguir patrón de ningún tipo. Esto complica la estimación del cálculo automático del tamaño de celda DMF, ya que este procedimiento estaba basado en el estudio de las distancias entre mediciones consecutivas (utilizando el fichero IGM), mientras que en el caso de las campañas de campo este parámetro es inválido.

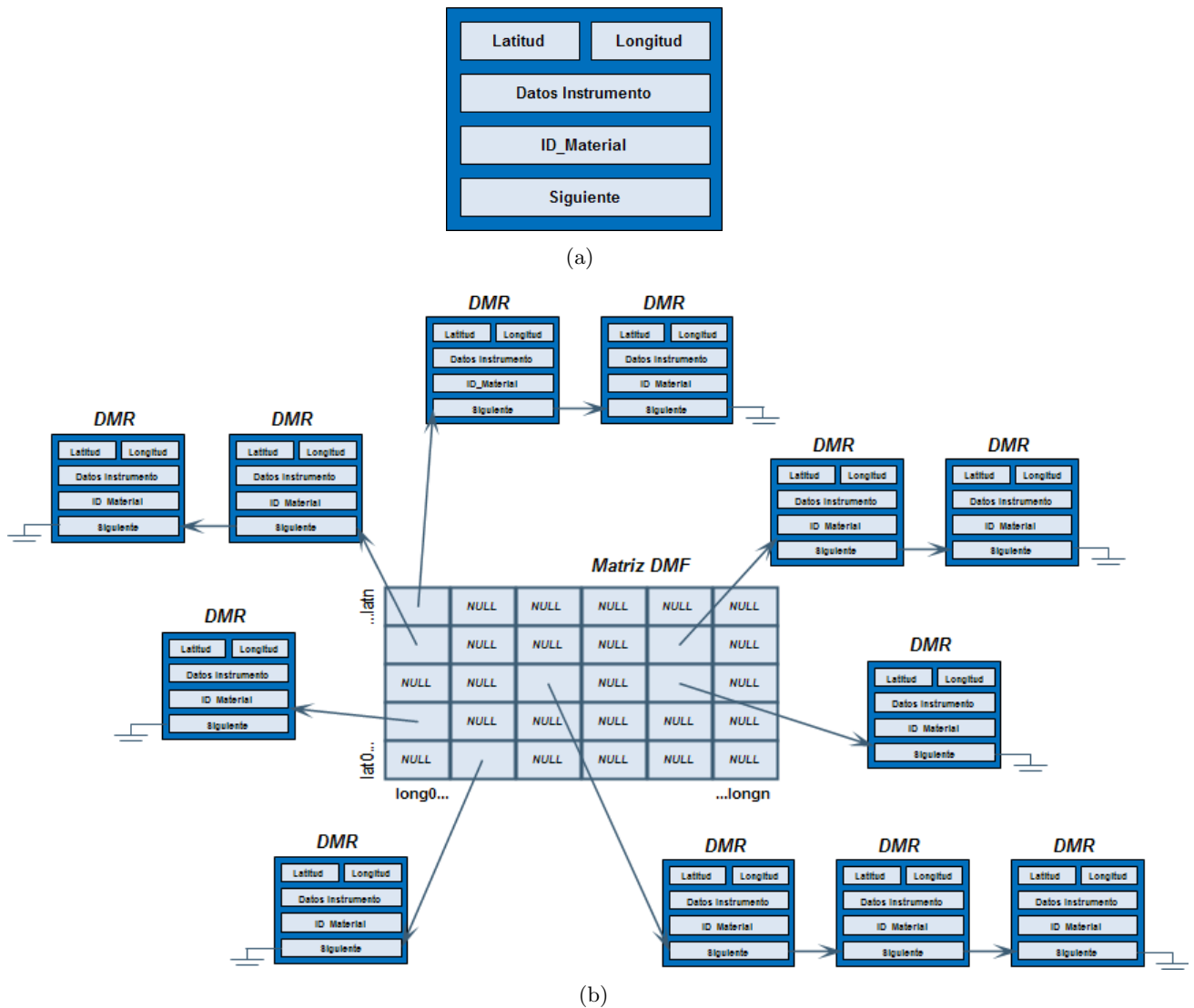


Figura 5.76: Formato DMF adaptado para campañas de campo.

(a) Modificación del registro DMR para campañas de campo. (b) Almacenamiento de registros DMR en la matriz DMF.

De esta manera, el procedimiento de cálculo automático del tamaño de celda DMF presentado en la Sección 5.3 ha sido ligeramente modificado:

1. Cálculo del tamaño de celda DMF en longitud:

- Para cada medición m , determinar la medición más cercana espacialmente m' mediante el cálculo de la distancia Euclídea.
- Calcular la distancia en longitud entre ambas mediciones, tal que $m_{long} - m'_{long}$.
- Determinar la media de todas las distancias mínimas en longitud, considerando este valor el tamaño de celda DMF en longitud.

2. Cálculo del tamaño de celda DMF en latitud:

- Para cada medición m , determinar la medición más cercana espacialmente m' mediante el cálculo de la distancia Euclídea.
- Calcular la distancia en latitud entre ambas mediciones, tal que $m_{lat} - m'_{lat}$.
- Determinar la media de todas las distancias mínimas en latitud, considerando este valor el tamaño de celda DMF en latitud.

Tras el cálculo del tamaño de celda DMF tanto en longitud como en latitud, es posible construir una matriz DMF que abarque el área de interés de manera apropiada, tanto para la visualización, como para un posterior procesamiento.

5.7.3. Resultados del experimento.

A partir del fichero de texto que almacena todos los datos correspondientes a las mediciones adquiridas en el área de interés, se extrajeron las coordenadas espaciales, los valores de espectro y los identificadores de material.

El procedimiento de cálculo automático del tamaño de celda DMF fue ejecutado utilizando las coordenadas espaciales de las mediciones, obteniendo un tamaño de celda DMF de 0.7 m para longitud, y 0.7 m para latitud. Teniendo en cuenta esta información, se construyó la matriz DMF, se construyeron los registros DMR con las coordenadas espaciales de las mediciones, sus espectros y sus identificadores de material, y fueron situados sobre dicha matriz en sus posiciones correspondientes.

El resultado de la visualización se muestra en la Figura 5.77, en donde es posible comprobar que el experto se movió principalmente en 5 sub-áreas de interés, adquiriendo diferentes mediciones en cada una de ellas, y haciendo que éstas aparezcan distribuidas en forma de grupo.

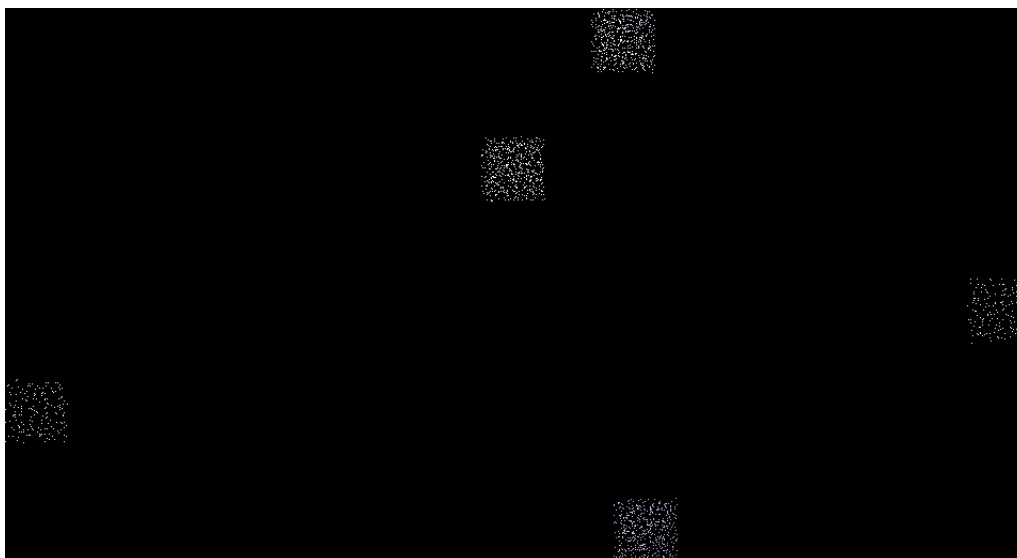


Figura 5.77: Visualización de la matriz DMF adaptada a campañas de campo [Banda 10].

Como el identificador de material de cada medición ha sido almacenado en el campo *ID_Material* de los registros DMR, es posible realizar un mapa de materiales pintando cada uno de ellos de un determinado color sobre la matriz DMF. La Figura 5.78 muestra los tipos de materiales analizados espectralmente, utilizando el siguiente esquema de color:

- *Festuca*: Verde.
- *Agua*: Azul.
- *Suelo descubierto*: Rojo.
- *Maíz*: Amarillo.
- *Trigo*: Blanco.

Como es posible comprobar, el experto estuvo adquiriendo muestras en 5 diferentes subáreas cuadradas de unos 40×40 m, en las que es característico un único tipo de material.

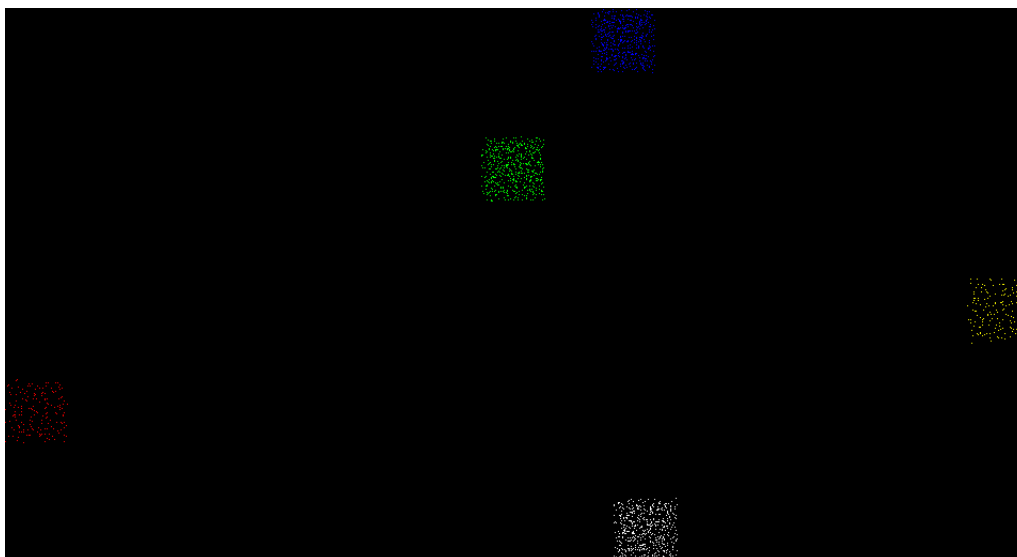


Figura 5.78: Visualización de la matriz DMF de acuerdo a los tipos de materiales analizados.

5.7.4. Análisis de los resultados del experimento.

En este experimento, en el cual se adaptó el formato DMF para el procesamiento y manipulación de los datos adquiridos *in situ* durante el transcurso de una campaña de campo, se obtuvieron los siguientes resultados:

1. Visualización de los datos adquiridos por el espectrómetro (Figura 5.77).
2. Identificación de los tipos de materiales adquiridos por el espectrómetro (Figura 5.78).

La adaptación del formato DMF a las campañas de campo supone una gran ventaja a la hora de poder almacenar, representar y visualizar con gran precisión todos los datos referentes a las mediciones adquiridas.

El principal problema que surge de esta adaptación es el hecho de que, por regla general, cada espectrómetro proporciona la información recogida en formatos propietarios, por lo que es necesario conocer bien el formato origen para una correcta conversión, así como realizar esta operación por cada tipo de espectrómetro. En este aspecto, el formato DMF puede suponer una gran alternativa como formato general.

El segundo problema aparece a la hora de construir la matriz DMF en la que disponer las mediciones en formato registro DMR. El experto pudo haber seguido una trayectoria irregular e incluso adquirir mediciones en diferentes subáreas previamente establecidas, por lo que las distancias entre dichas mediciones puede ser variable. Es necesario modificar ligeramente el procedimiento de cálculo automático del tamaño de celda DMF presentado en la Sección 5.3 y basar el principio de la metodología en otro parámetro, como por ejemplo, la media de distancias mínima. Este parámetro permite una correcta representación de los datos adquiridos, tal y como se muestra en la Figura 5.77. Esto supone una gran ventaja respecto a los formatos de datos propios de los espectrómetros, ya que es posible visualizar la información recogida sobre la propia escena, así como los valores de espectro a lo largo de las diferentes bandas y los tipos de materiales a los que corresponden.

Aparte de la visualización y el almacenamiento de los datos correspondientes a la campaña de campo, una vez construida la matriz DMF también existe la posibilidad de realizar las operaciones y procesamientos específicos de este formato.

5.8. Fusión de datos.

Muchas metodologías requieren el análisis de la información recogida no únicamente por un sensor, sino por diferentes sensores que pueden tener diferente naturaleza y origen, y cuyos datos han sido obtenidos bajo diferentes circunstancias. Este hecho dificulta la comparación directa de la información, así como la correspondencia entre mediciones. En este ámbito, los procesos de representación de imágenes convencionales, basados en matrices en la que los datos de diferentes orígenes son representados por estructuras de datos (y ficheros) individuales, no facilitan la localización de áreas comunes para análisis y estudios más concretos y detallados.

5.8.1. *Objetivos del experimento.*

En este experimento se adaptó el formato DMF para el almacenamiento de mediciones procedentes de diferentes sensores, con el propósito de almacenar toda la información en una misma estructura para un más fácil análisis y procesamiento.

Los objetivos que se pretenden alcanzar con este estudio son los siguientes:

- Adaptación del formato DMF como estructura de datos única para el almacenamiento de mediciones de diferentes orígenes.
- Análisis del estado de la matriz DMF tras el almacenamiento de mediciones de diferentes orígenes.
- Estudio de la dispersión y localización de los datos de los diferentes sensores sobre la matriz DMF.
- Estudio de la densidad de las celdas DMF tras el almacenamiento de mediciones de diferentes orígenes.

5.8.2. *Descripción del experimento.*

Este estudio partió de una necesidad existente en la comunidad de científicos involucrados en la misión conjunta FLEX / Sentinel-3, aún en desarrollo por la European Space Agency (ESA) [Mor09, Gua10], y que supondrá el lanzamiento de dos satélites que orbitarán en *tándem* y que trabajarán juntos para generar productos de datos optimizados.

La misión FLEX (Fluorescence Explorer) [Ras08] está siendo desarrollada con el objetivo de realizar observaciones, a un nivel global, del proceso de fotosíntesis a través de la medición de la fluorescencia de la clorofila. Este tipo de radiación es emitida por la vegetación en las regiones del visible e infrarrojo cercano (NIR) del espectro electromagnético y proporciona gran información sobre la actividad fotosintética de las plantas. Esta misión supone un innovador proceso de monitorizar la actividad vegetal desde el Espacio. Para poder capturar la débil señal de la fluorescencia de manera correcta, FLEX hará uso del Fluorescence Imager Spectrometer (FIMAS) [Kra10] (Figura 5.79(a)), un espectrómetro de alta resolución espectral con geometría *pushbroom*, el cual proporcionará mediciones de la fluorescencia principalmente en las frecuencias O2-A y O2-B, así como de otras frecuencias que permiten un mejor análisis de la información recogida por el sensor.

Por otro lado, la misión Sentinel-3 [Agu07] está siendo desarrollada para ofrecer apoyo a servicios relacionados con el entorno marítimo, así como la aplicación indirecta a numerosas áreas, como la terrestre, la atmosférica o la criosférica. Se basará en la elaboración, durante un período de 20 años, de una serie de satélites de aproximadamente 7 años de vida cada uno. El lanzamiento del primer satélite de la serie (Sentinel-3A) está planificado para finales del año 2013 [Ber12]. Dicho satélite dispondrá de dos instrumentos que trabajarán junto al sensor FIMAS para optimizar el estudio

del espectro electromagnético aplicado al análisis del intercambio del carbono entre las plantas y la atmósfera [Don08]. Estos instrumentos son:

- *Ocean & Land Color Instrument (OLCI)* (Figura 5.79(b)): Se trata de un espectrómetro programable que trabaja en el rango del visible e infrarrojo cercano (NIR). Está compuesto por cinco cámaras acopladas en forma de abanico de tal manera que forman una línea de adquisición. Por esto, puede considerarse como un instrumento de geometría *pushbroom*. Las cámaras estarán desplazadas 12° respecto al Nadir para evitar el reflejo del Sol sobre la escena (efectos *sunglint*).
- *Sea & Land Surface Temperature Radiometer (SLSTR)* (Figura 5.79(c)): Se trata de un scanner con observación biangular, es decir, ofrece dos líneas de escaneo diferentes en un mismo instante. La primera de ellas está situada en el Nadir, mientras que la segunda está un poco más retrasada espacialmente. Ambas líneas de adquisición no son rectas, sino curvilíneas, por lo que se puede considerar como una doble visión cónica. Esta característica permitirá optimizar la corrección atmosférica y hacerla robusta a cambios en los aerosoles y vapores de agua presentes. Además, dicha información correspondiente a los aerosoles, derivada de los canales proporcionados por la vista dual, también permitirá una mejora significativa en la precisión con la que se generarán los productos de vegetación y temperatura de superficie terrestre.

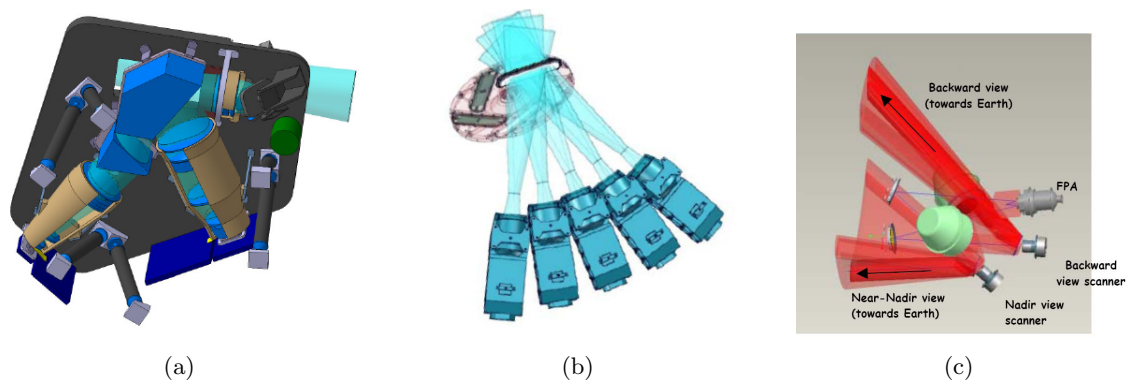


Figura 5.79: Representación de los instrumentos diseñados para la misión FLEX / Sentinel-3. (a) FIMAS. (b) OLCI. (c) SLSTR.

La Tabla 5.13 y la Figura 5.80 muestra algunas características técnicas de ambos satélites útiles para la elaboración de este estudio, como son el tamaño del *swath* presentado por cada línea de escaneo, así como la posición y forma de cada una de las cuatro líneas de escaneo sobre la escena para un determinado instante t .

	OLCI	SLSTR Nadir	SLSTR Backward	FIMAS
<i>Satélite</i>	Sentinel-3	Sentinel-3	Sentinel-3	FLEX
<i>Rango Espacial (Swath)</i>	1270 Km	1675 Km	750 Km	> 100 Km
<i>Resolución Espacial</i>	300 m	500 - 1000 m	500 - 1000 m	300 - 500 m
<i>Rango Espectral</i>	400 - 1020 nm	555 - 12000 nm	555 - 12000 nm	500 - 800 nm
<i>Resolución Temporal</i>	< 2 días	< 4 días	< 4 días	< 27 días
<i>Altitud Plataforma</i>	815 Km	815 Km	815 Km	815 Km
<i>Tipo Geometría</i>	Pushbroom	Cónica	Cónica	Pushbroom

Tabla 5.13: Comparación de las características y funcionalidades de los diferentes sensores.

De esta manera, el escaneo de la corteza terrestre por parte de los dos satélites generará como resultado cuatro imágenes diferentes, una por visión (o línea de escaneo). Por tanto, una escena determinada estará representada por mediciones de cuatro orígenes de datos diferentes. Al ser así, el formato DMF ha sido adaptado tal y como se detalló en la Sección 4.1.2.1. En esta versión modificada, el registro DMR contendrá un campo adicional *ID_Sensor*, que almacenará un identificador por vista u

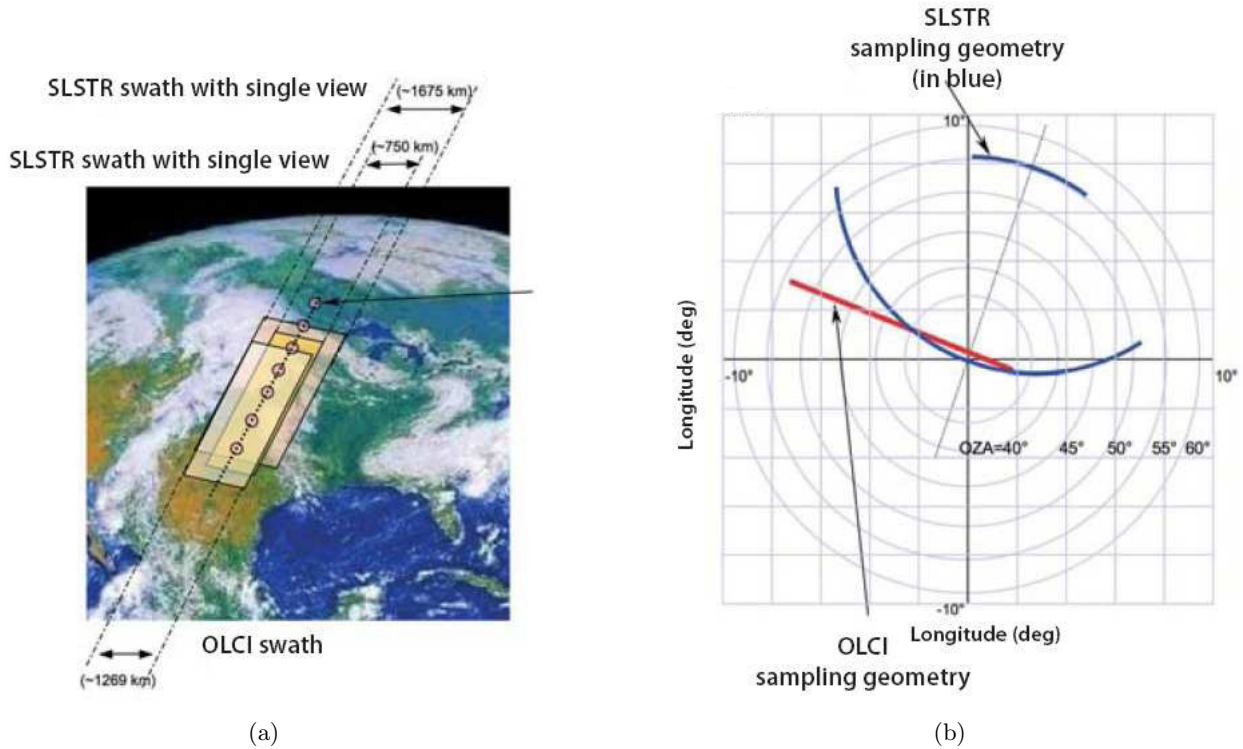


Figura 5.80: Información técnica de las líneas de escaneo de los instrumentos de Sentinel-3. (a) Comparativa de los tamaños de los swaths. (b) Localización y representación de las líneas de escaneo.

origen indicando de esta manera la procedencia de cada medición. Los identificadores fueron asignados de la siguiente manera:

- $ID_Sensor = 0$: Instrumento FIMAS.
- $ID_Sensor = 1$: Instrumento OLCI.
- $ID_Sensor = 2$: Instrumento SLSTR en su vista SLSTR Nadir (SLSTR_Nadir).
- $ID_Sensor = 3$: Instrumento SLSTR en su vista SLSTR Backward (SLSTR_Backward).

Esta modificación del registro DMR permite la inserción de las mediciones de diferentes orígenes en la misma matriz DMF, tal y como se indicó en la Figura 4.14(b) y en la Figura 4.14(c).

Para la conversión de una escena de datos adquiridos por diferentes sensores, hay que tener en cuenta que FLEX y Sentinel-3 no serán puestos en órbita hasta finales del año 2013. Por tanto, no se disponen de datos reales adquiridos por los instrumentos a bordo. De esta manera, se llevó a cabo una simulación del aspecto que tendrían los datos sobre una matriz DMF, mediante la utilización de imágenes obtenidas por el sensor AHS. Para ello, se utilizaron dos imágenes L0 de diferentes vuelos sobre la escena de Canteras. La primera de ellas es la imagen utilizada en todos los experimentos, la cual fue visualizada en la Figura 5.7(i) y en la Figura 5.7(j). La segunda imagen L0 (*Canteras2*) presenta las mismas características que la anterior (Tabla 5.14), ya que se trata del mismo sensor aerotransportado sobre el mismo área de estudio, pero en un instante diferente. La Figura 5.81 muestra una visualización de esta segunda imagen L0.

El hecho de elegir dos imágenes diferentes de la misma escena es porque de esta forma es posible simular la adquisición desde dos plataformas diferentes (FLEX y Sentinel-3). Es posible comprobar que las distorsiones presentes en ambas imágenes son diferentes, permitiendo una simulación de adquisición de mediciones procedentes de diferentes orígenes de una manera más real. De esta forma, la primera

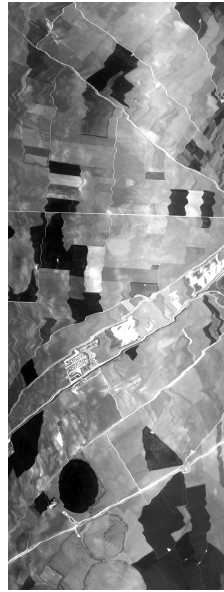


Figura 5.81: Visualización de la segunda imagen L0 sobre la escena de Cantera (Cantera2) [Banda 0].

	Cantera2
<i>Sensor</i>	AHS
<i>Plataforma</i>	Avión
<i>Sist. Coordenadas</i>	UTM (m)
<i>Fil. × Col. L0</i>	2000 × 750
<i>Nº Bandas</i>	63

Tabla 5.14: Detalle de las características de la segunda imagen L0 sobre la escena de Cantera (Cantera2).

imagen se utilizó para simular los datos procedentes del satélite FLEX, mientras que la segunda fue utilizada para simular los del Sentinel-3.

Con estas dos imágenes L0 se representa una escena escaneada teniendo en cuenta los posibles desplazamientos de la plataforma. A continuación, y a partir de ellas, es necesario simular el escaneo de los instrumentos de manera independiente, teniendo en cuenta el tamaño del *swath*, las líneas de adquisición y la resolución espacial de los sensores, tal y como se muestran en la Figura 5.80.

Para reflejar las dimensiones espaciales de la escena en las imágenes L0 resultantes, ambas imágenes fueron recortadas espacialmente y de manera relativa de acuerdo al *swath* de los instrumentos, por lo que la imagen L0 simulada correspondiente al instrumento FIMAS (con un *swath* de 100 Km) es bastante más estrecha que la imagen L0 simulada correspondiente a los instrumentos OLCI (con un *swath* de 1270 Km) y SLSTR (con *swaths* 1675 Km y 750 Km para las vistas SLSTR.Nadir y SLSTR.Backward, respectivamente).

Para reflejar la geometría de los sensores, se simuló la adquisición de los datos teniendo en cuenta las geometrías *pushbroom* y cónica de los instrumentos, tal y como se detalla a continuación:

- Para el caso de las geometrías *pushbroom* (FIMAS y OLCI), la adquisición de los datos se realizó simplemente recorriendo las correspondientes imágenes L0 a lo largo del eje Y, acorde a un ancho proporcional al *swath* de cada instrumentos.
- Para el caso de las geometrías cónicas, la adquisición de los datos es un poco más compleja. Para empezar, no se disponían de los datos exactos y reales de las geometrías cónicas mostradas en la Figura 5.80, por lo que se consideraron para las vistas SLSTR.Nadir y SLSTR.Backward dos elipses aproximados. Teniendo en cuenta que una elipse viene determinada, en píxeles, por la posición de su centro (c_x, c_y) y por la longitud de su eje mayor a y su eje menor b , para

cada píxel (x, y) de la imagen L0 (escena), el verdadero punto de adquisición del sensor (x', y') estará definida por la Ecuación 5.7.

Los valores de las elipses establecidos en la simulación para las dos vistas SLSTR fueron los siguientes:

- *SLSTR_Nadir*: $a = 600, b = 500, (c_x, c_y) = (375, 585)$.
- *SLSTR_Backward*: $a = 600, b = 500, (c_x, c_y) = (375, 200)$, con $y' = -y'$ para que la abertura de la línea de escaneo esté invertida.

Como ejemplo, la Figura 5.82 muestra las líneas de escaneo de OLCI, SLSTR_Nadir y SLSTR_Backward (x'_i, y'_j) para todos los píxeles situados en la primera línea de la imagen L0 (x_i, y_0) . Los píxeles (x', y') resultantes serán los considerados para construir las simulaciones de las imágenes L0 adquiridas por estos instrumentos.

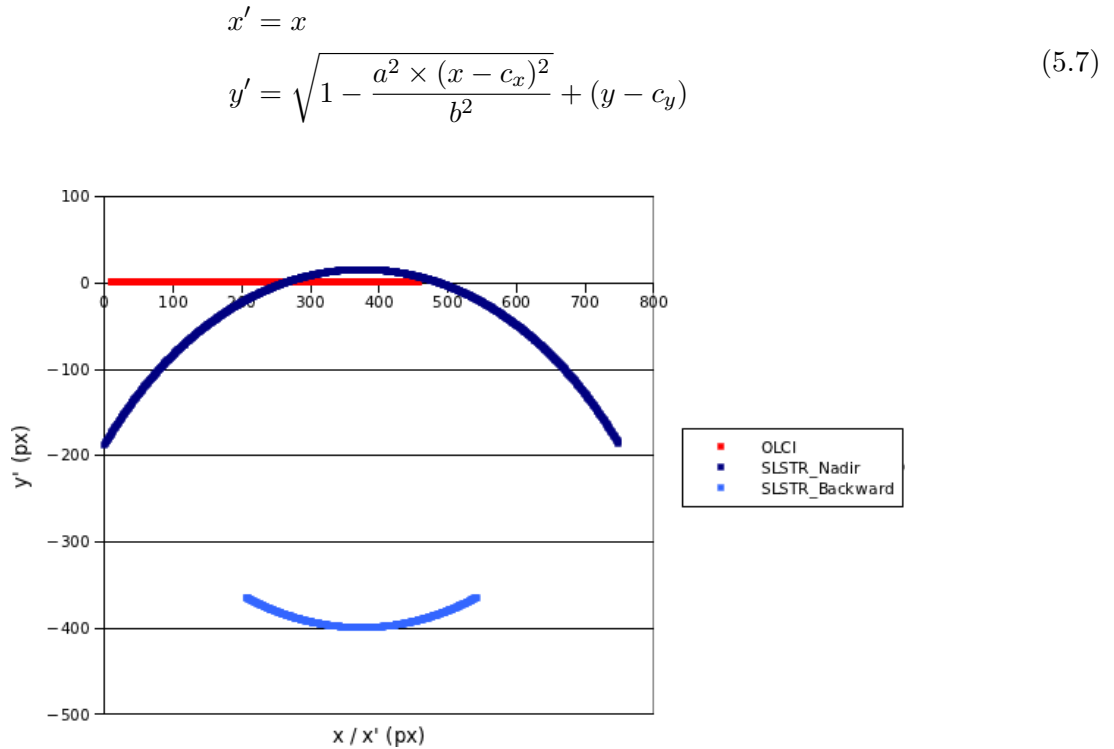


Figura 5.82: Líneas de adquisición simuladas para los instrumentos OLCI y SLSTR, considerando caso $y=0$.

Tras aplicar el escaneo de la escena L0 con las líneas de adquisición detalladas con anterioridad para cada instrumento, las imágenes simuladas obtenida por cada uno de ellos se muestran en la Figura 5.83. Para una mejor representación, dichas imágenes se muestran en horizontal en vez de en vertical. En dicha figura, se puede destacar principalmente el *swath* del sensor FIMAS, considerablemente más pequeño; la mayor resolución espacial del sensor FIMAS (más destacable en las naves del centro de las imágenes); y la adquisición cónica del sensor SLSTR (en sus dos vistas, ambas en sentido opuesto).

Finalmente, para reflejar las resoluciones espaciales de los instrumentos, ya que las de los sensores FIMAS y OLCI es aproximadamente el doble que las del sensor SLSTR, durante la simulación de la adquisición se considerarán todas las filas (y_i) y columnas (x_i) de la primera imagen L0 original (FIMAS), y de la segunda imagen L0 original (OLCI), mientras que para el caso de SLSTR se considerarán cada dos filas y cada dos columnas en la segunda imagen L0 original. De esta manera, la imagen simulada resultante en este último caso tendrá una resolución espacial de aproximadamente la mitad ($8.8 \times 8.8 \text{ m}$) respecto a las obtenidas para FIMAS y OLCI ($4.4 \times 4.4 \text{ m}$).

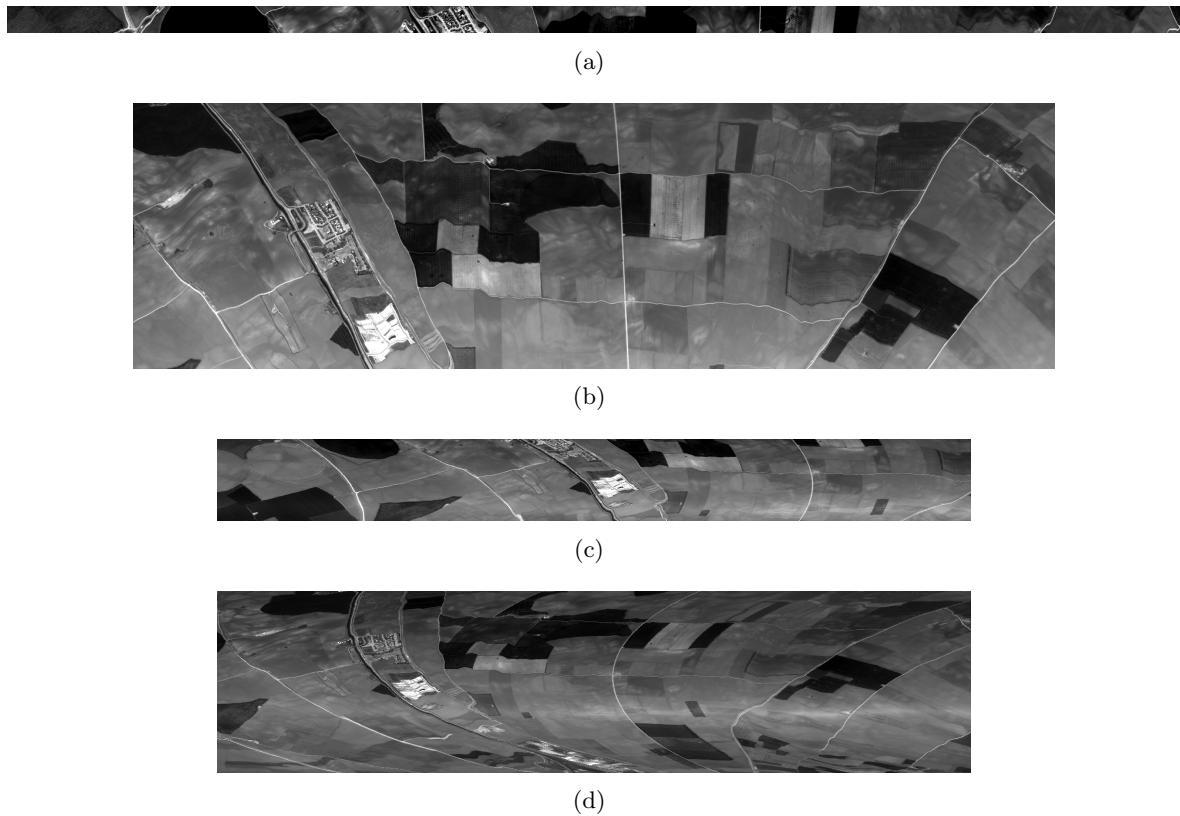


Figura 5.83: *Imágenes L0 simuladas para los diferentes orígenes.*

(a) *Imagen L0 simulada para FIMAS [Banda 0]. (b) Imagen L0 simulada para OLCI [Banda 0]. (c) Imagen L0 simulada para SLSTR_Backward [Banda 0]. (d) Imagen L0 simulada para SLSTR_Nadir [Banda 0].*

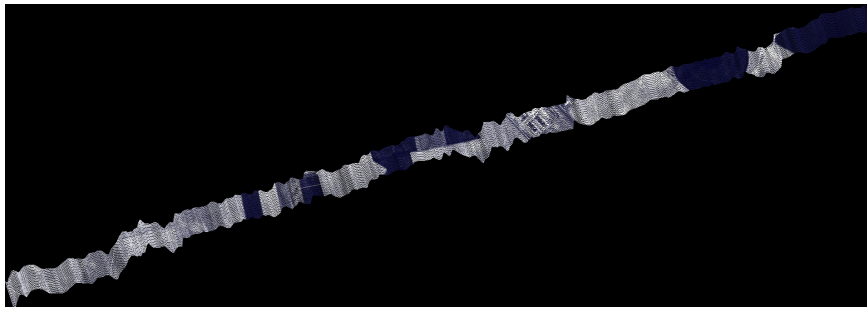
Por supuesto, para posteriormente poder construir la matriz DMF, es necesario que para cada píxel considerado en las imágenes L0, se almacenen también sus coordenadas espaciales. Por tanto, el proceso de simulación de adquisición de los datos generará un total de cuatro ficheros L0 simulados, junto con sus IGM simulados.

5.8.3. *Resultados del experimento.*

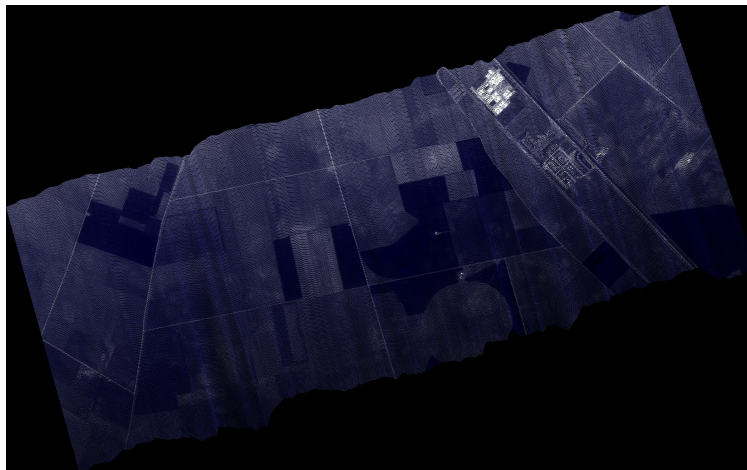
A partir de los ficheros IGM y L0 generados en la simulación de adquisición de los datos, se construyeron las matrices DMF correspondientes para comprobar visualmente cómo se llevaba a cabo la distribución de los datos. Como se comentó en la Sección 5.3, el principal problema encontrado a la hora de almacenar todas las mediciones procedentes de diferentes sistemas de adquisición sobre una misma estructura de datos, es la selección del tamaño de celda DMF. De esta manera, en este estudio la fusión de datos se realizó utilizando como tamaño de celda DMF el tamaño de píxel mayor de las imágenes simuladas originales (8.8×8.8 m), el menor (4.4×4.4 m) y el promedio (6.6×6.6 m)

En un principio, se construyeron las matrices DMF correspondientes a las imágenes generadas en la simulación de manera independiente, para comprobar el aspecto que presentaban las mediciones y que la geocorrección se realizaba de manera correcta. Dichas matrices DMF se muestran en la Figura 5.84.

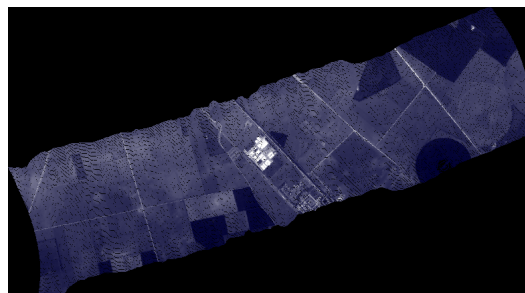
Tras esto, todas las mediciones adquiridas por los cuatro orígenes fueron introducidas en la misma matriz DMF utilizando los tres diferentes tamaños de celda DMF, tal y como se muestra en la Figura 5.85. Además, para analizar la dispersión y localización de los datos, se recorrieron todas las celdas DMF de cada matriz y se analizó de cuántas mediciones estaba compuesta cada una, así como sus procedencias. De esta forma, la Figura 5.86 muestra la densidad de mediciones en dicha matriz (para los diferentes tamaños de celda DMF), siendo el píxel más blanco el que más mediciones posee dentro



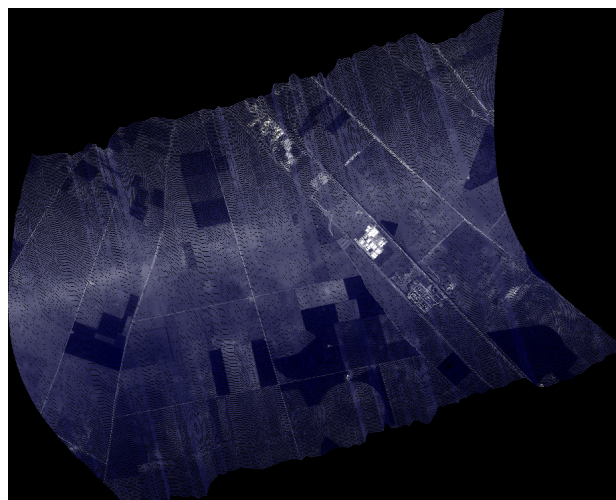
(a)



(b)



(c)



(d)

Figura 5.84: Versiones DMF de las imágenes simuladas.

(a) Versión DMF de la imagen simulada para FIMAS [Banda 0]. (b) Versión DMF de la imagen simulada para OLCI [Banda 0]. (c) Versión DMF de la imagen simulada para SLSTR_Backward [Banda 0]. (d) Versión DMF de la imagen simulada para SLSTR_Nadir [Banda 0].

de la celda DMF. De igual manera, la Figura 5.87 indica los orígenes de los datos (también para los diferentes tamaños de celda DMF), siendo cada medición coloreada acorde al sensor con la que fue adquirida.

5.8.4. *Análisis de los resultados del experimento.*

En el estudio sobre la adaptación del formato DMF para fusión de datos de diferentes orígenes, se realizó un experimento basado en la futura misión conjunta FLEX / Sentinel-3 de la ESA, con el cual se obtuvieron los siguientes resultados:

1. Generación de las imágenes L0 simuladas de acuerdo a las características de los experimentos (Figura 5.83).
2. Visualización independiente de cada una de las imágenes L0 simuladas en formato DMF (Figura 5.84).
3. Fusión de los datos utilizando el formato DMF con tamaños de celda DMF mínimo, máximo y medio (Figura 5.85).
4. Análisis de la densidad de registros DMR de acuerdo al tamaño de celda DMF mínimo, máximo y medio (Figura 5.86).
5. Identificación de los orígenes de datos utilizando el formato DMF con tamaños de celda DMF mínimo, máximo y medio (Figura 5.87).

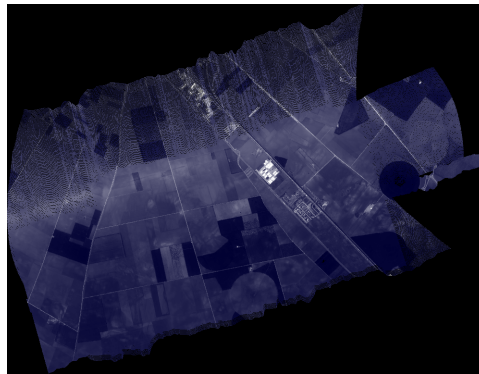
La adaptación del formato DMF para el almacenamiento de datos de diferentes sensores puede suponer una gran alternativa a la comunidad científica debido principalmente a que todos los datos están integrados espacialmente en una misma estructura de datos, y no en diferentes como ocurre en los formatos convencionales. Esto facilita la localización espacial de mediciones de orígenes diferentes, potenciando y mejorando la precisión de algunas metodologías como la detección de cambios.

Tras un análisis a las imágenes resultantes del experimento, en el cual se fusionaron los datos simulados de los satélites FLEX y Sentinel-3, se puede comprobar que la morfología de la escena cambia bastante respecto a las adquiridas por un único sensor. Mientras que en estos casos la escena solía tener una tendencia rectangular, ahora ésta puede tener formas irregulares de acuerdo a la posición y el tamaño del *swath*, y a la resolución y geometría de los sensores.

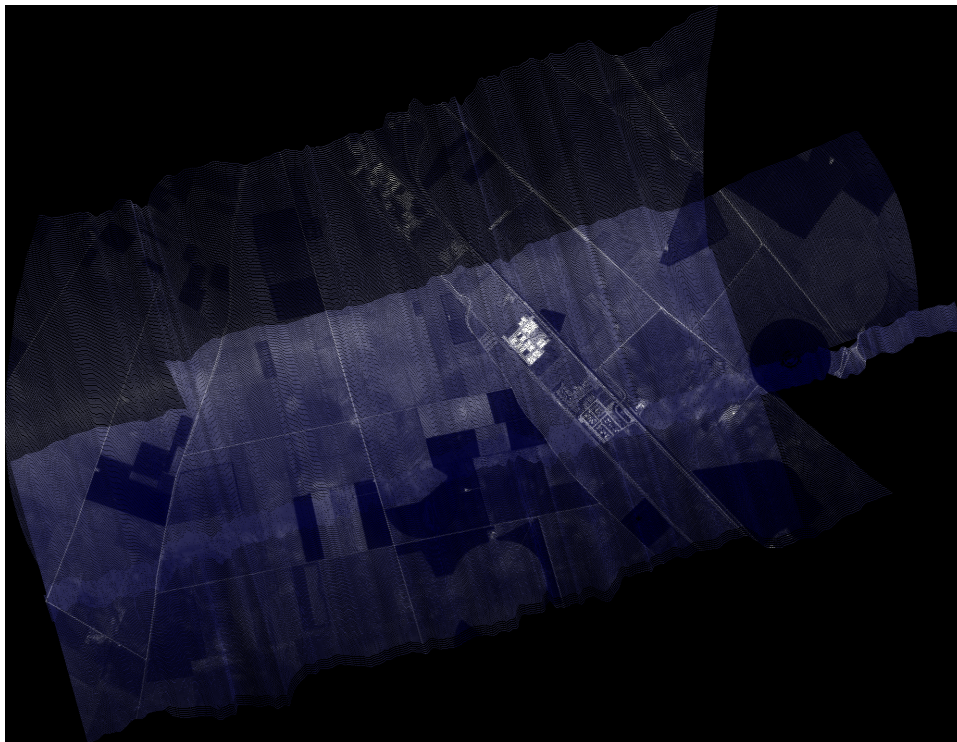
Respecto a la densidad de las celdas DMF, la primera circunstancia posible de comprobar es que tras la fusión de los datos aparecen zonas en las cuales el número de mediciones por celda DMF es mayor, obviamente en aquellas áreas donde los *swaths* de los diferentes orígenes coinciden. Por lo que hay que tener en cuenta la posiciones de los mismos si se quiere hacer un procesamiento o un estudio detallado de la imagen, existiendo la posibilidad de aislar dichas zonas, por ejemplo, considerándolas aparte en matrices DMF individuales.

Otra circunstancia es la manera en la que la densidad de los datos varía si se utilizan diferentes tamaños de celda DMF. Según las diferentes políticas consideradas, la repercusión sobre la estructura de datos es diferente:

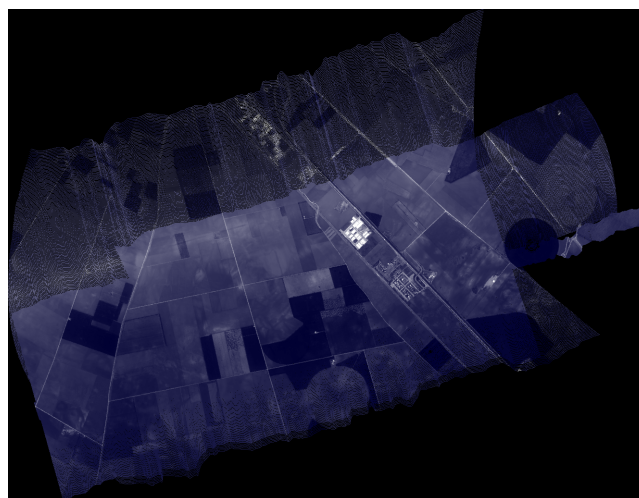
- Si se utiliza un tamaño de celda DMF correspondiente al mayor tamaño de píxel de las imágenes simuladas, la matriz DMF resultante puede ser demasiado pequeña, por lo que la densidad de mediciones puede ser demasiado elevada. Al tener listas de registros DMR más largas, puede repercutir negativamente al procesamiento.



(a)

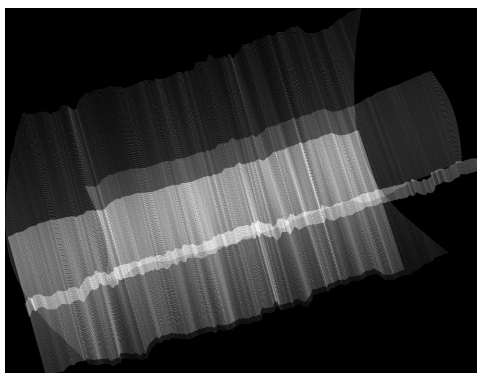


(b)

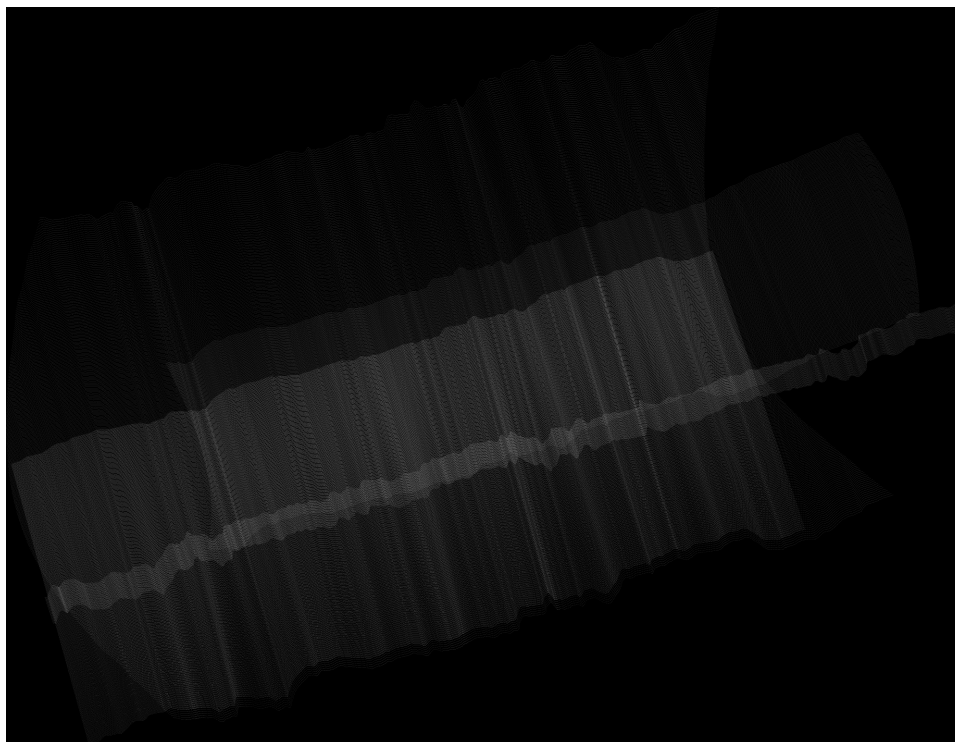


(c)

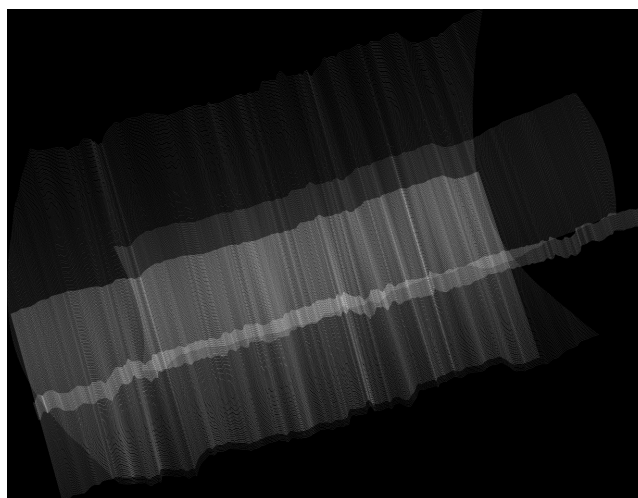
Figura 5.85: Análisis visual de la matriz DMF resultante tras introducir todas las mediciones adquiridas por FIMAS, OLCI, SLSTR_Nadir y SLSTR_Backward.
(a) Utilizando un tamaño de celda DMF máximo [Banda 0]. (b) Utilizando un tamaño de celda DMF mínimo [Banda 0]. (c) Utilizando un tamaño de celda DMF medio [Banda 0].



(a)



(b)



(c)

Figura 5.86: Densidad de mediciones de la matriz DMF resultante tras introducir todas las mediciones adquiridas por FIMAS, OLCI, SLSTR_Nadir y SLSTR_Backward. (a) Utilizando un tamaño de celda DMF máximo. (b) Utilizando un tamaño de celda DMF mínimo. (c) Utilizando un tamaño de celda DMF medio.

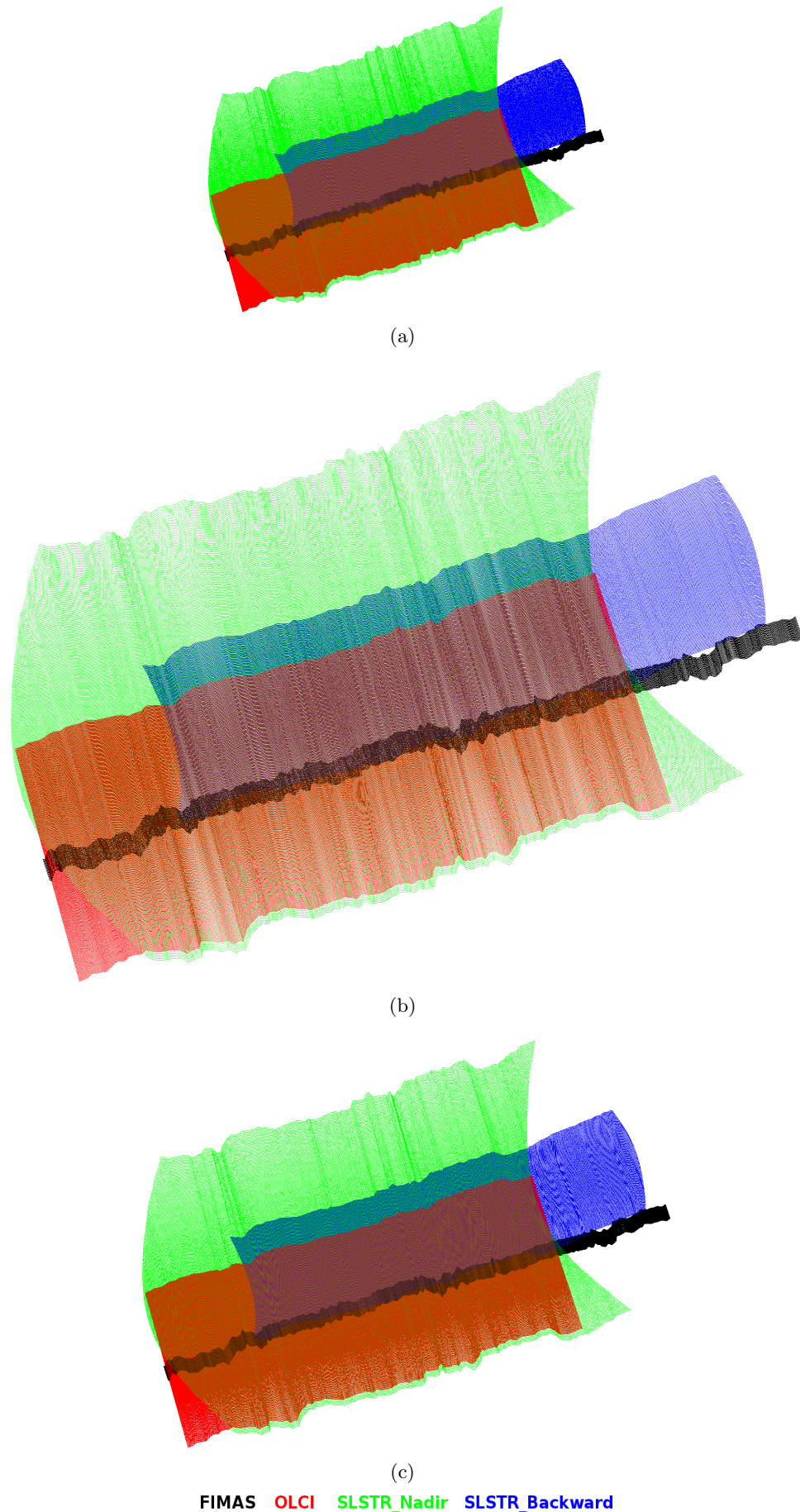


Figura 5.87: Identificación de los orígenes de las mediciones en la matriz DMF resultante tras la fusión de datos.

(a) Utilizando un tamaño de celda DMF máximo. (b) Utilizando un tamaño de celda DMF mínimo. (c) Utilizando un tamaño de celda DMF medio.

- Si se utiliza un tamaño de celda DMF correspondiente al menor tamaño de píxel de las imágenes simuladas, la matriz DMF resultante puede ser demasiado grande, así como la dispersión de los datos, por lo que el número de celdas DMF a *Null* puede ser demasiado elevado. Al tener que dedicar más tiempo a comprobar si una celda tiene medición o no, esto puede repercutir negativamente al procesamiento.
- Si se utiliza un tamaño de celda DMF correspondiente al tamaño de píxel medio de las imágenes simuladas, la matriz DMF tiene un tamaño intermedio, más acorde a la distribución de los datos, por lo que no repercutirá tan negativamente en el procesamiento como los casos anteriores.

5.9. Detección de cambios.

La adaptación del formato DMF para la aplicación de metodologías de detección de cambios supone una gran alternativa al poder almacenar todas las mediciones de diferentes intervalos de tiempo, pero de una misma escena, en una misma estructura de datos. Esto supone una mejor localización de los datos, pues se minimizan problemas derivados de las diferentes geometrías de los sensores implicados y de las diferentes trayectorias y alturas de los vuelos.

5.9.1. *Objetivos del experimento.*

En este experimento el formato DMF fue modificado para poder recibir mediciones adquiridas en diferentes intervalos de tiempo, con el propósito de optimizar el rendimiento frente a metodologías basadas en detección de cambios.

Los objetivos marcados para este estudio se detallan a continuación:

- Adaptación del formato DMF como estructura de datos única para el almacenamiento de mediciones adquiridas en diferentes intervalos de tiempo.
- Análisis del estado de la matriz DMF tras el almacenamiento de mediciones adquiridas en diferente fecha.
- Rendimiento en la detección de cambios respecto a metodologías convencionales.

5.9.2. *Descripción del experimento.*

Este estudio realiza un análisis de los cambios detectados en dos imágenes adquiridas en diferentes instantes referentes a la escena de Cantera. Por tanto, se utilizaron dos imágenes correspondientes a dos vuelos diferentes utilizando el sensor AHS. La primera de ellas es la imagen de Cantera utilizada en todos los experimentos (Figura 5.7(i) y Figura 5.7(j)). De ella, se extrajo una subimagen referencia, la correspondiente a las naves industriales, utilizada y presentada en la Sección 5.6 (Figuras 5.46(b) y 5.46(c)). La segunda imagen, corresponde a un segundo vuelo instantes después sobre la misma escena y bajo las mismas condiciones, y tal y como se hizo con la primera imagen, se recortó a la subescena de las naves industriales (Figura 5.88).

Para comprobar el rendimiento del formato DMF respecto a la detección de cambios, las dos subescenas serán comparadas tanto en sus versiones geocorregidas como en las versiones DMF adaptadas tal y como se presentó en la Sección 5.6. La metodología para realizar dicha comparación se detalla a continuación:

1. A partir de la primera imagen L0 de la subescena de Cantera ($SubCantera1_{L0}$) y del correspondiente fichero IGM, se obtuvo la versión geocorregida ($SubCantera1_{Geo}$) (Figura 5.46(c)).
2. A partir de la segunda imagen L0 de la subescena de Cantera ($SubCantera2_{L0}$) y del correspondiente fichero IGM, se obtuvo la versión geocorregida ($SubCantera2_{Geo}$) (Figura 5.88(b)).
3. Para cada píxel de $SubCantera1_{Geo}$ ($P_SubCantera1_{Geo}$), se localizó el píxel más cercano en $SubCantera2_{Geo}$ ($P_SubCantera2_{Geo}$), y se calculó el ángulo (α) entre el espectro de $P_SubCantera1_{Geo}$ (x) y el espectro de $P_SubCantera2_{Geo}$ (y) tal y como se indica en Ecuación 5.6, presentada en la Sección 5.6.2. El ángulo espectral medio de todos los píxeles de $SubCantera1_{Geo}$ indicará el nivel de cambios localizados para la versión geocorregida.
4. A partir de la primera imagen L0 de la subescena de Cantera ($SubCantera1_{L0}$) y del correspondiente fichero IGM, se obtuvo la versión DMF ($SubCantera_{DMF}$) (Figura 5.46(d)). Para los registros DMR de esta primera imagen se utilizó un $Timestamp = Instante1$.
5. A partir de la segunda imagen L0 de la subescena de Cantera ($SubCantera2_{L0}$) y del correspondiente fichero IGM, se obtuvieron las mediciones correspondientes y fueron también insertadas en la matriz DMF $SubCantera_{DMF}$, junto con los datos de $SubCantera1_{L0}$ (Figura 5.89). Para los registros DMR de esta segunda imagen se utilizó un $Timestamp = Instante2$.
6. Para cada registro DMR de $SubCantera_{DMF}$ con $Timestamp = Instante1$ ($DMR_SubCantera_{Instante1}$), se localizó el registro DMR más cercano con $Timestamp = Instante2$ ($DMR_SubCantera_{Instante2}$) dentro de la misma celda DMF, y se calculó el ángulo (α) entre el espectro de $DMR_SubCantera_{Instante1}$ (x) y el espectro de $DMR_SubCantera_{Instante2}$ (y) tal y como se indica en Ecuación 5.6, presentada en la Sección 5.6.2. El ángulo espectral medio de todas las mediciones de $SubCantera_{DMF}$ con $Timestamp = Instante1$ indicará el nivel de cambios localizados para la versión DMF.

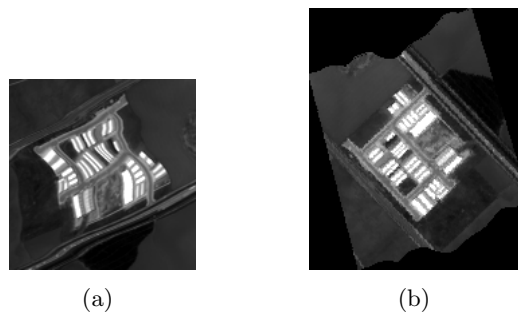


Figura 5.88: Segunda subimagen de Cantera utilizada para el estudio de la detección de cambios.
 (a) Imagen L0 [Banda 0]. (b) Versión geocorregida [Banda 0].



Figura 5.89: Imagen DMF con las todas las mediciones de la primera subimagen ($Timestamp=Instante1$) y de la segunda subimagen ($Timestamp=Instante2$) almacenadas.

A la hora de introducir todos los datos procedentes de las imágenes, fue necesario elegir el tamaño de celda DMF de la matriz DMF resultante, ya que la resolución espacial para el primer vuelo era de 4.4 m, mientras que para el segundo era de 4.1 m. Tal y como se demostró en la Sección 5.8.4, ante esta circunstancia existente en la fusión de datos, la mejor opción es seleccionar un tamaño de celda DMF promedio. Por tanto, la matriz DMF fue construida con un tamaño de celda DMF de 4.25 m. De esta manera, las características de las subimágenes utilizadas y generadas, correspondientes al segundo vuelo, se muestran en la Tabla 5.15.

	[Filas \times Columnas \times Bandas]	TamPíxel
$SubBarrax_{L0}$	[143 \times 139 \times 63]	—
$SubBarrax_{Geo}$	[196 \times 164 \times 63]	4.1 m
$SubBarrax_{DMF}$	[205 \times 162 \times 63]	4.25 m

Tabla 5.15: Descripción de las subimágenes, correspondientes al segundo vuelo, utilizadas y generadas para el estudio de la detección de cambios.

5.9.3. Resultados del experimento.

Tras aplicar la metodología expuesta en la sección anterior, se obtuvieron dos imágenes resultantes. La primera de ellas, representada en la Figura 5.90(a), es el ángulo espectral (α) obtenido para cada píxel de la versión geocorregida del primer vuelo, respecto a su análogo en la versión geocorregida del segundo vuelo. Cuanto más blanco es el píxel, mayor es el ángulo espectral (α) entre ambos y por tanto mayor es la diferencia entre los espectros. Del mismo modo, la segunda imagen (Figura 5.90(b)) corresponde al ángulo espectral (α) para todas los registros DMR almacenadas en la versión DMF con $Timestamp = Instante1$, respecto a sus análogos almacenados en la misma celda DMF con $Timestamp = Instante2$.

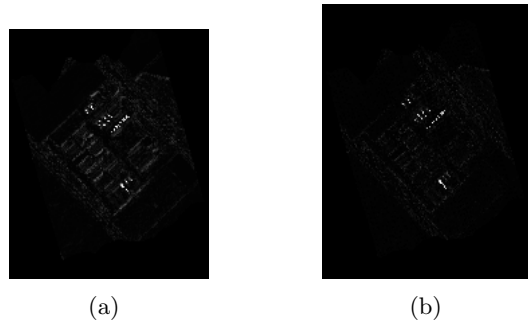


Figura 5.90: Visualización de los cambios detectados para las subescenas de Cantera. (a) Detección de cambios para las imágenes en versión geocorregida. (b) Detección de cambios para las imágenes en versión DMF.

El ángulo espectral medio ($\bar{\alpha}$) obtenido para las versiones geocorregidas y DMF, así como el número de mediciones consideradas en la detección de cambios, para cada caso, se muestran en la Tabla 5.16.

	Nº Mediciones	$\bar{\alpha}$
Geocorregida	15496 med	1.95°
DMF	15256 med	1.43°

Tabla 5.16: Ángulo espectral medio obtenido tras realizar la detección de cambios usando versiones geocorregidas y el formato DMF.

5.9.4. Análisis de los resultados del experimento.

En este apartado se ha adaptado el formato DMF para la fusión de datos temporales con el objetivo de optimizar las metodologías de detección de cambios, obteniendo los siguientes resultados:

1. Fusión temporal de los datos correspondientes a dos vuelos diferentes sobre la escena de Cantera (Figura 5.89).
2. Análisis de las diferencias espectrales para las imágenes geocorregidas (Figura 5.90(a)).
3. Análisis de las diferencias espectrales para las imágenes en formato DMF (Figura 5.90(b)).
4. Cálculo del ángulo espectral medio para las imágenes geocorregidas y en formato DMF (Tabla 5.16).

Tras analizar la figura correspondiente a la detección de cambios usando las versiones DMF de las subimágenes de Cantera (Figura 5.90(b)), es posible apreciar una circunstancia derivada de la búsqueda de mediciones análogas adquiridas en diferentes periodos temporales. Al restringir dicha búsqueda a la vecindad correspondiente a una celda DMF, aparecen determinadas celdas DMF asignadas como *Null*, inexistentes en la matriz DMF original. Éstas corresponden a zonas en las que, o bien no existen registros DMR con $Timestamp = Instante1$, o bien sí existen pero no hay ningún registro DMF con $Timestamp = Instante2$ en la misma celda DMF (cercano). Al ser este último resultado representado mediante una matriz DMF, se muestra el ángulo espectral (α) correspondiente al registro con $Timestamp = Instante1$ más cercano al centro de cada celda DMF.

También es posible comprobar que en ambas imágenes, los cambios detectados son principalmente en las naves industriales del centro de la imagen. Tras analizar los espectros de manera independiente, se pudo comprobar que efectivamente, la mayor diferencia espectral entre ambas adquisiciones se producía en dichas construcciones. Además, tal y como ocurría y se explicaba en la Sección 5.6, los mayores cambios se producen en los bordes de los elementos presentes en la escena, siendo este hecho más pronunciado en la versión geocorregida. Aunque ya se explicaron los motivos en la sección correspondiente, de manera resumida éstos son provocados por el desplazamiento de las mediciones durante el proceso de geocorrección. Como estos desplazamientos son inexistentes en el formato DMF (solamente ocurrirán leves desplazamientos provocados por la incertidumbre del dispositivo GPS/INS), la detección de cambios en los bordes se minimizan en este caso.

Respecto a la Tabla 5.16, la detección de cambios ha sido más precisa para el caso de la versión en DMF, debido principalmente a dos circunstancias:

- Los errores espaciales introducidos durante el proceso de geocorrección hacen que la búsqueda de los píxeles de la primera imagen geocorregida en la segunda no sea del todo precisa. A este hecho hay que sumarle las diferencias entre ambas respecto a la geometría del sensor y a la trayectoria y altura del vuelo.
- La búsqueda en la versión DMF se limita a la misma celda DMF. Esto hace que no se comparen espectros que estén situados a demasiada distancia, decrementando el error global.

Por último, el hecho de tener todos los datos de interés en una misma estructura de datos, en vez de en estructuras individuales, facilita la manera de detectar los cambios, optimizando los resultados de esta metodología y haciendo mucho más cómoda y precisa la generación de resultados.

El siguiente capítulo estudia la posibilidad de utilizar el formato DMF como tipo de datos de salida de un sistema de procesamiento en tiempo real, del cual se presenta, adicionalmente, su diseño e implementación a un nivel de hardware. Dicho sistema de procesamiento se basa en el dispositivo SpaceCube desarrollado por la NASA, compuesto por dos tarjetas FPGA interconectadas, lo que proporciona la funcionalidad de ser completamente reconfigurable y adaptable a cualquier tipo de misión.

Capítulo 6

Resultados: Adaptación del formato DMF a sistemas de adquisición

Es interesante estudiar la manera en la que los sistemas de adquisición proporcionan los resultados, sobre todo cuando disponen de un sistema de procesamiento en tiempo real. El tipo de datos seleccionado es muy importante, ya que influye directamente en la calidad de la representación de la información, su almacenamiento, y su distribución. Por ello, es aconsejable dedicar un porcentaje del diseño del sistema en la definición de una estructura capaz de almacenar todos los datos proporcionados por los dispositivos presentes en la plataforma, incluyendo instrumentos (sensores) y sistema de posicionamiento. El formato definido debe asegurar, además, una correcta sincronización entre todos estos dispositivos así como la capacidad de proporcionar los resultados con frecuencias acordes a las establecidas por la misión, pues en caso contrario pueden aparecer imprecisiones e inconsistencias procedentes de la saturación de los módulos internos.

Debido a que el formato DMF se basa en la medición como unidad básica de procesamiento, la generación de resultados puede simplificarse considerablemente, garantizando una salida del sistema organizada, sincronizada, íntegra y, además, georreferenciada, por lo que la información resultante de las metodologías implementadas se encontrarían aptas para su distribución desde un primer momento.

Este capítulo está compuesto por una serie de experimentos que utilizarán el conjunto de imágenes real presentado en la Sección 5.1, con el objetivo de analizar el comportamiento del formato DMF cuando es integrado como formato de salida en un sistema de computación en tiempo real. Igual que en el capítulo de resultados anterior (Capítulo 5), cada experimento estará dividido en cuatro apartados:

1. *Objetivos del experimento*: Finalidad del experimento. Descripción de los objetivos que se desean alcanzar mediante la elaboración de dicho experimento.
2. *Descripción del experimento*: Desarrollo detallado de la elaboración del experimento, indicando las imágenes utilizadas.
3. *Resultados del experimento*: Resultados obtenidos al aplicar la metodología detallada en el apartado anterior.
4. *Análisis de los resultados del experimento*: Estudio de los resultados obtenidos para comprobar si con la metodología desarrollada se han conseguido cubrir los objetivos propuestos.

Los experimentos realizados abarcan análisis del rendimiento del formato DMF cuando es utilizado como tipo de salida de un sistema de procesamiento en tiempo real, teniendo como referencia principal el dispositivo basado en chips FPGA SpaceCube. Estos experimentos pueden ser clasificados en tres grupos diferentes:

1. *Estudio de la viabilidad del uso del formato DMF como tipo de salida*: Análisis de cómo influye el hecho de considerar el formato DMF como tipo de salida en un sistema de computación en tiempo real. El experimento realizado (Sección 6.1) estudia la optimización del tráfico de datos entre satélite - Tierra cuando el formato DMF se adapta a metodologías reales concretas, como la detección de nubes.
2. *Primeras aproximaciones al dispositivo SpaceCube*: Está constituido por dos experimentos, presentados en la Sección 6.2 y en la Sección 6.3, que tienen como objetivo estudiar el funcionamiento de las tarjetas FPGA Xilinx Virtex-5, utilizadas por el dispositivo SpaceCube, mediante la optimización de las comunicaciones de los módulos internos.
3. *Emulación de un módulo de procesamiento SpaceCube*: Desarrollo de un sistema de computación análogo al módulo de procesamiento SpaceCube, compuesto por la interconexión de dos tarjetas FPGA Xilinx Virtex-5. Mientras que experimentos anteriores se centraban en la optimización de las comunicaciones internas entre los componentes de las tarjetas FPGA, el experimento presentado en la Sección 6.4 analiza los diferentes puertos y conexiones de entrada / salida al sistema general y estudia la eficiencia máxima de los protocolos (con diferentes tasas de velocidad) cuando se considera el formato DMF como tipo de datos de salida.

6.1. Procesamiento a bordo en tiempo real.

El formato DMF ha sido diseñado como una estructura de datos capaz de almacenar todas las mediciones recogidas por los sensores siempre considerando la localización espacial exacta de los datos. En todos los experimentos anteriores, se partía de la construcción de una matriz DMF a partir de unos ficheros L0 e IGM ya generados y existentes en disco. Sin embargo, este formato supone también una gran alternativa para el procesamiento en tiempo real, capaz de procesar las mediciones según van siendo capturadas, y transformarlas posteriormente al formato DMR.

6.1.1. Objetivos del experimento.

El formato DMF, al estar basado en una estructura de datos cuya unidad básica es el dato adquirido por el sensor, presenta la ventaja de poder ser procesada a un nivel de medición. Esto posibilita el análisis y el procesamiento de cada medición, generando como salida el resultado de la operación junto con las coordenadas espaciales, en un formato DMR. Por lo tanto, al tratarse de una estructura tan ligera y sencilla, se convierte en potencial en sistemas que requieran procesamiento a bordo y en tiempo real, abriendo la posibilidad de que los resultados sean enviados a Tierra en dicho formato, o bien almacenados en memorias internas.

En este experimento se estudiará el rendimiento del formato DMF cuando es utilizado como estructura de datos para procesamiento a bordo en tiempo real, por lo que se han definido los siguientes objetivos:

- Estudiar la posibilidad de integrar la estructura de datos (software) en un sistema de adquisición (hardware).
- Diseñar de manera teórica un sistema de adquisición de datos cuya unidad básica de procesamiento sea el registro DMR.
- Analizar el rendimiento global y localizar las situaciones en las que el formato DMF supone una ventaja.

6.1.2. Descripción del experimento.

Para comprobar el rendimiento del formato DMF para un caso teórico en el cual se realiza un procesamiento a bordo en tiempo real, se realizó un estudio sobre cómo podrían descartarse mediciones en las que las nubes estuvieran presentes, directamente desde el espacio. Para algunos campos científicos, esta información es completamente inútil, por lo que se consume un alto porcentaje del tráfico satélite - Tierra enviando datos que no se van a utilizar.

De esta manera, sería posible implementar una metodología software / hardware que, basándose en diferentes características de la medición, pudiera decidir si ésta es considerada o descartada. El formato de datos utilizado para el envío sería DMF, transfiriendo únicamente los registros DMR considerados como válidos (en las que no se han detectado presencia de nubes) uno a uno. Esta forma de transmisión es exactamente igual a la forma en la que los datos en el formato DMF son almacenados en disco (Sección 4.1.1.3), por lo que es posible el salvado de la información, así como de la visualización y procesamiento de las mediciones adquiridas de forma directa mediante la matriz DMF, agilizando el proceso de preparación y distribución a la comunidad científica.

El diseño teórico de este modelo se resume y presenta de manera gráfica en la Figura 6.1. Básicamente, estaría constituido por una unidad de procesamiento software o hardware que implementa la metodología de detección de nubes. Para ello, se valdría de la siguiente información:

- Imagen RAW: Datos adquiridos por los sensores en formato imagen, o bien en formato medición individual. Serán la fuente de la metodología, y podrán ser descartados o considerados por el algoritmo de detección de acuerdo al nivel de presencia de nubes en los mismos.
- Información del Sistema de Posicionamiento: Información adicional que permita localizar espacialmente las mediciones sobre la escena (por ejemplo, latitud y longitud en un determinado sistema cartográfico).
- Algoritmo de Detección de Nubes: Implementado a nivel de hardware o software, permite seleccionar entre las mediciones útiles de acuerdo a la presencia de nubes en las mismas.
- Estación en Tierra (*Ground-Station*): Recibe los datos válidos del satélite como secuencia de registros DMR, y los salva en el sistema de almacenamiento de manera directa en formato DMF. De esta manera el acceso a la información puede estar disponible casi en el acto, al menos a niveles bajos de procesamiento.

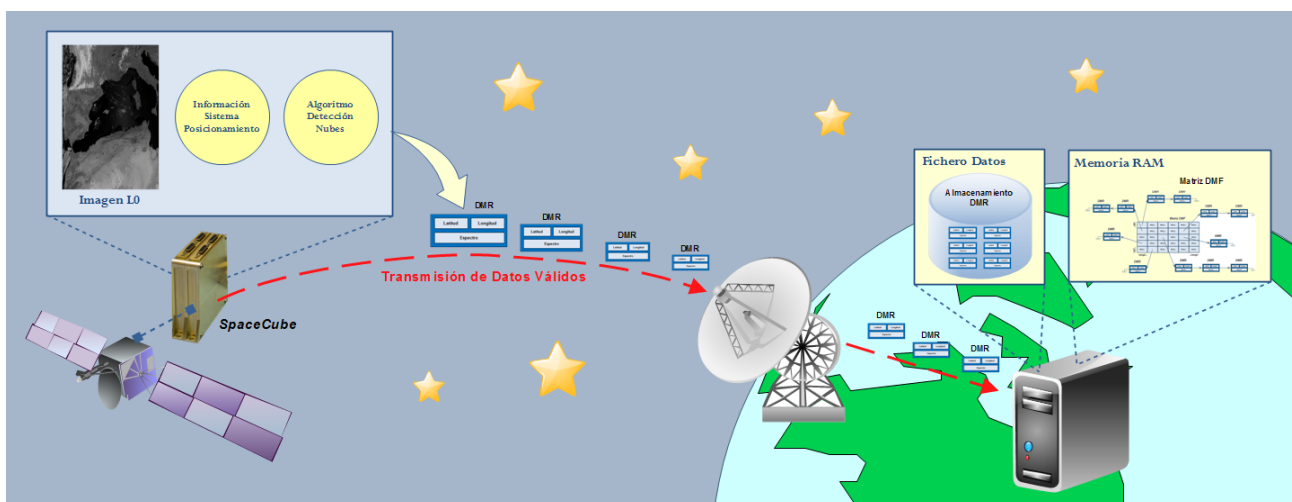


Figura 6.1: Diseño del sistema de detección de nubes a bordo de satélite, con el formato DMF como formato de datos básico.

Uno de los sistemas potenciales para poder implementar esta metodología es SpaceCube, gracias a la posibilidad de poder ser programado tanto a un nivel software como hardware, y por la facilidad que aporta a la hora de poder distribuir la información en el formato DMF.

Para la realización de este experimento, se ha utilizado la imagen de Mediterráneo (Figura 5.7(e)), adquirida de manera real por un satélite (NASA/MODIS) y de la cual se disponía información adicional para poder llevarlo a cabo, como una máscara de nubes asociada. Esta máscara de nubes asociada (Producto MOD35_L2) [Hub12, Pla03] asigna a cada medición de la imagen L0 uno de cuatro niveles posibles de presencia de nubes:

- *Nivel 0*: Con nube.
- *Nivel 1*: Probablemente con nube.
- *Nivel 2*: Probablemente sin nube.
- *Nivel 3*: Sin nube.

Toda esta información referente a la máscara de presencia de nubes, junto con los ficheros L0 e IGM, permitirán realizar un estudio completo de la reducción de tráfico cuando una metodología de detección de nubes es implementada a bordo del satélite, utilizando el formato DMF.

6.1.3. Resultados del experimento.

A través de los ficheros L0 e IGM, se construyeron cuatro tipos de imágenes DMF diferentes, de acuerdo a los niveles de presencia de nubes definidos por la máscara:

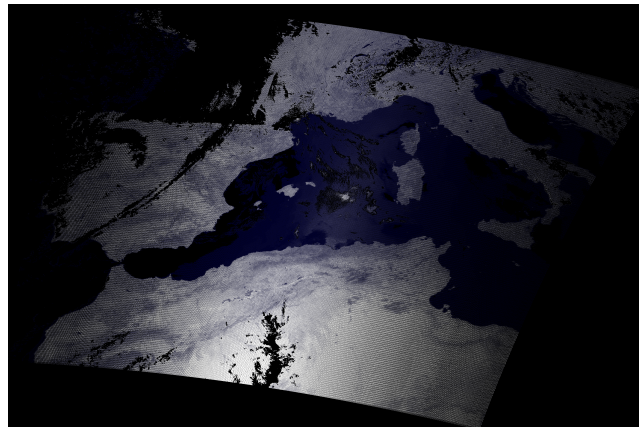
- DMF_{Or} (Figura 5.9(c)): Imagen en formato DMF original, sin eliminar ninguna medición por presencia de nubes.
- DMF_{N0} (Figura 6.2(a)): Imagen en formato DMF, en la que se han eliminado las mediciones correspondientes a un Nivel 0 (nubes con seguridad).
- DMF_{N1} (Figura 6.2(b)): Imagen en formato DMF, en la que se han eliminado las mediciones correspondientes a un Nivel 0 y 1 (nubes con seguridad o alta probabilidad).
- DMF_{N2} (Figura 6.2(c)): Imagen en formato DMF, en la que se han eliminado las mediciones correspondientes a un Nivel 0, 1 y 2 (nubes con seguridad o alta o baja probabilidad).

El tamaño de la información útil, tras eliminar aquéllas mediciones inválidas con presencia de nubes a diferentes niveles, fue comparada respecto a la información completa original, tanto a la hora de transmitir y almacenar los datos como a la hora de representar y trabajar con la matriz DMF en memoria RAM. El tamaño de un registro DMR (T_{DMR}) está definido por:

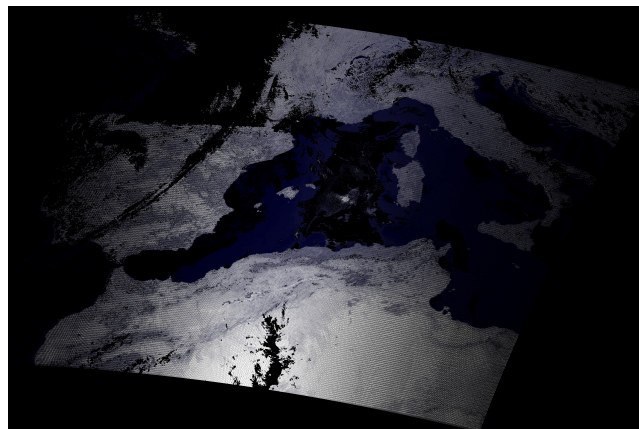
$$T_{DMR} = (N_{Bandas} \times T_{Datos}) + (2 \times T_{Coord}) \quad (6.1)$$

Donde N_{Bandas} es el número de bandas de la imagen, T_{Datos} , es el tamaño de un valor de espectro, y T_{Coord} es el tamaño de un valor de coordenadas.

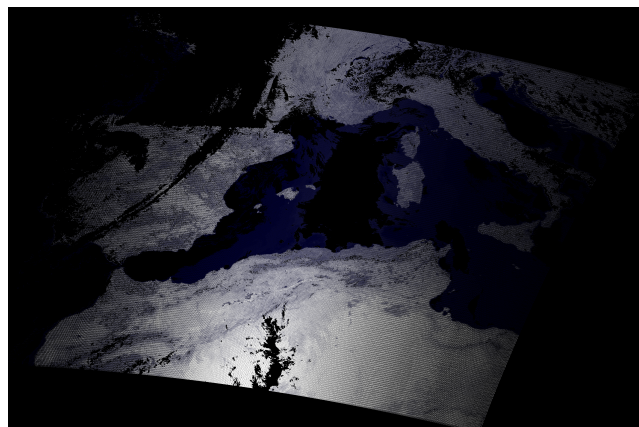
De esta manera, el tamaño de los datos en formato DMF al ser transmitidos ($T_{Transferencia}$) o salvados en disco (T_{Disco}), y el tamaño en memoria RAM al construir la matriz DMF (T_{RAM}) quedarán definidos, respectivamente, por:



(a)



(b)



(c)

Figura 6.2: Visualización de la imagen Mediterráneo en formato DMF tras considerar diferentes niveles de presencia de nubes.

(a) Con eliminación de nubes a un Nivel 0 (DMF_{N_0}). (b) Con eliminación de nubes a un Nivel 0 y 1 (DMF_{N_1}). (c) Con eliminación de nubes a un Nivel 0, 1 y 2 (DMF_{N_2}).

$$T_{Transferencia} = T_{Disco} = N_{Mediciones} \times T_{DMR} \quad (6.2)$$

$$T_{RAM} = N_{Filas} \times N_{Columnas} \times T_{Sistema} + N_{Mediciones} \times T_{DMR} \quad (6.3)$$

Donde $N_{Mediciones}$ es el número de mediciones consideradas, N_{Filas} y $N_{Columnas}$ el número de filas y columnas de la matriz DMF resultante, y $T_{Sistema}$ el número de bytes del sistema.

En el caso de las imágenes geocorregidas análogas, el tamaño resultante (T_{Geo}) queda definido por:

$$T_{Geo} = N_{Filas} \times N_{Columnas} \times N_{Bandas} \times T_{Datos} \quad (6.4)$$

Donde N_{Filas} y $N_{Columnas}$ es el número de filas y columnas de la imagen geocorregida.

Una comparación de los tamaños de transmisión y almacenado, así como los de memoria RAM, para las imágenes en formato DMF considerando los diferentes niveles de presencia de nubes, se muestran en la Tabla 6.1. En ella se muestran el tamaño del registro DMR, el tamaño de la imagen geocorregida correspondiente (presentada en la Figura 5.7(f)), y los tamaños de las imágenes DMF para los diferentes niveles de presencia de nubes, tanto en la transferencia ($T_{Transferencia}$), como en el salvado en disco (T_{Disco}), como en el formato matricial para procesamiento en memoria RAM (T_{RAM}). Para el caso de la transferencia, se muestra la reducción obtenida respecto al envío de todas las mediciones sin realizar detección de nubes a bordo del satélite. Para el caso del almacenamiento en disco, se muestra la reducción obtenida respecto a la versión geocorregida de la imagen (T_{Geo}). De la misma manera, para el caso del procesamiento de la información en la memoria RAM, también se muestra la reducción en el tamaño de la matriz DMF resultante respecto a la geocorregida.

T_{DMR}		
Tamaño		
$22 \times 2B + 2 \times 4B = 52B$		
T_{Geo}		
Tamaño		
$2278 \times 3416 \times 22 \times 2B = 326.53MB$		
$T_{Transferencia}$		
Tamaño		Reducción
DMF_{Or}	$2748620 \times 52B = 136.31MB$	0 %
DMF_{N0}	$2256889 \times 52B = 111.92MB$	17.89 %
DMF_{N1}	$2004276 \times 52B = 99.39MB$	27.09 %
DMF_{N2}	$1935624 \times 52B = 95.99MB$	29.58 %
T_{Disco}		
Tamaño		Reducción
DMF_{Or}	$2748620 \times 52B = 136.31MB$	58.25 %
DMF_{N0}	$2256889 \times 52B = 111.92MB$	65.72 %
DMF_{N1}	$2004276 \times 52B = 99.39MB$	69.56 %
DMF_{N2}	$1935624 \times 52B = 95.99MB$	70.6 %
T_{RAM}		
Tamaño		Reducción
DMF_{Or}	$2279 \times 3417 \times 4B + 2748620 \times 52B = 166.01MB$	49.16 %
DMF_{N0}	$2279 \times 3417 \times 4B + 2256889 \times 52B = 141.63MB$	56.63 %
DMF_{N1}	$2279 \times 3417 \times 4B + 2004276 \times 52B = 129.1MB$	60.46 %
DMF_{N2}	$2279 \times 3417 \times 4B + 1935624 \times 52B = 125.7MB$	61.5 %

Tabla 6.1: Tamaños de la información adquirida por el satélite de acuerdo al nivel de presencia de nubes, incluyendo la reducción obtenida cuando el formato DMF se ha utilizado.

6.1.4. *Análisis de los resultados del experimento.*

En esta sección, se ha analizado el comportamiento del formato DMF como estructura de datos orientado al procesamiento a bordo, utilizando como metodología concreta la detección de nubes, y obteniendo los siguientes resultados:

1. Visualización de los datos tras eliminar las mediciones con presencia de nubes a diferentes niveles (Figura 6.2).
2. Comparación de los diferentes tamaños DMF, respecto a la versión geocorregida, tras eliminar las mediciones con presencia de nubes a diferentes niveles (Tabla 6.1).

El formato DMF puede ser una gran alternativa si se utiliza como estructura de datos básica a la hora de realizar computación en tiempo real a bordo de sistemas de procesamiento en aviones o satélites. Debido principalmente a su sencillez y a su fácil integración, es posible obtener directamente un producto de alto nivel nada más aterrizar el avión (en donde los registros DMR estarían almacenados en una memoria interna o externa), o nada más recibir los datos a Tierra si se trata de un satélite. Esto proporciona una mayor fiabilidad una reducción del tiempo de espera a la hora de ofrecer los datos a la comunidad científica, los cuales pueden acceder a la información adquirida en el mismo instante de la recepción de los datos, siempre que no se requiera post-procesamiento.

Los resultados muestran que el formato DMF y los sistemas de computación a bordo son una combinación que puede ofrecer grandes posibilidades. El hecho de ligar cada medición a una localización espacial permite:

- El procesamiento a bordo de la nave para la obtención de unos resultados ya georreferenciados justo en el momento de recepción de los datos. Junto a las mediciones originales, sería posible presentar resultados de clasificación de imágenes, mapas, máscaras, flags, etc, como nuevos campos en el registro DMR, tal y como se detalló en el apartado de Fusión de Datos (Sección 4.1.2.1).
- La elaboración de sistemas de monitorización en tiempo real más eficientes. Sería posible enviar registros DMR especiales que indicaran de diferentes eventos detectados en Tierra, por ejemplo fuegos, vertidos en el mar, etc. Al estar cada registro DMR perfectamente localizado mediante coordenadas, la recepción consistiría en una secuencia de mediciones indicando un problema y su posición en el planeta.

De esta manera, no sólo el acceso a los datos estarían disponibles en el momento de recepción de los mismos, sino también resultados previos que pueden ser utilizados directamente, o bien como refinamiento de otros posteriores.

6.2. Primera aproximación a SpaceCube: El periférico MEDiC.

El sistema SpaceCube, tal y como se comentó en la Sección 3.3.2.1, es un dispositivo desarrollado por la NASA para ser utilizado como unidad de procesamiento, o como sistema auxiliar, cuya principal característica es la capacidad de ser reprogramado. Esto se consigue mediante la utilización de dos tarjetas FPGA Xilinx Virtex-5 interconectadas, que pueden albergar varias metodologías o algoritmos, capaces de ser ejecutados de manera paralela y a un nivel de hardware.

6.2.1. *Objetivos del experimento.*

La programación de tarjetas FPGA es un campo complicado, en el que es difícil introducirse y cuya curva de aprendizaje es bastante lenta [McD08]. El principal motivo de esto son los altos conocimientos previos requeridos de diseño, análisis e implementación tanto a un nivel de hardware como a un nivel de software. Este experimento sirve como introducción a la programación de las tarjetas FPGA Xilinx Virtex-5, mediante el desarrollo de operaciones simples que resuelvan problemas complejos, o bien redundantes, de metodologías más avanzadas, y que permitan su compatibilidad con la arquitectura SpaceCube, y con la finalidad de ofrecer la posibilidad de ser ejecutadas a bordo de la nave. Por tanto, los objetivos que se proponen para este experimento son los siguientes:

- Familiarización con la programación de tarjetas FPGA mediante el desarrollo de un cálculo hardware simple que permita resolver problemas característicos de una metodología más compleja.
- Estudio de las conexiones entre los componentes, módulos y periféricos implicadas en el sistema propuesto.
- Comparativa del speed-up conseguido por una metodología implementada en hardware respecto a su versión implementada en software.

6.2.2. *Descripción del experimento.*

Para favorecer el desarrollo de metodologías de procesamiento a bordo de aviones o satélites, es importante la creación de dispositivos auxiliares que permitan liberar al Sistema de Computación global de tareas tediosas o repetitivas, cuidando principalmente que éstas sean ligeras de tamaño y rápidas. Por ello, un sistema basado en hardware reconfigurable es una gran opción. Para este experimento, se ha implementado un circuito en la tarjeta FPGA Xilinx Virtex-5 capaz de calcular y determinar la distancia Euclídea entre dos espectros. Esta operación queda determinada mediante:

$$d_E(A, B) = \sqrt{\sum_{i=1}^n (a_i - b_i)^2} \quad (6.5)$$

Donde los espectros A y B están representados por los componentes $A = [a_1, a_2, a_3, \dots, a_n]$ y $B = [b_1, b_2, b_3, \dots, b_n]$.

La distancia Euclídea multi-espectral se trata de un cálculo que requiere un consumo de tiempo elevado y que es ampliamente usado por metodologías ejecutadas a bordo de la nave, especialmente aquéllas dedicadas al procesamiento de imágenes hiperespectrales, por lo que el hecho de implementarla a un nivel de hardware permitiría disminuir los tiempos de ejecución de un gran número de algoritmos comunes y convencionales. Una de las metodologías que se pueden beneficiar del circuito desarrollado, y la cual fue la principal motivación de este experimento, es el algoritmo de segmentación de imágenes Hierarchical Segmentation (HSEG) [Til02, Gua02]. Este algoritmo se basa en la división espacial de la escena representada en la imagen hiperespectral y el cálculo recursivo de distancias Euclídea multi-espectrales, las cuales representan más del 90% del tiempo de ejecución total [Til03]. El desarrollo de un circuito en la tarjeta FPGA Xilinx Virtex-5 supondría una integración directa y una máxima compatibilidad con el sistema SpaceCube, por lo que podría ser utilizada a bordo de un avión o satélite para generar resultados en tiempo real.

Para el cálculo de la distancia Euclídea multi-espectral dos circuitos diferentes fueron implementados en una tarjeta FPGA. Ambas son el resultado de un diseño capaz de realizar este tipo de cálculo, pero en dos versiones alternativas: una secuencial y otra paralela. Para ello, se utilizó un

esquema hardware como el presentado en la Sección 4.2 (Figura 6.3), incluyendo procesador PowerPC y el periférico que albergará el circuito encargado de realizar la distancia Euclídea multi-espectral (Multi-Spectral Euclidean Distance Calculator, MEDiC), ambos conectados por el bus central PLBv46 junto con otros periféricos, como el encargado de las comunicaciones con el exterior mediante el protocolo RS-232. Sin embargo, este último periférico únicamente estará destinado a enviar el resultado obtenido a través del conector para su posterior visualización en una terminal (en una computadora externa).

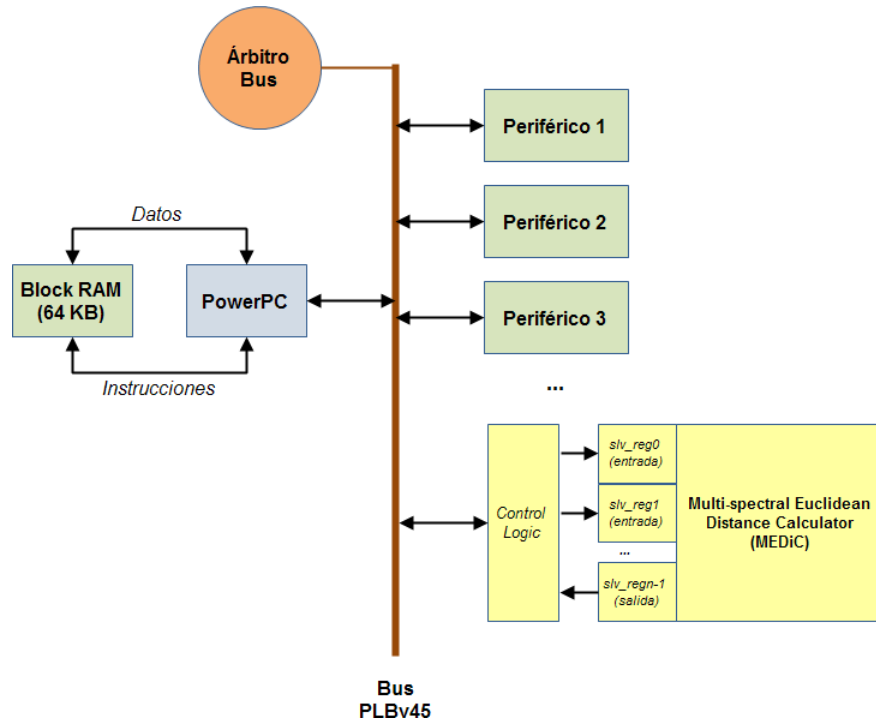


Figura 6.3: Diagrama del sistema hardware con el periférico MEDiC adjunto.

Como se comentó en la Sección 4.2, cada periférico hardware presenta una serie de registros de comunicaciones que permitirá la transferencia de datos con el procesador (y otros periféricos) a través del bus central. En este caso, los registros permitirán la recepción de los dos espectros A y B de la cual se desea realizar el cálculo de la distancia Euclídea multi-espectral, el resultado de la operación, y otras señales de sincronización, como Reset (reseteo del circuito), Enable (habilitación para que comience el cálculo, tras el almacenamiento y obtención de los espectros por parte del periférico) y Done (el cálculo ya se ha realizado y el resultado se encuentra disponible en el registro correspondiente).

El módulo Sequential MEDiC.

El módulo Sequential MEDiC realiza el cálculo de la distancia Euclídea multi-espectral entre dos espectros de manera secuencial (Figura 6.4(a)). Sólo puede realizar una operación $(a_i - b_i)^2$ por ciclo de reloj, por lo que necesitará n iteraciones completas (siendo n el número de componentes de cada uno de los espectros) para determinar el resultado.

Dicho módulo está compuesto por los siguientes submódulos:

- Un restador (Subtractor) que realiza la operación $a_i - b_i$.
- Un multiplicador (Multiplier) que realiza la operación $(a_i - b_i)^2$.
- Un acumulador (Accumulator) que almacena las sumas $\sum_{i=1}^n (a_i - b_i)^2$, en cada una de las n iteraciones necesarias.

- Tres latches que retrasarán la señal durante dos ciclos de reloj. La señal E indicará que un nuevo componente de los espectros A y B ya han sido almacenados en sus registros correspondientes (se activará justo cuando el valor correspondiente de B haya sido almacenado en el registro *slv_reg1*). Tras dos ciclos de reloj (equivalente a la ejecución del restador y del multiplicador), se activará el acumulador y sumará el resultado a los obtenidos en iteraciones anteriores. Si no se hubieran añadido dichos latches, el acumulador estaría realizando sumas de todo lo que le llegara como entrada, incluyendo información inválida entre iteración e iteración.
- El módulo SQRT (raíz cuadrada) que realiza la operación $\sqrt{\sum_{i=1}^n (a_i - b_i)^2}$ con una latencia de 17 cr. Para ello, hace uso del algoritmo CORDIC [And93, Xil11f].

El número de ciclos de reloj requeridos por el módulo MEDiC para realizar el cálculo de la distancia Euclídea multi-espectral de manera secuencial (S_MEDiC_{cc}), para dos espectros de n bandas, es definido por:

$$S_MEDiC_{cc} = (SUB_{cc} + MUL_{cc} + ACC_{cc}) + (n - 1) + SQRT_{cc} = 3 + n - 1 + 17 = n + 19 \quad (6.6)$$

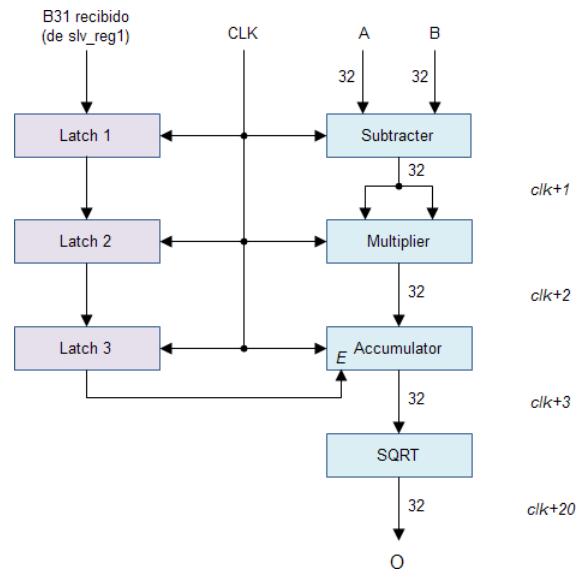
Donde SUB_{cc} , MUL_{cc} , ACC_{cc} y $SQRT_{cc}$ son los ciclos de reloj requeridos por los submódulos restador, multiplicador, acumulador y SQRT, respectivamente.

El módulo Parallel MEDiC.

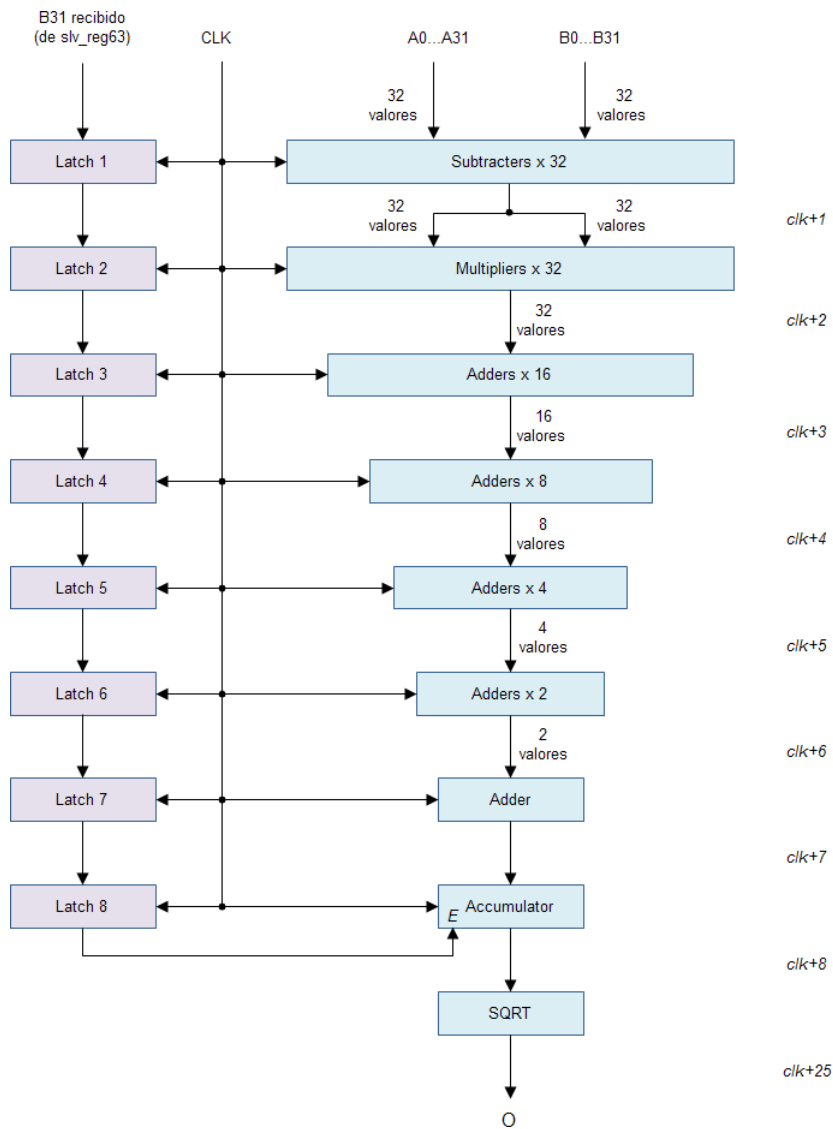
El módulo Parallel MEDiC realiza el cálculo de la distancia Euclídea multi-espectral entre dos espectros de manera paralela (Figura 6.4(b)). Es capaz de procesar un total de 32 operaciones $(a_i - b_i)^2$ por ciclo de reloj, por lo que necesitará $\lceil n/32 \rceil$ iteraciones completas (siendo n el número de componentes de cada uno de los espectros) para determinar el resultado.

Dicho módulo está compuesto por los siguientes submódulos:

- 32 restadores (Subtracters) que realizan 32 operaciones $a_i - b_i$ en paralelo.
- 32 multiplicadores (Multipliers) que realizan 32 operaciones $(a_i - b_i)^2$ en paralelo.
- 5 niveles de sumadores que realizan el sumatorio $\sum_{i=iter_ini}^{iter_fin} (a_i - b_i)^2$ de manera paralela y escalada (16 sumadores en el primer nivel, 8 en el segundo nivel, 4 en el tercer nivel, 2 en el cuarto nivel, y 1 en el quinto nivel).
- Un acumulador (Accumulator) que almacena las sumas $\sum_{i=1}^n (a_i - b_i)^2$, en cada una de las n iteraciones necesarias.
- 8 latches que retrasarán la señal durante siete ciclos de reloj. La señal E indicará que 32 nuevos componentes de los espectros A y B ya han sido almacenados en sus registros correspondientes (se activará justo cuando el valor 31 correspondiente de B haya sido almacenado en el registro *slv_reg63*). Tras siete ciclos de reloj (equivalente a la ejecución de los restadores, multiplicadores y sumadores), se activará el acumulador y sumará el resultado a los obtenidos en iteraciones anteriores. Si no se hubieran añadido dichos latches, el acumulador estaría realizando sumas de todo lo que le llegara como entrada, incluyendo información inválida entre iteración e iteración.
- El módulo SQRT (raíz cuadrada) que realiza la operación $\sqrt{\sum_{i=1}^n (a_i - b_i)^2}$ con una latencia de 17 cr. Para ello, también hace uso del algoritmo CORDIC.



(a)



(b)

Figura 6.4: Circuitos diseñados para el cálculo de la distancia Euclídea multi-espectral.
 (a) Versión secuencial. (b) Versión paralela.

El número de ciclos de reloj requeridos por el módulo MEDiC para realizar el cálculo de la distancia Euclídea multi-espectral de manera paralela (P_MEDiC_{cc}), para dos espectros de n bandas, es definido por:

$$\begin{aligned} P_MEDiC_{cc} &= (SUBS_{cc} + MULS_{cc} + ADDS_{cc} + ACC_{cc}) + \left(\left\lceil \frac{n}{32} \right\rceil - 1 \right) + SQRT_{cc} = \\ &= 8 + \left\lceil \frac{n}{32} \right\rceil - 1 + 17 = \left\lceil \frac{n}{32} \right\rceil + 24 \end{aligned} \quad (6.7)$$

Donde $SUBS_{cc}$, $MULS_{cc}$, $ADDS_{cc}$, ACC_{cc} y $SQRT_{cc}$ son los ciclos de reloj requeridos por los submódulos restadores, multiplicadores, sumadores, acumulador y SQRT, respectivamente.

6.2.3. Resultados del experimento.

Las dos versiones del circuito que realiza el cálculo de la distancia Euclídea multi-espectral fue desarrollado e implementado sobre una sola tarjeta FPGA Xilinx Virtex-5 ML507 usando el entorno Xilinx XPS 12.2. Fue necesario realizar esta tarea a dos diferentes niveles:

1. Desarrollo del hardware: Desarrollo y programación de los circuitos Sequential MEDiC y Parallel MEDiC presentados, los cuales son albergados en la FPGA.
2. Desarrollo del software: Desarrollo y programación de la aplicación que será ejecutada por el procesador PowerPC, éste funcionando a una frecuencia de 400 MHz. Consistirá en la generación de dos vectores (espectros) de 64 elementos (bandas) con valores aleatorios. Debido al diseño de los circuitos, se requerirán 64 iteraciones para procesar todas las bandas para la versión secuencial, mientras que la versión paralela requerirá sólo 2 iteraciones. La aplicación software fue escrita en lenguaje C.

La Figura 6.5 muestra el proceso de ejecución de una iteración para las versiones secuencial y paralela.

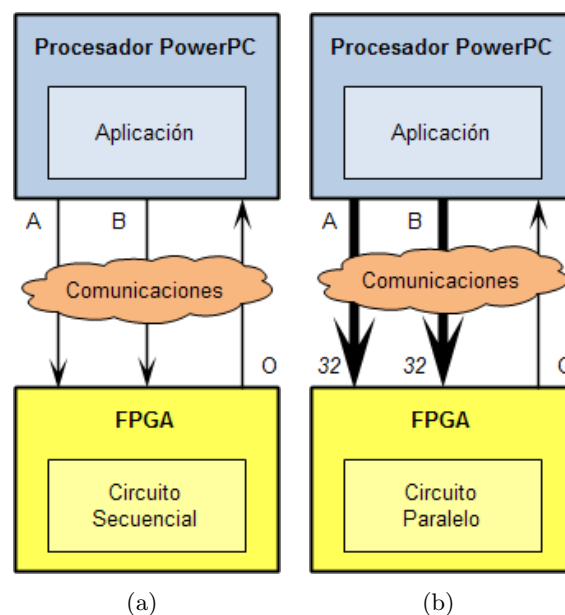


Figura 6.5: Procesos de ejecución.
(a) Caso secuencial. (b) Caso paralelo.

Además, por otro lado, y como comparativa, se realizó el cálculo de la distancia Euclídea multi-espectral por el procesador PowerPC de manera independiente (completamente a un nivel de software, sin utilizar componentes hardware de ningún tipo). Para ello, y de la misma manera que para el estudio de los tiempos de ejecución de los circuitos, el procesador generará de manera aleatoria dos vectores (espectros) de 64 elementos (bandas), de los cuales determinará el ya comentado cálculo. Como la secuencia de instrucciones a ejecutar por el procesador debe ser escrita en lenguaje C, existen diferentes sentencias para realizar la operación $(a_i - b_i)^2$, cuyos tiempos de ejecución puede variar de acuerdo a la sentencia seleccionada:

- $(A[i]-B[i])*(A[i]-B[i])$: En este caso, el tiempo de ejecución total (restas, multiplicaciones y raíz cuadrada) fue de 6545 cr.
- $pow(A[i]-B[i], 2)$: En este caso, el tiempo de ejecución total (restas, multiplicaciones y raíz cuadrada) fue de 314445 cr.

Los mejores resultados fueron obtenidos mediante la sentencia $(A[i]-B[i])*(A[i]-B[i])$, por lo que dicho tiempo de ejecución, a nivel de software, fue considerado para realizar una comparación respecto a las versiones hardware (circuitos) y calcular el speed-up.

La Tabla 6.2 muestra los tiempos de ejecución obtenidos para las dos versiones hardware desarrolladas, incluyendo el speed-up obtenido respecto a la versión de software, previamente detallada.

	Secuencial	Paralelo
<i>Tiempos ejecución (circuito)</i>	83 cr	26 cr
<i>Tiempos ejecución (total)</i>	4357 cr	4421 cr
<i>Tiempo dedicado a comunicaciones</i>	98.1 %	99.412 %
<i>Speed-up</i>	6545/4357=1.502X	6545/4421=1.48X

Tabla 6.2: *Tiempos empleados y speed-up de los circuitos secuencial y paralelo en el cálculo de la distancia Euclídea multi-espectral para espectros de 64 bandas.*

6.2.4. Análisis de los resultados del experimento.

En este estudio se han desarrollado dos circuitos que realizan un cálculo simple y muy común en las metodologías convencionales ejecutadas a bordo de un avión o satélite: la distancia Euclídea multi-espectral. Ambos circuitos, que realizan la misma operación, corresponden a dos representaciones diferentes, la secuencial y la paralela, que fueron añadidos como un periférico denominado MEDiC a un sistema compuesto por un procesador y un bus central, a partir del cual se realizan todas las comunicaciones.

Los resultados obtenidos fueron representados en la Tabla 6.2, que muestra los tiempos de ejecución de ambas versiones de los circuitos, junto con el porcentaje de tiempo dedicado a las comunicaciones (transferencia de los espectros del procesador al periférico MEDiC, y del resultado del periférico MEDiC al procesador) es posible considerar dos determinadas circunstancias.

En un primer lugar, el total de ciclos de reloj dedicados al cálculo de las distancia Euclídea multi-espectral es mucho mayor que el requerido por los circuitos. Esto es debido al tiempo empleado por el bus central en transferir los espectros al periférico MEDiC, así como el resultado final del cálculo. Aunque no es el caso, en sistemas más complejos varios periféricos podrían estar conectados también a dicho bus central, por lo que cada vez que fuera necesario transferir un dato, habría que esperar a que el árbitro del bus le diera prioridad cuando éste estuviera libre. Esto, por supuesto, supondría un incremento adicional en los tiempos dedicados a las comunicaciones.

En segundo lugar, la versión paralela del circuito presenta unos tiempos de ejecución totales similares a los de la versión secuencial. Esto es debido a que el procesador envía los valores de

los dos espectros generados, a través del bus central, de manera secuencial, uno tras otro. Esto implica que, a pesar de que la versión paralela es más rápida en ejecución, el modo y los tiempos de transferencia, que implican un alto porcentaje sobre los tiempos totales de ejecución, son los mismos para ambas versiones de la metodología. La pequeña diferencia de tiempos entre ambas propuestas está determinada únicamente por la ejecución individual de cada uno de los circuitos.

Por tanto, el problema principal reside en la manera en la que se transfiere la información del procesador al periférico que implementa la metodología. El uso de un bus central implica una transmisión secuencial de la información con unos tiempos bastante superiores a los de la ejecución de los circuitos y obtención del resultado. La siguiente sección (Sección 6.3) se centrará en la propuesta de nuevos diseños hardware para optimizar los tiempos de transferencia de datos entre los módulos y dispositivos dentro de la tarjeta, utilizando técnicas, métodos y componentes más avanzados.

6.3. Segunda aproximación a SpaceCube: Optimización del periférico MEDiC.

A la hora de diseñar una metodología hardware, es importante realizar un diseño que minimice los tiempos de transferencia de datos entre los componentes y módulos que conforman la metodología para un óptimo rendimiento. Un diseño que no tenga en cuenta este parámetro puede implicar unos porcentajes de tiempos de comunicaciones superiores a 90 % sobre los tiempos de ejecución total. Por tanto, esta sección pretende estudiar diferentes estrategias a la hora de transferir los datos entre los diferentes componentes residentes en una tarjeta FPGA.

6.3.1. *Objetivos del experimento.*

En la sección anterior (Sección 6.2), se presentó una metodología implementada en hardware capaz de determinar la distancia Euclídea multi-espectral entre dos espectros. Esta metodología fue implementada como un periférico (denominado MEDiC), adjunto a un bus central que permitía realizar comunicaciones con el procesador PowerPC.

El principal problema que presenta este diseño, es que todas las transferencias de datos entre el periférico MEDiC y el procesador son realizadas por este bus central, de manera lenta (comparado con los tiempos de procesamiento) e implicando unos tiempos de comunicaciones superiores al 98 % respecto a los tiempos de ejecución de los circuitos.

En esta sección, se presentan diferentes alternativas al bus central para la realización de las transferencias, por lo que los objetivos determinados son los siguientes:

- Desarrollar una nueva estrategia de comunicaciones, basada en hardware, entre el periférico MEDiC y el procesador PowerPC.
- Comparativa del speed-up conseguido por la nueva arquitectura hardware optimizada, respecto a la versión no optimizada y la versión software.

6.3.2. *Descripción del experimento.*

En la sección anterior (Sección 6.2), se presentó una metodología capaz de realizar el cálculo de la distancia Euclídea multi-espectral entre dos espectros. Para ello, se desarrolló el periférico MEDiC, el cual realizaba todas las comunicaciones con el procesador PowerPC a través de un bus central, tal y como se indica en la Figura 6.3. Sin embargo, el utilizar un bus central, compartido con todos los periféricos presentes en el sistema, y con un modo de transferencia de datos lento (comparado con los

tiempos de ejecución reales de los circuitos) y secuencial, suponía un incremento considerable en los tiempos de ejecución totales y un decremento en el rendimiento global del sistema propuesto.

En este estudio, se ha desarrollado una nueva arquitectura que evita el bus central a la hora de realizar las transferencias de los valores de los espectros del procesador al periférico MEDiC (incluyendo la transferencia del resultado del cálculo en sentido inverso), y que hace uso de memorias dedicadas y compartidas únicamente entre dicho procesador y periférico. El tipo de memoria compartida utilizada será la memoria DDR2 presente en la tarjeta FPGA Xilinx Virtex-5 ML507, con capacidad de 128 MB y utilizando un ancho de palabra de 64 bits. De esta manera, tanto procesador como periférico MEDiC estarán conectados directamente a dicha memoria DDR2 a través del componente Multi-Port Memory Controller (MPMC) [Xil11g]. Este componente ofrece un total de ocho puertos independientes para transferir datos de manera paralela usando una conexión Native Port Interface (NPI), la conexión al componente MPMC más rápida ofrecida por Xilinx.

Se consideraron dos diseños diferentes utilizando esta misma propuesta, los cuales difieren únicamente en la forma en la que tanto procesador PowerPC y periférico MEDiC son conectados a la memoria DDR2 a través del componente MPMC:

- *Primer diseño:* Se dedica un puerto al procesador PowerPC, y los siete restantes se dedican al periférico MEDiC. En este caso, la lectura o escritura de un dato requerirá varios ciclos de reloj.
- *Segundo diseño:* Se dedica un puerto al procesador PowerPC, y cuatro puertos se dedican al periférico MEDiC. En este caso, la escritura de un dato requerirá varios ciclos de reloj, pero será posible realizar la lectura de un bloque compuesto por varios datos en un único ciclo de reloj.

Con esta arquitectura hardware, todas las transferencias de los espectros son realizadas a través de los puertos MPMC, mientras que el bus central es usado únicamente por el procesador PowerPC para indicar al periférico MEDiC que se resetee o empiece su ejecución, y por dicho periférico para indicar al procesador que ya ha terminado su ejecución y que el resultado ya está disponible. Es decir, el bus central será utilizado únicamente como medio de transmisión de señales de control entre el procesador PowerPC y el periférico MEDiC.

6.3.2.1. Primer diseño de la arquitectura hardware.

En este primer diseño, el procesador PowerPC se conecta a la memoria DDR2 a través de un único puerto del componente MPMC, mientras que el periférico MEDiC lo hace a través de los siete puertos restantes (Figura 6.6). Además, es necesario tener en cuenta las siguientes consideraciones:

- La conexión NPI posee un ancho de palabra de 64 bits. Como los valores de los espectros se representan mediante 32 bits, cada vez que se realice una lectura se recibirán a la vez dos espectros. Esto permite disminuir los tiempos de transferencia a prácticamente la mitad.
- El periférico MEDiC usará los siete puertos restantes para leer los valores de memoria en paralelo. Existen un total de 64 valores a leer en total (32 del espectro A y 32 del espectro B), por lo que los cuatro primeros puertos realizarán 5 lecturas (para obtener 10 valores de espectro cada uno, o 40 en total), mientras que los tres últimos puertos realizarán 4 lecturas (para obtener 8 valores de espectro cada uno en total, o 24 en total). Además, el último puerto (Puerto 7), debe ser capaz de escribir el resultado del cálculo en memoria.

De manera adicional, el periférico MEDiC está compuesto por varios registros de comunicaciones que permitirán al procesador PowerPC indicar a dicho periférico que resetee los circuitos (*Reset*),

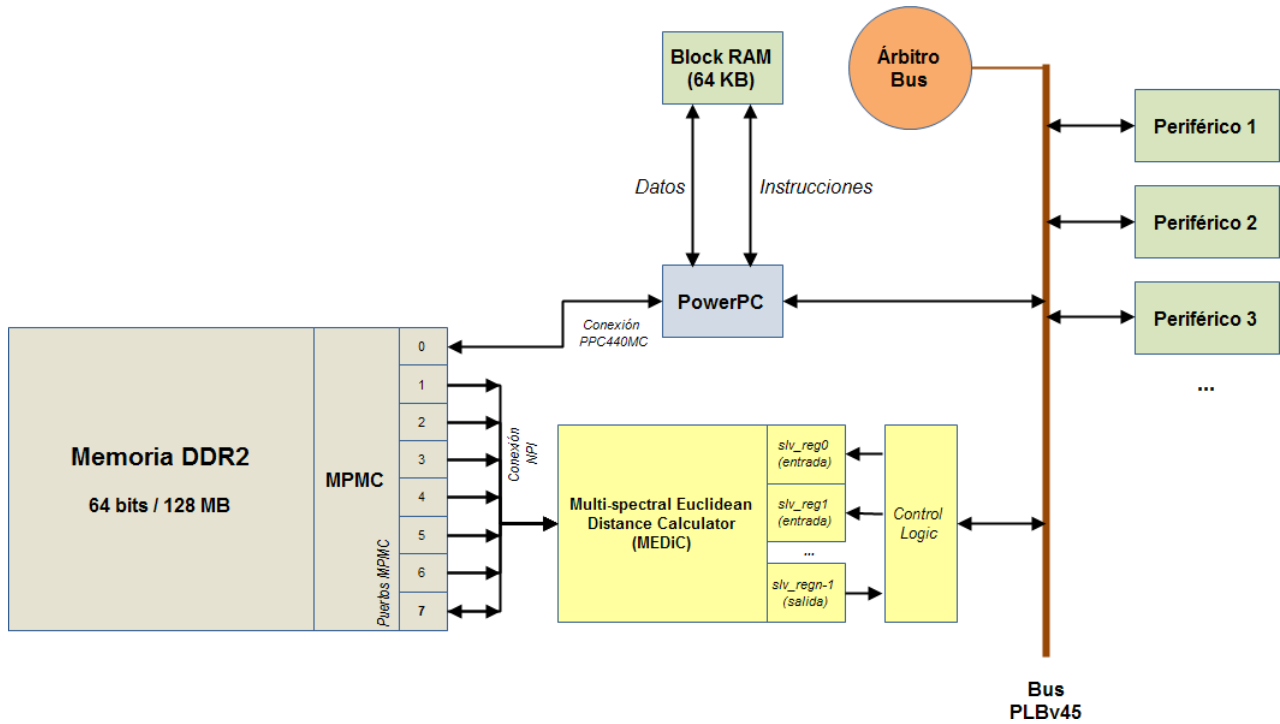


Figura 6.6: Diagrama de la conexión Memoria - Periférico - Procesador para el Diseño 1, en donde el periférico MEDiC hace uso de 7 puertos MPMC mediante una conexión NPI de 64 bits.

o que comience la ejecución cuando los espectros ya están disponibles en memoria (*Enable*); o que permitirán a dicho periférico indicar al procesador PowerPC que la ejecución ya ha finalizado y que el resultado ya está disponible en una determinada dirección de memoria (*Done*) (Figura 6.7).

El periférico MEDiC está principalmente compuesto por los siguientes módulos y registros:

- *MPMC Control Logic*: Módulo que transfiere datos de un determinado puerto MPMC. Se han desarrollado tres tipos de diferentes de módulos MPMC Control Logic, de acuerdo al número de lecturas que deben ser realizadas en cada puerto, y si es necesario que realicen únicamente lecturas, o también escrituras:
 - *MPMC Control Logic 1*: Realiza 5 lecturas de 64 bits de memoria (obteniendo 10 valores de espectro de 32 bits). Este módulo será utilizado para los puertos MPMC del 1 al 4.
 - *MPMC Control Logic 2*: Realiza 4 lecturas de 64 bits de memoria (obteniendo 8 valores de espectro de 32 bits). Este módulo será utilizado para los puertos MPMC 5 y 6.
 - *MPMC Control Logic 3*: Realiza 4 lecturas de 64 bits de memoria (obteniendo 8 valores de espectro de 32 bits). Además, está habilitado para escribir en memoria el resultado de la distancia Euclídea multi-espectral. Este módulo será utilizado para séptimo puerto MPMC.
- *Parallel MEDiC*: Módulo que realiza el cálculo de la distancia Euclídea multi-espectral de manera paralela.
- *Registro Enable*: El procesador almacena un 1 en este registro para indicar al periférico MEDiC que empiece su ejecución.
- *Registro Reset*: El procesador almacena un 1 en este registro para indicar al periférico MEDiC que resetee los circuitos.
- *Registro Done*: El periférico MEDiC almacena un 1 en este registro cuando su ejecución ha finalizado y el resultado de la distancia Euclídea multi-espectral está disponible en memoria.

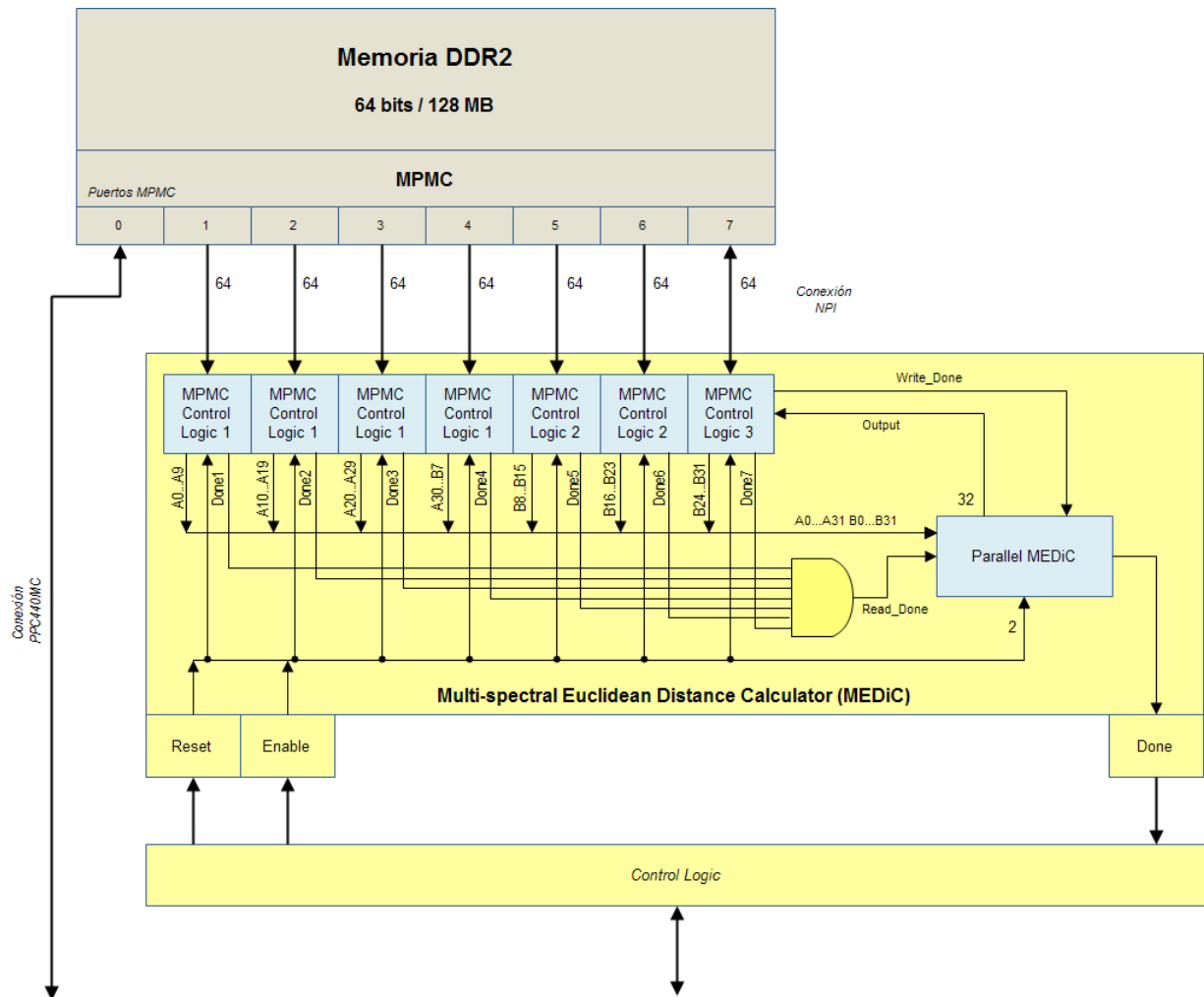


Figura 6.7: Diagrama del periférico MEDiC, mostrando los módulos del control del componente MPMC, las conexiones a los 7 puertos MPMC, y los registros de comunicaciones convencionales.

- **Control Logic:** Este módulo gestiona todas las comunicaciones procesador - periférico MEDiC, y almacena todos los valores convenientes en sus registros correspondientes.

Estos módulos se presentan de manera detallada en el Anexo A.

6.3.2.2. Segundo diseño de la arquitectura hardware.

El diseño anterior usaba la memoria DDR2 a través del componente MPMC, el cual ofrece un total de 8 puertos para transferir los datos desde y hasta memoria usando una conexión NPI. La lectura o escritura de un dato en memoria requiere varios ciclos de reloj. Xilinx ofrece, sin embargo, una manera de mejorar estos tiempos de transferencia (muy útil para optimizar las operaciones de lectura del sistema propuesto) mediante el uso de la conexión NPI en modo Burst Read. Si se habilita el modo Burst Read, y se indica la dirección base de lectura, puede obtenerse de memoria un bloque de varias palabras, en el orden de una palabra por ciclo de reloj. La limitación de este modo es que el número de palabras del bloque debe ser potencia de dos. El diseño propuesto debe leer un total de 64 valores de espectro de una memoria de 64 bits (obteniendo dos valores de espectro por lectura), por lo que es posible utilizar diferentes números de puertos MPMC:

- Si se utiliza 1 puerto MPMC, sería necesario leer un bloque de 32 palabras (64 valores de espectro). Tiempo de ejecución total: 32 cr.

- Si se utilizan 2 puertos MPMC, cada puerto debería leer un bloque de 16 palabras (32 valores de espectro). Tiempo de ejecución total: 16 cr.
- Si se utilizan 3 puertos MPMC, dos puertos deberían leer un bloque de 11 palabras cada uno (22 valores de espectro), mientras que otro puerto debería leer un bloque de 10 palabras (20 valores de espectro). Sin embargo, como el tamaño de bloque debe ser potencia de dos, los dos primeros puertos deberían leer un bloque de 16 palabras cada uno (32 valores de espectro), mientras que el otro puerto restante no leería nada. Tiempo de ejecución total: 16 cr.
- Si se utilizan 4 puertos MPMC, cada puerto debería leer un bloque de 8 palabras (16 valores de espectro). Tiempo de ejecución total: 8 cr.
- Si se utilizan 5 puertos MPMC, cuatro puertos deberían leer un bloque de 7 palabras cada uno (14 valores de espectro), mientras que otro puerto debería leer un bloque de 6 palabras (12 valores de espectro). Sin embargo, como el tamaño de bloque debe ser potencia de dos, los cuatro primeros puertos deberían leer un bloque de 8 palabras cada uno (16 valores de espectro), mientras que el otro puerto restante no leería nada. Tiempo de ejecución total: 8 cr.
- Si se utilizan 6 puertos MPMC, cinco puertos deberían leer un bloque de 6 palabras cada uno (12 valores de espectro), mientras que otro puerto debería leer un bloque de 2 palabras (4 valores de espectro). Sin embargo, como el tamaño de bloque debe ser potencia de dos, los cuatro primeros puertos deberían leer un bloque de 8 palabras cada uno (16 valores de espectro), mientras que los otros dos puertos restantes no leerían nada. Tiempo de ejecución total: 8 cr.
- Si se utilizan 7 puertos MPMC, seis puertos deberían leer un bloque de 5 palabras cada uno (10 valores de espectro), mientras que otro puerto debería leer un bloque de 2 palabras (4 valores de espectro). Sin embargo, como el tamaño de bloque debe ser potencia de dos, los cuatro primeros puertos deberían leer un bloque de 8 palabras cada uno (16 valores de espectro), mientras que los otros dos puertos restantes no leerían nada. Tiempo de ejecución total: 8 cr.

Por lo tanto, la opción óptima sería utilizar 4 puertos MPMC para leer los datos de memoria, y un quinto puerto para almacenar el resultado obtenido por el módulo Parallel MEDiC (Figura 6.8). No hay necesidad de habilitar el modo Burst Write para el quinto puerto, ya que sólo se almacenará una palabra (el resultado del cálculo de la distancia Euclídea multi-espectral).

De esta manera, el periférico MEDiC está compuesto por los mismos módulos que en el primer diseño (Figura 6.7), excepto en que sólo se hace uso de cinco puertos MPMC, y en que el módulo MPMC Control Logic ha sido modificado (Figura 6.9).

Estos módulos se presentan de manera detallada en el Anexo A.

6.3.3. Resultados del experimento.

La manera en la que se desarrollaron los resultados fueron las mismas que en la sección anterior (Sección 6.2). Sin embargo, únicamente se consideró el circuito Parallel MEDiC optimizado, y el circuito Sequential MEDiC fue descartado. Los dos diseños de arquitecturas hardware propuestas en esta sección fueron desarrollados e implementados sobre una sola tarjeta FPGA Xilinx Virtex-5 ML507 usando el entorno Xilinx XPS 12.2. Fue necesario realizar esta tarea a dos diferentes niveles:

1. Desarrollo del hardware: Desarrollo y programación de las dos arquitecturas hardware presentadas en la FPGA.
2. Desarrollo del software: Desarrollo y programación de la aplicación que será ejecutada por el procesador PowerPC, éste funcionando a una frecuencia de 400 MHz. Consistirá en la generación

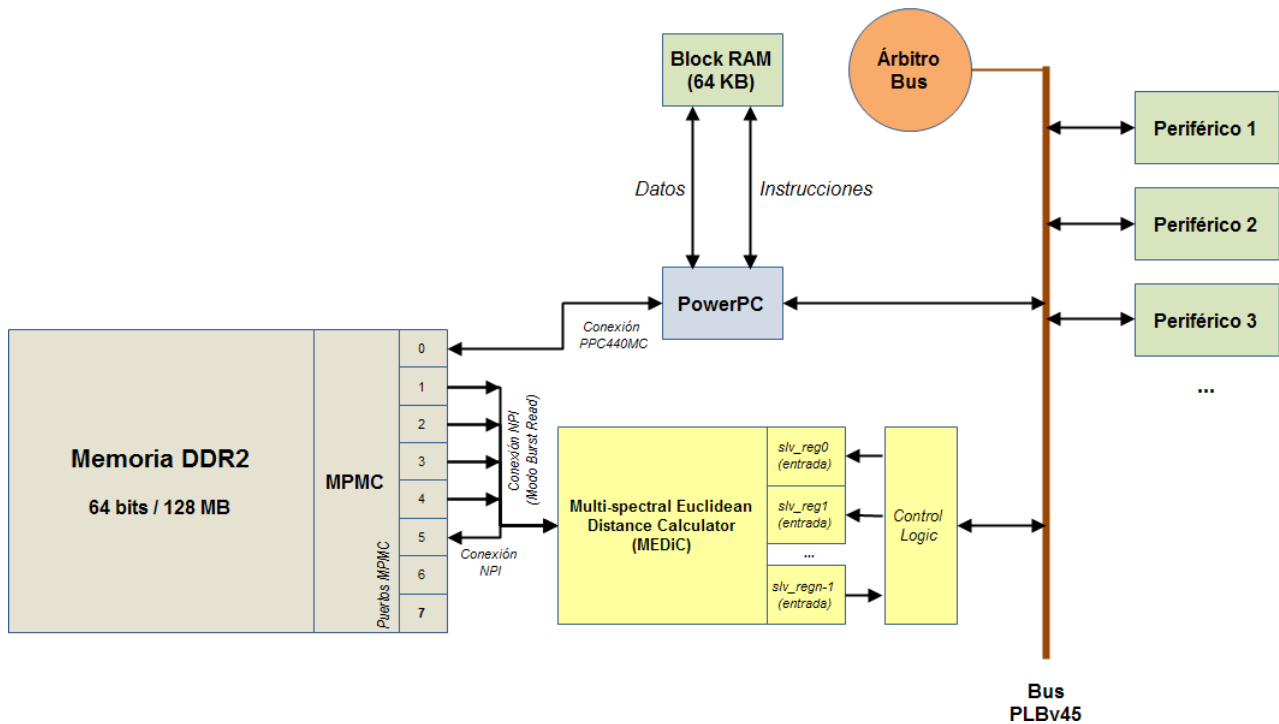


Figura 6.8: Diagrama de la conexión Memoria - Periférico - Procesador para el Diseño 2, en donde el periférico MEDiC hace uso de 4 puertos MPMC en modo Burst Read para la lectura de los datos, y un quinto para la escritura del resultado. Todos los puertos usan conexiones NPI de 64 bits.

de dos vectores (espectros) de 64 elementos (bandas) con valores aleatorios, y su posterior almacenamiento en memoria DDR2. Debido al diseño de los circuitos, se requerirán 64 iteraciones para procesar todas las bandas. La aplicación software fue escrita en lenguaje C.

Tal y como se realizó en la sección anterior, los tiempos de comunicaciones y totales de las diferentes arquitecturas fueron comparadas con el tiempo empleado por el procesador en realizar el mismo cálculo de manera independiente, y con el periférico Parallel MEDiC sin optimizar. La Tabla 6.3 muestra toda esta información, incluyendo el porcentaje de tiempo dedicado expresamente a la transferencia de los espectros y resultados (comunicaciones) y el speed-up obtenido respecto a la versión software ejecutada por el procesador (6545 cr).

	Hardware (No optimizado)	Hardware (Diseño 1)	Hardware (Diseño 2)
Tiempos ejecución (circuito)	26 cr	19 cr	19 cr
Tiempos ejecución (total)	4421 cr	612 cr	168 cr
Tiempo dedicado a comunicaciones	99.412 %	96.895 %	88.69 %
Speed-up	6545/4421=1.48X	6545/612=10.694X	6545/168=38.958X

Tabla 6.3: Tiempos empleados y speed-up de los diseños propuestos en el cálculo de la distancia Euclídea multi-espectral para espectros de 64 bandas.

6.3.4. Análisis de los resultados del experimento.

En esta sección, se ha optimizado el sistema encargado de realizar el cálculo de la distancia Euclídea multi-espectral mediante el re-diseño del método mediante el cual se llevaban a cabo las transferencias entre el procesador y el periférico MEDiC. Originalmente, las comunicaciones entre ambos dispositivos se hacían a través un bus central, el cual transfería la información de manera lenta (respecto a los tiempos de ejecución) y secuencial. En esta optimización, las transferencias a través de este bus central han quedado minimizada a únicamente las señales de control, mientras que la

procesador se realice a través de memorias compartidas, esta modalidad debería ser utilizada.

Respecto a los porcentajes dedicados a las comunicaciones, éstos son muy elevados cuando el bus central es utilizado (en el sistema original), representado cerca del 99 % de la computación global. El hecho de utilizar memorias compartidas para las transferencias entre el procesador y el periférico puede suponer un decremento del porcentaje de tiempo dedicado a esta tarea al 96 % para el primer diseño, y hasta 88 % para el segundo diseño.

De manera más general, los resultados muestran que la adaptación a hardware de operaciones básicas y sencillas, que son utilizadas en metodologías a bordo más complejas, pueden suponer una mejora del rendimiento de hasta 40X en el sistema si se consideran dispositivos como SpaceCube. Además, ya que la computación ya no recae en el procesador, sino en el sistema hardware adjunto, éste puede dedicarse a la realización de otras tareas o cálculos de otras operaciones mientras que el resultado es generado por el componente externo, mejorando de esta manera el rendimiento global. También, como este sistema externo es totalmente reconfigurable (al hacer uso de FPGAs), los diseñadores pueden mejorar o actualizar los módulos hardware e incluso crear nuevos, una vez que el satélite esté ya en órbita. Los satélites que hagan uso de la plataforma SpaceCube pueden ser totalmente configurados y adaptados a nuevas misiones, reutilizando de esta manera el satélite para propósitos nuevos o adaptados una vez que la misión principal ha finalizado. Para el caso de aviones, la ventaja es la misma. La posibilidad de utilizar un sistema hardware reconfigurable permite la elaboración de varios diseños que cumplan diferentes requisitos e implementen varias metodologías. De esta forma, el sistema podría ser re-programado de manera diferente antes de cada vuelo, de acuerdo a las necesidades de la misión.

6.4. Simulación de un sistema SpaceCube para procesamiento a bordo.

Actualmente, las agencias espaciales están apostando por sistemas que permitan minimizar el riesgo de fallos cuando la nave está adquiriendo mediciones y generando resultados en tiempo real, ya sea éstas aviones o satélites. En el caso de los aviones, siempre es posible solucionar los posibles problemas surgidos en un vuelo de cara a optimizar la adquisición en vuelos posteriores, pero en el caso de los satélites, es difícil una reparación *in situ*. Las únicas soluciones posibles es, o bien devolver el satélite a Tierra y volverlo a poner en órbita tras su reparación, o bien preparar una misión de mantenimiento y enviar astronautas para que hagan la reparación en órbita. Ambas opciones son extremadamente caras, y en ocasiones, muchas misiones espaciales terminan tras aparecer daños o fallos que impiden el correcto transcurso de las mismas. Por tanto, las arquitecturas hardware reconfigurables son una gran alternativa y se están utilizando cada vez más en los diseños de los sistemas de adquisición.

6.4.1. *Objetivos del experimento.*

La gran tendencia que existe actualmente a utilizar sistemas hardware reconfigurables, y el interés que existe alrededor de este tipo de tecnología en la comunidad científica, ha impulsado al desarrollo de nuevos diseños de procesamiento a bordo. Por tanto, los objetivos que se abarcan en este experimentos son los siguientes:

- Estudiar las posibilidades que ofrece un determinado sistema hardware reconfigurable: SpaceCube.
- Desarrollo de un sistema hardware de procesamiento a bordo del satélite.
- Utilización del formato DMF como tipo de datos de salida, y análisis de los resultados obtenidos utilizando este tipo de estructura.

- Análisis del rendimiento general del dispositivo, así como de los posibles cuellos de botellas y posibles optimizaciones en las comunicaciones.

6.4.2. Descripción del experimento.

Para este experimento, se ha realizado una simulación del dispositivo SpaceCube mediante tarjetas FPGA Xilinx Virtex-5 ML507. La manera de realizar esta simulación fue detallado de manera general en la Sección 4.2. En este apartado, se concretarán las metodologías que se han implementado y cómo se ha adaptado el diseño de la simulación propuesta para su desarrollo e implementación.

En un principio, el sistema hardware desarrollado seguirá el mismo esquema que en la Figura 4.20, pero el periférico Metodología implementará una determinada metodología en la primera tarjeta FPGA, y otra diferente en la segunda tarjeta FPGA. Estas metodologías son las siguientes:

- *Tarjeta FPGA 1*: Recibirá del exterior los espectros adquiridos por los instrumentos, junto con sus coordenadas y les aplicará un suavizado para eliminarles el ruido. Espectro suavizado y coordenadas serán enviadas a la Tarjeta FPGA 2.
- *Tarjeta FPGA 2*: Recibe el espectro suavizado de la Tarjeta FPGA 1 junto con sus coordenadas correspondientes y realiza un estudio estadístico de los valores de dicho espectro, como son la media, la desviación estándar, el máximo y el mínimo. Además, realizará un par de operaciones básicas de la teledetección relacionadas con el estudio de la vegetación. Dichos resultados, junto con el espectro suavizado y las coordenadas, serán proporcionados al exterior, donde pueden ser considerados como formato DMR.

De esta forma, el esquema general presentado con anterioridad correspondería al de la Figura 6.10 para las dos metodologías propuestas.

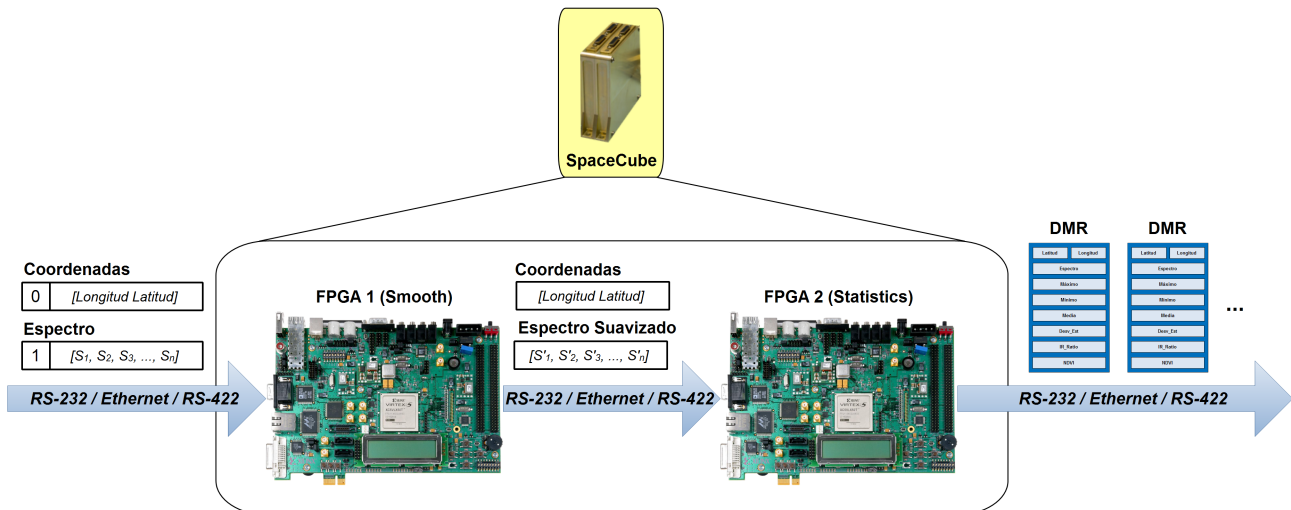


Figura 6.10: Adaptación del diseño general para la implementación de las metodologías de suavizado espectral y cálculos estadísticos.

Para la adaptación del sistema hardware a las metodologías propuestas, se han considerado un tamaño de valor de espectro de 16 bits, y un tamaño de coordenada de 32 bits, ésta última codificada en tipo coma flotante *Single* de acuerdo al estándar IEEE-754 [IEEE85]. En el resto de documentación, siempre que se haga referencia al tipo *Single*, implicará la codificación a dicho estándar.

En las subsecciones siguientes se detalla la manera en que dicha adaptación se ha realizado. Se tratan de figuras de circuitería que mostrarán los módulos que implementan las diferentes funciones y

las relaciones entre ellos. Para entender correctamente dichas figuras, es necesario tener en cuenta las siguientes consideraciones:

- Todos los módulos se rigen por el reloj del sistema (Clock), con una frecuencia máxima de 100 MHz. Sin embargo, a pesar de que todos ellos reciben una señal *CLK*, ésta ha sido eliminada de las figuras para una mejor representación de los circuitos.
- Del mismo modo, y por la misma circunstancia, todos los módulos reciben una señal Reset que permite reiniciar sus componentes. Dicha señal *RST* también ha sido eliminada de las figuras para una mejor representación de los circuitos.
- Las interacciones entre los módulos se representan mediante una flecha que indica las entradas y salidas de la señal. El formato de la flecha determina diferentes circunstancias:
 - El número de acompañe a cada flecha indica el número de bits que conforman la comunicación. En caso de que la flecha no presente ningún número, implica la transferencia de una señal de 1 bit.
 - Flechas continuas con punta negra indica que las señales transferidas son binarios naturales. Flechas punteadas con punta blanca indica que las señales transferidas son de tipo *Single*, siempre de 32 bits.
- Componentes básicos de circuitos digitales, como comparadores, latches, multiplexores, demultiplexores, puertas lógicas, contadores, registros, etc, no son detallados en la documentación. Únicamente lo son los módulos de mayor nivel desarrollados.
- El sistema es capaz de procesar espectros de hasta 256 componentes.

6.4.2.1. Metodología desarrollada en la primera tarjeta FPGA.

La primera tarjeta FPGA realizará un suavizado (smooth) al espectro recibido del exterior por el método de Savitzky-Golay (S-C) [Sav65, Mat04]. Esta técnica combina metodologías básicas de suavizado (como por ejemplo, un filtro de paso bajo) con el cálculo de derivativas de una manera muy efectiva computacionalmente. El resultado es una operación de suavizado puramente matemática que permite eliminar las componentes de alta frecuencia de las series de datos completamente o casi en su totalidad, lo que se traduce en una reducción del nivel de detalle o en la eliminación del ruido. A partir de una serie de datos $S = [S_1, S_2, S_3, \dots, S_n]$ (espectro), dicha metodología procede de la siguiente manera:

1. Aplicar un filtro de paso bajo (Low Pass Filter, LPF) [Lee80, Mor08a] a la serie de datos S . Dicho filtro consiste en un recorrido de ventana de 1×3 aplicando un kernel $k = [1/3, 1/3, 1/3]$, de tal manera que $S_i = S_{i-1} \times k_1 + S_i \times k_2 + S_{i+1} \times k_3$. El resultado es una nueva serie $S_{LPF} = [S_{LPF_1}, S_{LPF_2}, S_{LPF_3}, \dots, S_{LPF_n}]$.
2. Aplicar un filtro de paso alto (High Pass Filter, HPF) [Pol00, Mor08b] a la serie de datos S . Dicho filtro consiste en un recorrido de ventana de 1×3 aplicando un kernel $k = [-1/3, 2/3, -1/3]$, de tal manera que $S_i = S_{i-1} \times k_1 + S_i \times k_2 + S_{i+1} \times k_3$. Del resultado, mantendremos únicamente los valores impares, por que se obtendrá una nueva serie compuesta por la mitad de valores ($\lfloor n/2 \rfloor$) que en la original $S_{HPF} = [S_{HPF_1}, S_{HPF_3}, S_{HPF_5}, \dots, S_{HPF_{n-(n\%2-1)}}] = [S'_{HPF_1}, S'_{HPF_2}, S'_{HPF_3}, \dots, S'_{HPF_{\lfloor n/2 \rfloor}}]$.
3. Calcular la media de S'_{HPF} , de acuerdo a:

$$\overline{S'_{HPF}} = \frac{\sum_{i=1}^{\lfloor n/2 \rfloor} S'_i}{\lfloor n/2 \rfloor} \quad (6.8)$$

4. Calcular la desviación estándar de S'_{HPF} , de acuerdo a:

$$\sigma_{S'_{HPF}} = \sqrt{\frac{\sum_{i=1}^{\lfloor n/2 \rfloor} (S'_i - \overline{S'_{HPF}})^2}{\lfloor n/2 \rfloor - 1}} \quad (6.9)$$

5. Cálculo del Umbralizado Universal (Universal Threshold, UT) [Don95, Agg11], definido por:

$$UT = \sqrt{2 \times \log(\lfloor n/2 \rfloor)} \times \sigma_{S'_{HPF}} \quad (6.10)$$

6. Realizar un umbralizado a S'_{HPF} , poniendo a 0 aquellos valores fuera del rango $[-UT, UT]$. El resultado es una nueva serie $T' = [T'_1, T'_2, T'_3, \dots, T'_{\lfloor n/2 \rfloor}]$.

7. Como la serie T' actualmente está compuesta por los valores impares de S'_{HPF} , recuperamos el número de componentes originales rellenando a 0 los valores pares, creando de esta manera una nueva serie tal que $T = [T'_1, 0, T'_2, 0, T'_3, 0, \dots, T'_{\lfloor n/2 \rfloor} \times n \% 2] = [T_1, T_2, T_3, \dots, T_n]$.

8. Realizar la suma componente a componente de las series de datos S_{LPF} y T , creando una nueva serie tal que $S_{Smooth} = [S_{LPF_1} + T_1, S_{LPF_2} + T_2, S_{LPF_3} + T_3, \dots, S_{LPF_n} + T_n]$.

La serie obtenida mediante S_{Smooth} se trata del espectro suavizado, el resultado proporcionado por la primera tarjeta FPGA.

De esta manera, para implementar esta metodología, se adaptó el sistema hardware genérico propuesto en la Sección 4.2. En la adaptación de la primera tarjeta FPGA, se desarrolló un periférico denominado *Smooth* compuesto en un alto nivel por tres módulos principales que permiten la ejecución del suavizado de los espectros, así como del control de las comunicaciones con el exterior (Figura 6.11):

- *FIFO_E*: Recibe o bien un valor de espectro (S), o bien una coordenada, representada por la latitud (Lat) y por la longitud ($Long$). Las señales S_W , S_Lat y S_Long se activarán cuando se quiera insertar en las memorias FIFO un nuevo valor de espectro, de latitud y de longitud, respectivamente. Finalmente, la señal RDY_I indica que el módulo Smooth está disponible para recibir un nuevo espectro, junto con sus coordenadas, para su procesamiento (si es que hay disponibles). Esto ocurrirá tanto en la primera ejecución del circuito, indicada mediante la señal E , o bien cuando el módulo Smooth haya terminado de procesar la medición actual (RDY). De ahí surge la necesidad de utilizar la puerta OR del centro del diagrama. Cuando el dispositivo está listo para procesar un nuevo espectro, y hay alguno disponible, *FIFO_E* se lo proporciona al módulo Smooth, y le indica que puede comenzar el suavizado mediante la señal $textit{RDY}$. Para el correcto funcionamiento del necesitará varias señales adicionales:
 - *Bands*: Número de valores que componen los espectros.
 - *Bands_2*: Siendo *Bands* el número de valores que componen el espectro, esta señal define $\lceil Bands/2 \rceil$.
- *Smooth*: Recibe del componente *FIFO_E* el espectro ($S1, S2, S3, \dots, S256$), y realiza el suavizado de acuerdo al algoritmo S-C detallado con anterioridad. El resultado (espectro suavizado) es proporcionado al módulo FIFO_S para su almacenamiento, indicándole de este hecho mediante la activación de la señal RDY . Para su correcto funcionamiento necesitará varias señales adicionales:

- $Bands2_1$: Siendo $Bands$ el número de valores que componen el espectro, esta señal define $\lceil Bands/2 \rceil - 1$.
 - $Bands2_D$: El número de valores que componen el espectro en formato *Single*.
 - $Bands2_D_1$: Se trata de la misma señal $Bands2_1$ pero en formato *Single*.
 - UT_SQRT : Resultado de $\sqrt{2 \times \log(\lfloor n/2 \rfloor)}$.
- $FIFO_S$: Recibe del módulo Smooth el espectro suavizado ($S1, S2, S3, \dots, Sn$) y lo almacena, junto con sus coordenadas Lat y $Long$, en las memorias FIFO correspondientes cuando se activa la señal RDY . Una vez almacenados, la señal $Data_RDY$ permanece activa indicando al exterior que ya existen resultados procesados. El exterior realiza una petición y recibe dichos datos activando la señal $Read_Data$, enviando de esta manera $FIFO_S$ toda la información almacenada relativa a una medición.

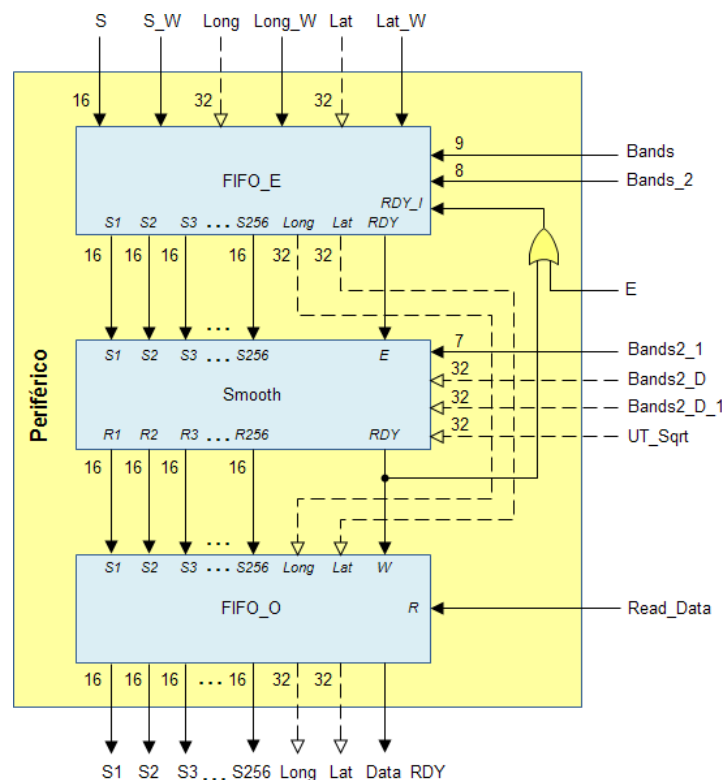


Figura 6.11: Diagrama detallado del periférico Smooth.

Las señales $Bands$, $Bands_2$, $Bands2_1$, $Bands2_D$, $Bands2_D_1$ y UT_SQRT , al tratarse de valores y operaciones constantes, son generadas por el procesador PowerPC antes de indicarle al circuito que empiece a funcionar.

El módulo FIFO_E.

Este módulo recibe ráfagas de valores de espectro, o bien de coordenadas (latitud y longitud) y los almacena en memorias FIFO internas. Cuando ya se encuentra almacenado un espectro completo junto con sus coordenadas, entonces se extraen de las memorias FIFO y se transfiere al módulo Smooth. De esta forma, a este módulo siempre le llegarán los datos de manera ordenada.

El diseño del módulo FIFO_E se muestra en la Figura 6.12. Se trata de una adaptación del genérico presentado en la Sección 4.2.1, siendo el funcionamiento completamente el mismo. Únicamente es necesario tener en cuenta las siguientes consideraciones:

- Se ha añadido el número de bits y tipo de cada camino de datos.
- El número de valores de espectro máximo que es posible considerar es 256.

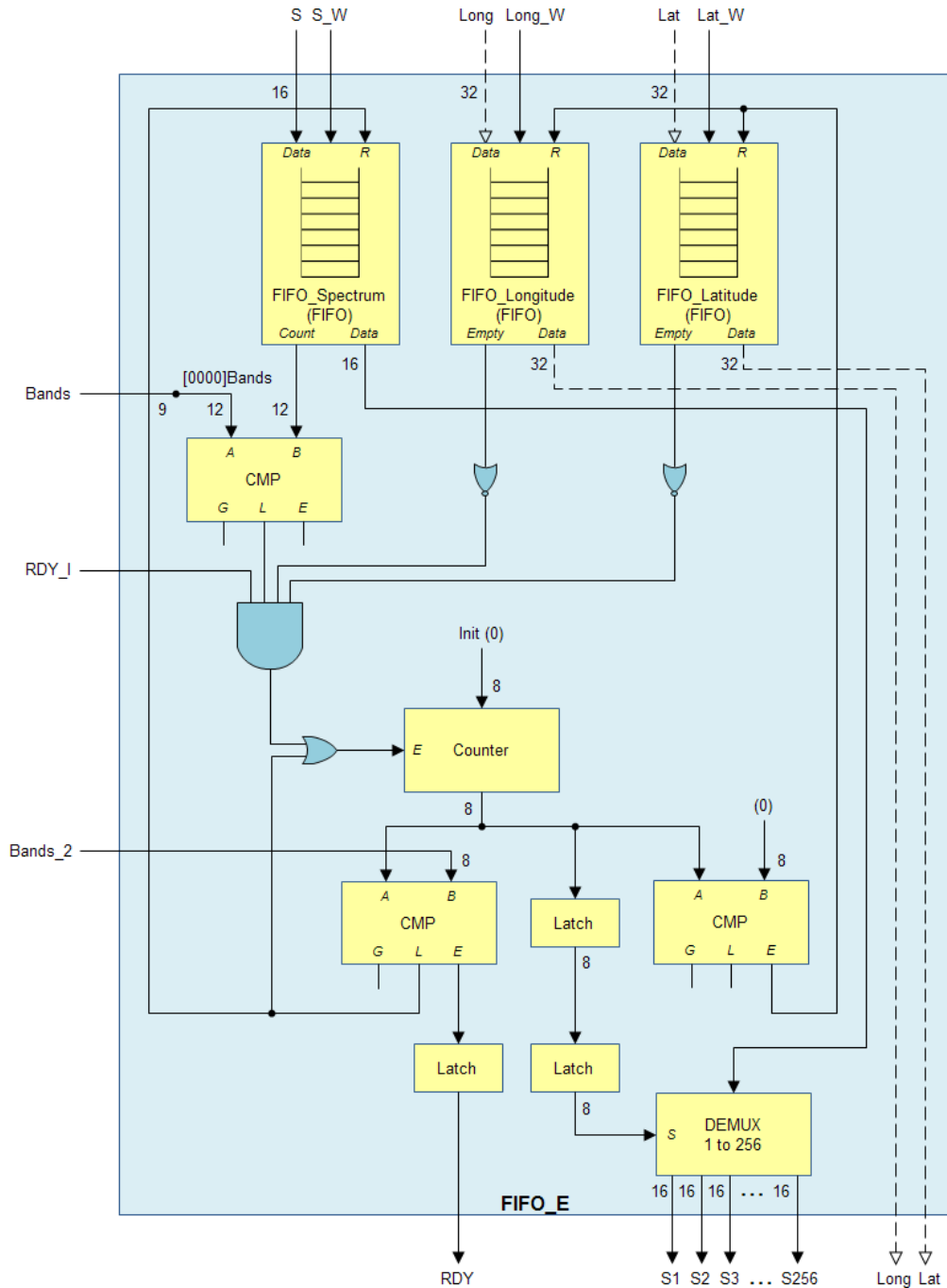


Figura 6.12: Diagrama del módulo FIFO_E.

El módulo Smooth.

El módulo Smooth recibe un espectro y realiza su suavizado de acuerdo al algoritmo S-C, el cual fue detallado en la sección anterior.

El diagrama de dicho módulo está representando en la Figura 6.13. Como puede observarse, se trata de una secuencia de submódulos que, tras recibir todos los valores del espectro, realizan con ellos cada uno de los pasos del algoritmo de suavizado.

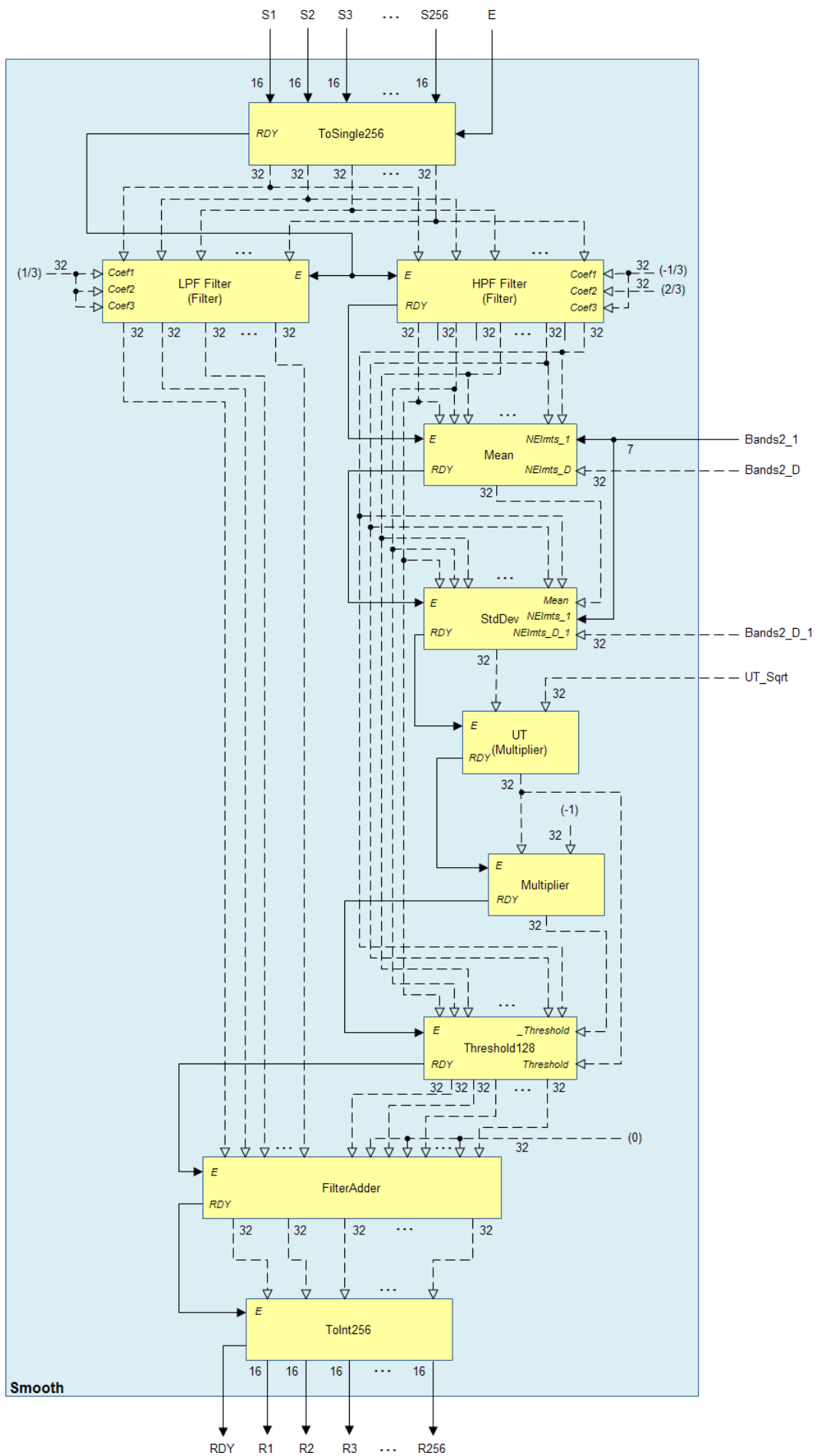


Figura 6.13: Diagrama del módulo Smooth.

El funcionamiento del mismo se muestra a continuación:

1. El módulo encargado de recibir los datos del espectro es *IntToDouble*. Como su nombre indica, realiza una conversión de los valores del espectro a tipo *Single*. El objetivo de esta conversión es conseguir mayor precisión a la hora de realizar los cálculos internos en el resto de módulos encargados del suavizado. Una vez que la conversión ha finalizado, se activa la señal *RDY*, la cual es recibida por los módulos *LPF Filter* y *HPF Filter*.
2. Una vez la señal *RDY* de *IntToSingle* se hace activa, el módulo *LPF Filter* comienza a funcionar. Obtendrá los valores del espectro en formato *Single* y les aplicará un filtrado paso bajo. Para ello, necesitará los valores del kernel a aplicar (para el caso del filtro de paso bajo será $[1/3, 1/3, 1/3]$), los cuales son introducidos como señales constantes *Coef1*, *Coef2* y *Coef3*.
3. De igual manera, la señal *RDY* de *IntToSingle* llega al módulo *HPF Filter*, el cual empezará a trabajar en paralelo con *LPF Filter*. Este módulo obtendrá los valores del espectro en formato *Single* y les aplicará un filtrado paso bajo. Para ello, necesitará los valores del kernel a aplicar (para el caso del filtro de paso bajo será $[-1/3, 2/3, -1/3]$), los cuales son introducidos como señales constantes *Coef1*, *Coef2* y *Coef3*. Una vez finalizada la ejecución, únicamente los espectros impares son transferidos al siguiente módulo *Mean*, el cual empezará a funcionar al activarse la señal *RDY* de *HPF Filter*.
4. Cuando se activa la señal *RDY* de *HPF Filter*, el módulo *Mean* recibe los valores resultantes del filtrado y calcula la media de los mismos. Para ello, necesita conocer los cálculos $\lceil Bands/2 \rceil - 1$ y $\lceil Bands/2 \rceil$ (formato *Single*, los cuales se proporcionan mediante las señales *Bands2_1* y *Bands2_D*). Una vez la media ha sido calculada, el resultado es transferido al módulo *StdDev*, el cual empezará a funcionar al activarse la señal *RDY* de *Mean*.
5. Al activarse la señal *RDY* de *Mean*, el módulo *StdDev* recibe los valores resultantes del filtrado junto con la media de los mismos, y calcula la desviación estándar. Para ello, necesita conocer el cálculo $\lceil Bands/2 \rceil - 1$ (en formatos binario natural y *Single*, los cuales son proporcionados mediante las señales *Bands2_1* y *Bands2_D_1*, respectivamente). Una vez la desviación estándar ha sido calculada se activará la señal *RDY*.
6. El siguiente multiplicador se encargará del cálculo del valor de Umbralizado Universal UT cuando se active la señal *RDY* de *StdDev*. A partir del cálculo $\sqrt{2 \times \log(\lceil n/2 \rceil)}$ representado como constante en la señal *UT_Sqrt*, realiza la multiplicación de esta componente con la desviación estándar calculada en el módulo anterior, proporcionando el resultado de la operación $\sqrt{2 \times \log(\lceil n/2 \rceil)} \times \sigma$. Cuando el valor de umbralizado UT haya sido calculado, se transferirá al siguiente multiplicador y se activará la señal *RDY*.
7. El segundo multiplicador se encarga de determinar el valor de -UT. Para ello, cuando se active la señal *RDY* del anterior multiplicador, obtendrá el valor de UT calculado y lo multiplicará por -1 . Este parámetro es proporcionado como una señal constante con la representación de -1 en formato *Single*. Una vez terminada la ejecución, se transfiere el resultado al módulo *Threshold128* y se activará la señal *RDY*.
8. Al activarse la señal *RDY* del segundo multiplicador, el módulo *Threshold128* empezará a funcionar. A los valores de espectro impares filtrados mediante el módulo *HPF Filter* (S'_{HPF_i}), les asignará un valor 0 a aquéllos que no cumplan $-UT \leq S'_{HPF_i} \leq UT$. Tras realizar el umbralizado a todos los valores del espectro, éste se transfiere en su totalidad y se activará la señal *RDY*.
9. Tras finalizar *Threshold128*, la señal *RDY* activada llega al módulo *FilterAdder*, junto con los 128 valores de espectro umbralizados rellenos con valores 0 (en formato *Single* en posiciones pares (por lo tanto 256 valores) y los 256 valores resultantes de aplicar el filtro de paso bajo a los espectros originales. Este módulo realiza la suma de ambas series de datos componente a componente, creando una nueva serie que representará el espectro original suavizado en formato

Single. Esta serie será transferida al módulo ToInt256, indicando que puede empezar a funcionar mediante la activación de la señal *RDY*.

10. Al recibir los valores en formato *Single* y la señal *RDY* activa, el módulo ToInt256 los convertirá en formato binario natural. Mediante la activación de la señal *RDY*, este módulo comunica al exterior que la operación de suavizado ha finalizado.

Estos módulos se presentan de manera detallada en el Anexo A.

El módulo FIFO_S.

Este módulo recibe el espectro suavizado del módulo Smooth, junto con las coordenadas (latitud y longitud) correspondientes y los almacena en memorias FIFO internas. Cuando ya se encuentra almacenado un espectro suavizado completo junto con sus coordenadas, entonces se envía una señal *DATA_RDY* al exterior, el cual puede proceder a la lectura de la medición completa activando la señal *R*.

El diseño del módulo FIFO_S se muestra en la Figura 6.14. Se trata de una adaptación del genérico presentado en la Sección 4.2.2, siendo el funcionamiento completamente el mismo. Únicamente es necesario tener en cuenta las siguientes consideraciones:

- Se ha añadido el número de bits y tipo de cada camino de datos.
- El resultado final es el espectro suavizado, pudiendo estar compuesto hasta por 256 elementos. Por lo tanto, serán necesarias un total de 256 memorias FIFO individuales y 2 memorias FIFO individuales para la latitud y la longitud.

6.4.2.2. Metodología desarrollada en la segunda tarjeta FPGA.

La segunda tarjeta FPGA recibirá los datos de la primera tarjeta, la cual ha realizado un suavizado de los espectros, y realizará una serie de cálculos estadísticos y operaciones básicas de la teledetección sobre ellos. Estos cálculos son los siguientes:

- *Máximo*: Cálculo del máximo valor del espectro suavizado.
- *Mínimo*: Cálculo del mínimo valor del espectro suavizado.
- *Media*: Cálculo de la media de los valores del espectro suavizado.
- *Desviación estándar*: Cálculo de la desviación estándar de los valores del espectro suavizado.
- *IR Ratio* [Jac91, Tuc79]: Operación básica de la teledetección que permite establecer la relación entre la magnitud del espectro en el infrarrojo cercano (NIR) respecto al visible en las frecuencias del rojo (R), tal que:

$$IR_Ratio = \frac{NIR}{R} \quad (6.11)$$

Esta operación es una de las operaciones más comunes en los estudios de los estados de la vegetación, ya que considera que una vegetación activa refleja fuertemente la radiación en el infrarrojo cercano, y la absorbe en frecuencias cercanas al rojo. A pesar de que esta operación no destaca precisamente por su precisión, es bastante útil como primera aproximación y como metodología auxiliar a bordo de la nave.

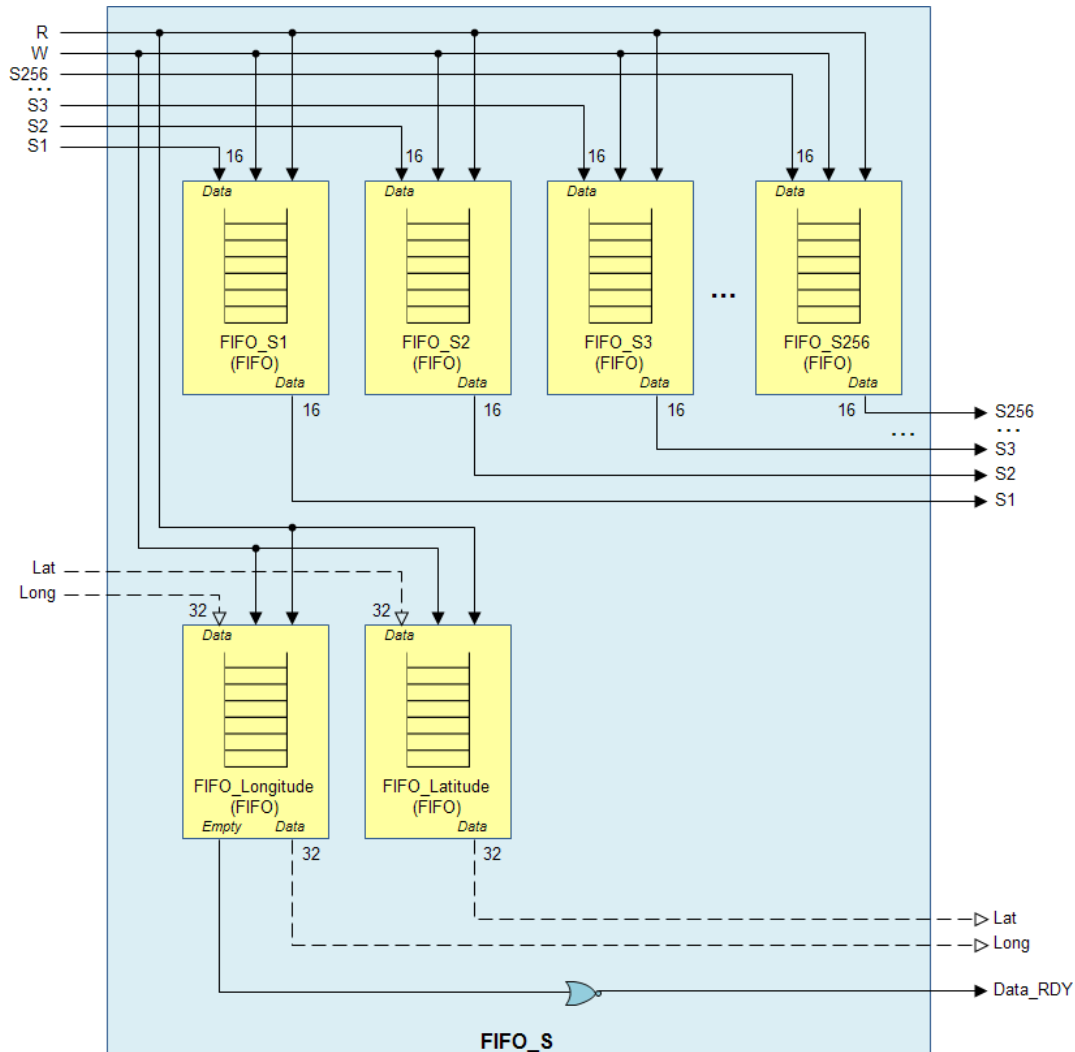


Figura 6.14: Diagrama del módulo FIFO_S.

- *Normalized Difference Vegetation Index (NDVI)* [Car97, Zen11]: Se trata de un ratio para estudio de la vegetación, igual que IR Ratio, pero que proporciona una mayor precisión. De igual manera que el anterior, se basa en el estudio de los valores del espectro en las frecuencias de infrarrojo cercano (NIR) y rojo (R), tal que:

$$NDVI = \frac{NIR - R}{NIR + R} \quad (6.12)$$

La Tabla 6.4 muestra las bandas de cada una de las imágenes del *dataset real*, así como las longitudes de onda (λ) correspondientes, que han sido seleccionadas como valores de infrarrojo cercano (NIR) y rojo (R) para la realización del cálculo de los índices IR Ratio y NDVI.

	R (Banda / λ)	NIR (Banda / λ)
<i>Cuprite</i>	35 / 673.64 nm	43 / 749.96 nm
<i>Barrax</i>	25 / 674.42 nm	37 / 751.36 nm
<i>Mediterráneo</i>	1 / 659 nm	2 / 865 nm
<i>GMéxico</i>	35 / 675.4 nm	43 / 753.63 nm
<i>Cantera</i>	8 / 659 nm	11 / 746 nm

Tabla 6.4: Bandas seleccionadas como valores R y NIR para cada una de las imágenes del *dataset real*.

Igual que en la primera tarjeta FPGA, para implementar esta metodología se adaptó el sistema hardware genérico propuesto en la Sección 4.2. De esta manera, se desarrolló un periférico denominado

Statistics compuesto en un alto nivel por tres módulos principales que permiten la ejecución de las operaciones, así como del control de las comunicaciones con el exterior (Figura 6.15):

- *FIFO_E*: Recibe o bien un valor de espectro suavizado (S), o bien una coordenada, representada por la latitud (Lat) y por la longitud ($Long$). Las señales S_W , S_Lat y S_Long se activarán cuando se quiera insertar en las memorias FIFO un nuevo valor de espectro suavizado, de latitud y de longitud, respectivamente. Finalmente, la señal RDY_I indica que el módulo *Statistics* está disponible para recibir un nuevo espectro, junto con sus coordenadas, para su procesamiento (si es que hay disponibles). Esto ocurrirá tanto en la primera ejecución del circuito, indicada mediante la señal E , o bien cuando el módulo *Statistics* haya terminado de procesar la medición actual (RDY). De ahí surge la necesidad de utilizar la puerta OR del centro del diagrama. Cuando el dispositivo está listo para procesar un nuevo espectro suavizado, y hay alguno disponible, *FIFO_E* se lo proporciona al módulo *Statistics*, y le indica que puede comenzar el suavizado mediante la señal RDY . Para el correcto funcionamiento del necesitará varias señales adicionales:
 - *Bands*: Número de valores que componen los espectros.
 - *Bands_2*: Siendo *Bands* el número de valores que componen el espectro, esta señal define $\lceil Bands/2 \rceil$.
- *Statistics*: Recibe del componente *FIFO_E* el espectro suavizado ($S_1, S_2, S_3, \dots, S_{256}$), y realiza el cálculo del máximo, mínimo, media, desviación estándar, IR Ratio y NDVI. Los resultados, junto con el espectro suavizado original son proporcionados al módulo *FIFO_S* para su almacenamiento, indicándole de este hecho mediante la activación de la señal RDY . Para su correcto funcionamiento necesitará varias señales adicionales:
 - *Bands2_1*: Siendo *Bands* el número de valores que componen el espectro, esta señal define $\lceil Bands/2 \rceil - 1$.
 - *Bands2_D*: El número de valores que componen el espectro en formato *Single*.
 - *Bands2_D_1*: Se trata de la misma señal *Bands2_1* pero en formato *Single*.
 - *BandIR*: Posición del valor de espectro correspondiente a la frecuencia de infrarrojo cercano.
 - *BandR*: Posición del valor de espectro correspondiente a la frecuencia del rojo.
- *FIFO_S*: Recibe del módulo *Statistics* el espectro suavizado ($S_1, S_2, S_3, \dots, S_n$) y los resultados de las operaciones y los almacena, junto con sus coordenadas Lat y $Long$, en las memorias FIFOs correspondientes cuando se activa la señal RDY . Una vez almacenados, la señal $Data_RDY$ permanece activa indicando al exterior que ya existen resultados procesados. El exterior realiza una petición y recibe dichos datos activando la señal $Read_Data$, enviando de esta manera *FIFO_S* toda la información almacenada relativa a una medición.

Como se ha comentado, todos los resultados proporcionados por la segunda tarjeta FPGA, junto con el espectro suavizado de la primera tarjeta FPGA, son proporcionados al exterior junto con las coordenadas (latitud y longitud) correspondientes. Esta salida, puede considerarse como un registro DMR en el cual se han añadidos nuevos campos correspondientes a los cálculos del máximo, mínimo, media, desviación estándar, IR Ratio y NDVI.

Además, al igual que en la primera tarjeta FPGA, las señales *Bands*, *Bands_2*, *Bands2_1*, *Bands2_D*, *Bands2_D_1*, *BandIR* y *BandR*, al tratarse de valores y operaciones constantes, son generadas por el procesador PowerPC antes de indicarle al circuito que empiece a funcionar.

El módulo FIFO_E.

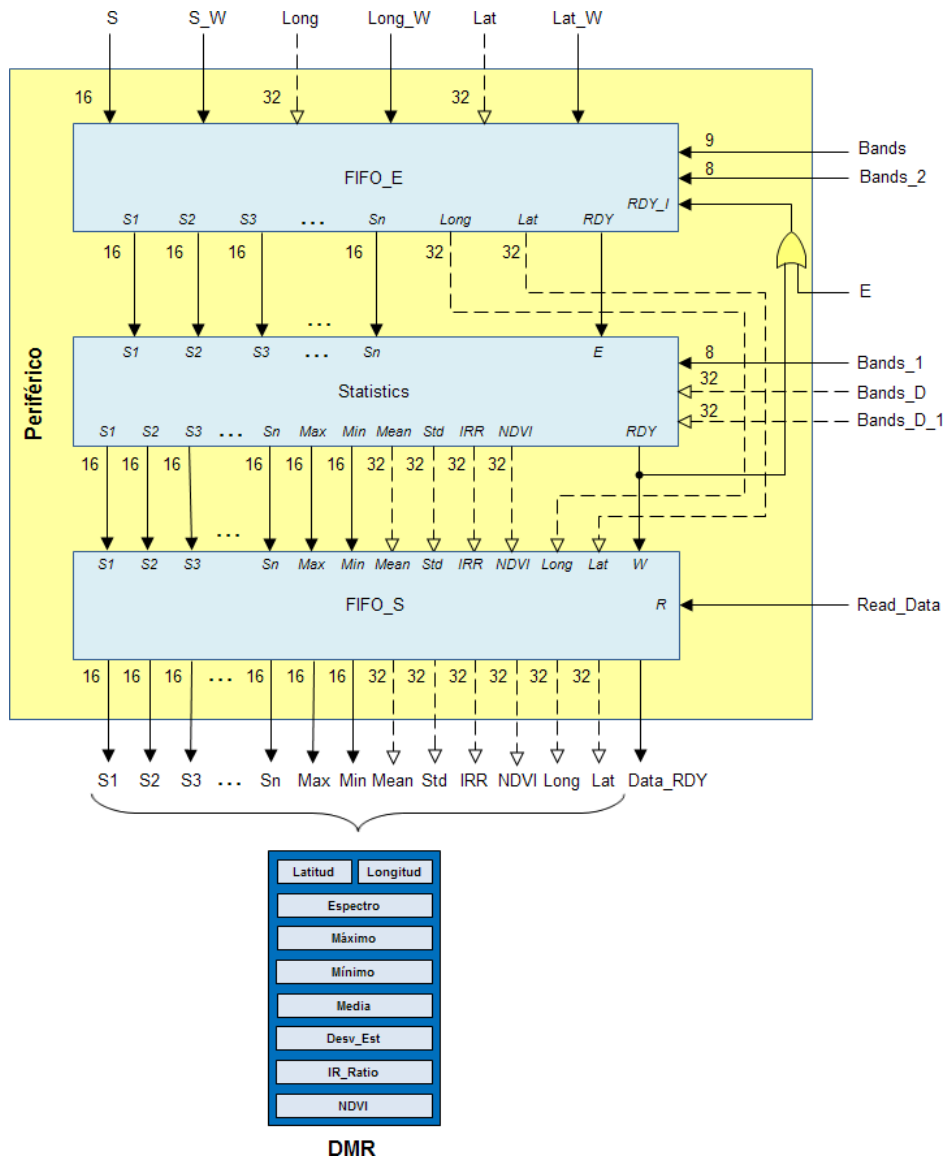


Figura 6.15: Diagrama detallado del periférico Statistics.

Este módulo recibe ráfagas de valores de espectro, o bien de coordenadas (latitud y longitud) y los almacena en memorias FIFO internas.

Su funcionamiento es exactamente el mismo que el de la primera tarjeta, por lo que su descripción ya está detallado en la Sección 6.4.2.1.

El módulo Statistics.

El módulo Statistics permite calcular los valores máximo, mínimo, media, desviación estándar, IR Ratio y NDVI de los valores de espectro de entrada.

El diagrama de dicho módulo está representando en la Figura 6.16. Como puede observarse, se trata de una secuencia de submódulos que, tras recibir todos los valores del espectro suavizado, realizan con ellos cada uno de los cálculos comentados.

El funcionamiento del mismo se muestra a continuación:

1. El módulo encargado de recibir los datos del espectro suavizado es IntToSingle, el cual realiza una

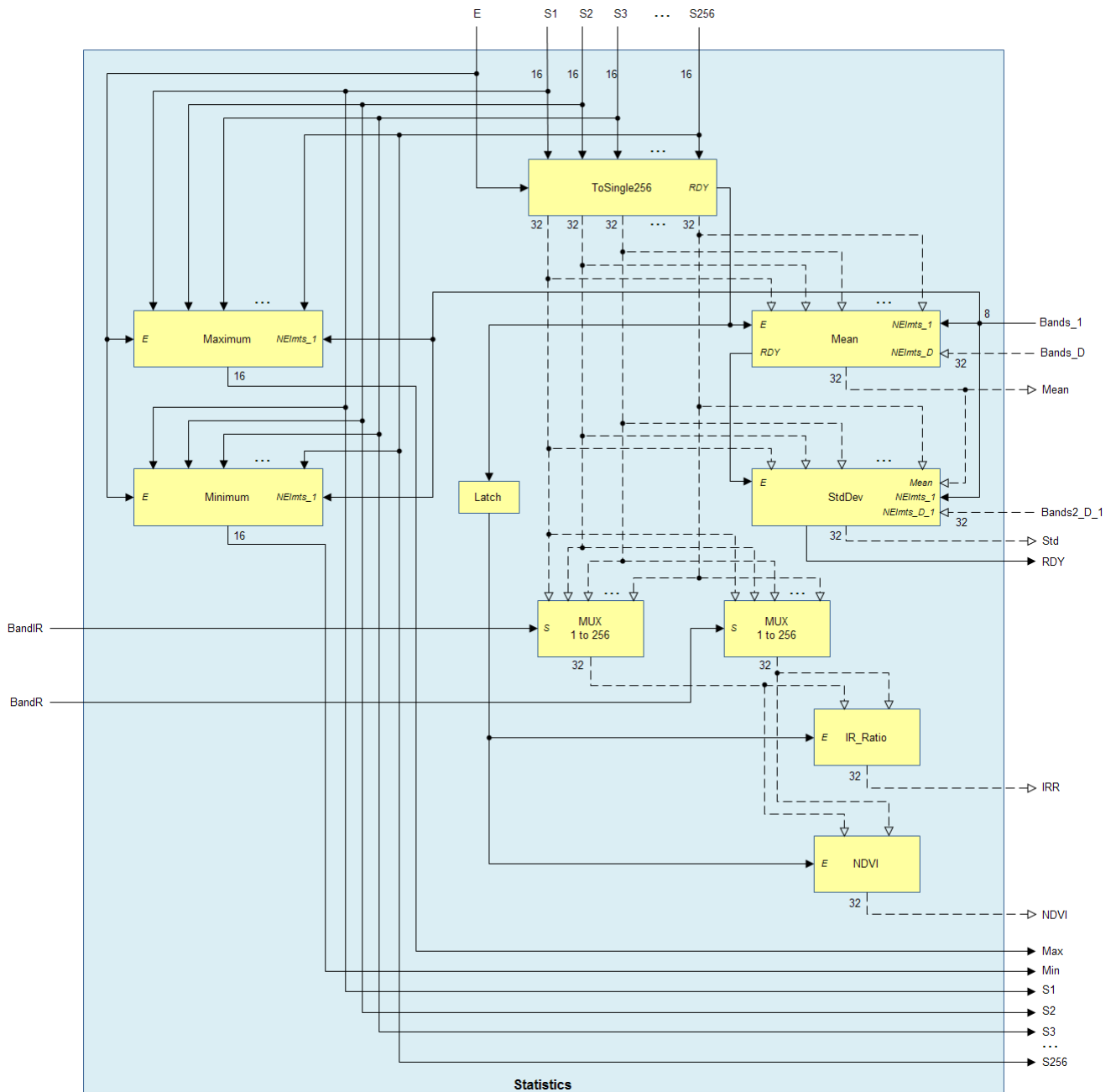


Figura 6.16: Diagrama del módulo Statistics.

conversión de los valores del espectro a tipo *Single*. El objetivo de esta conversión es conseguir mayor precisión a la hora de realizar los cálculos internos en el resto de módulos. Una vez que la conversión ha finalizado, se activa la señal *RDY*, la cual es recibida por el módulo Mean, y por los dos multiplexores encargados de seleccionar los valores de infrarrojo cercano y rojo correspondientes.

2. De manera paralela a *IntToSingle*, se ejecutan los módulos Maximum y Minimum, los cuales recibirán los espectros en su formato original (sin conversión a *Single*), y que determinarán los valores máximo y mínimo de los espectros suavizados, respectivamente.
3. Una vez recibida la señal *RDY* del módulo *ToDouble256*, el módulo Mean empezará a funcionar con el objetivo de calcular la media de los espectros suavizados en formato *Single*. Para ello, necesita conocer los valores $\lceil Bands/2 \rceil - 1$ y $\lceil Bands/2 \rceil$ (formato *Single*, los cuales se proporcionan mediante las señales *Bands2_1* y *Bands2_D*. Una vez la media ha sido calculada, el resultado es transferido al módulo *StdDev*, el cual empezará a funcionar al activarse la señal *RDY* de Mean.
4. Al activarse la señal *RDY* de Mean, el módulo *StdDev* recibe los valores resultantes del filtrado junto con la media de los mismos, y calcula la desviación estándar. Para ello, necesita conocer el cálculo $\lceil Bands/2 \rceil - 1$ (en formatos binario natural y *Single*, los cuales son proporcionados mediante las señales *Bands2_1* y *Bands2_D_1*, respectivamente. Una vez la desviación estándar ha sido calculada se activará la señal *RDY*. Como el módulo *StdDev* será el último en finalizar su ejecución, su señal *RDY* será utilizada para comunicar al exterior que todas las operaciones han sido finalizadas.
5. Una vez recibida la señal *RDY* del módulo *ToDouble256*, también los dos multiplexores se activarán. Mediante las selectoras *BandIR* y *BandR* permiten seleccionar los valores del espectro de entrada que representarán a las frecuencias de infrarrojo cercano y rojo, respectivamente. Ambos valores serán transmitidos a los módulos *IR_Ratio* y *NDVI*.
6. Los módulos *IR_Ratio* y *NDVI* recibirán los valores del espectro de entrada correspondientes a las frecuencias de infrarrojo cercano y rojo, respectivamente, un ciclo de reloj después de que hayan sido determinados por los multiplexores (retardo controlado por el latch central). Ambos módulos proporcionarán el cálculo de los índices *IR Ratio* y *NDVI* tal y como se detallaron en las Ecuaciones 6.11 y 6.12.
7. Las salidas del módulo *Statistics* consistirán en los resultados obtenidos tras calcular el máximo, el mínimo, la media, la desviación estándar, el *IR Ratio* y el *NDVI*, junto con el espectro suavizado original.

Estos módulos se presentan de manera detallada en el Anexo A.

El módulo FIFO_S.

Este módulo recibe el espectro suavizado y los cálculos de los valores máximo, mínimo, media, desviación estándar, *IR Ratio* y *NDVI* del módulo *Statistics*, junto con las coordenadas (latitud y longitud) correspondientes y los almacena en memorias FIFO internas. Cuando ya se encuentra almacenado un espectro suavizado completo junto con sus coordenadas y sus cálculos, entonces se envía una señal *DATA_RDY* al exterior, el cual puede proceder a la lectura de la medición completa activando la señal *R*.

El diseño del módulo *FIFO_S* se muestra en la Figura 6.17. Se trata de una adaptación del genérico presentado en la Sección 4.2.2, siendo el funcionamiento completamente el mismo. Únicamente es necesario tener en cuenta las siguientes consideraciones:

- Se ha añadido el número de bits y tipo de cada camino de datos.
- El resultado final es el espectro suavizado, pudiendo estar compuesto hasta por 256 elementos, y los cálculos de los valores máximo, mínimo, media, desviación estándar, IR Ratio y NDVI. Por lo tanto, serán necesarias un total de 256 memorias FIFO individuales para almacenar el espectro completo, 6 memorias FIFO individuales para los cálculos, y 2 memorias FIFO individuales para la latitud y la longitud.

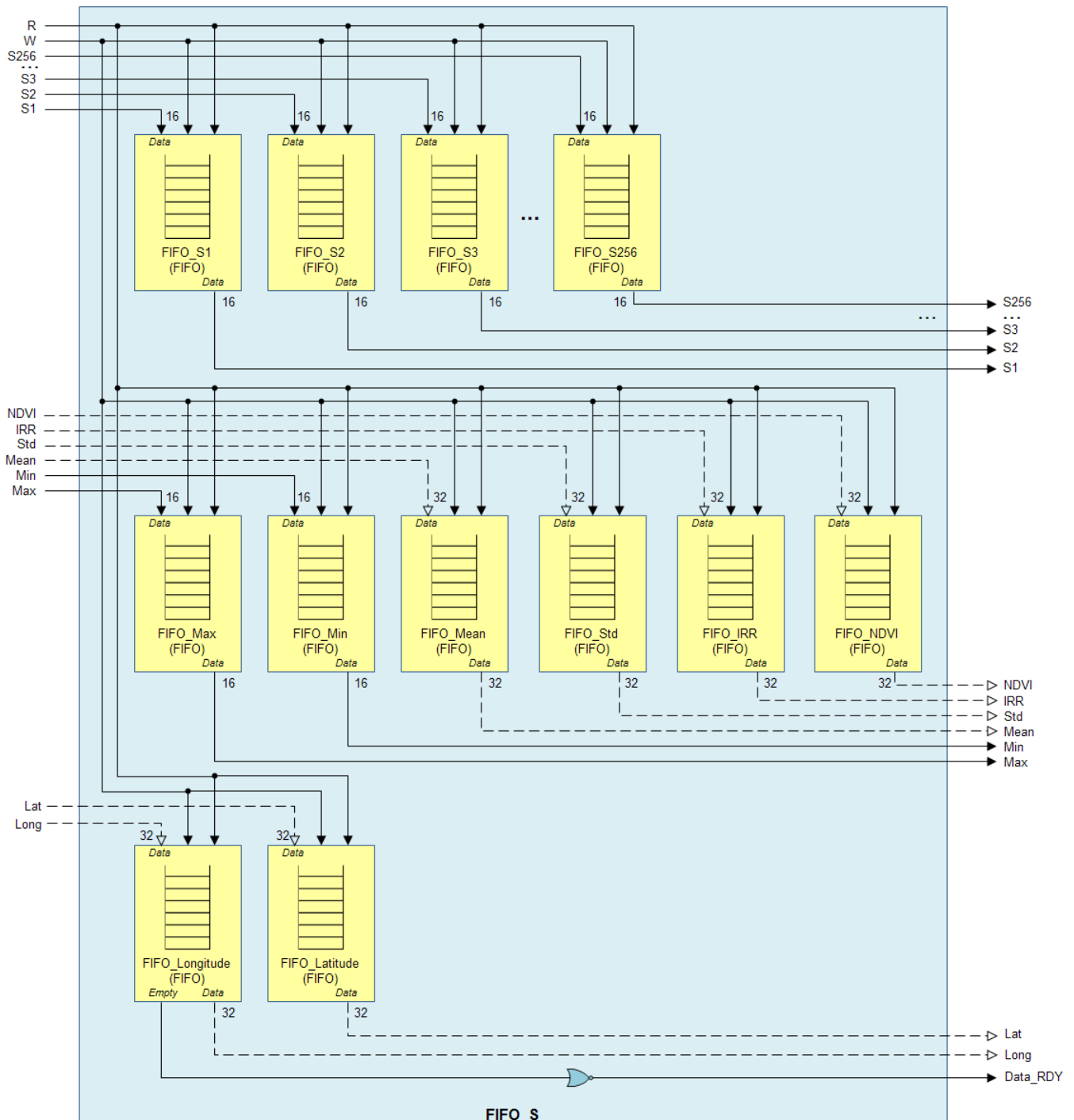


Figura 6.17: Diagrama del módulo FIFO_S.

6.4.3. Resultados del experimento.

Para comprobar el funcionamiento del sistema SpaceCube simulado, el cual ha sido detallado en las secciones anteriores, se utilizó una computadora de propósito general que actuará y realizará las funciones del sistema de computación global del satélite. Es decir, esta computadora será la encargada de enviar los espectros y las coordenadas en ráfagas aleatorias del mismo tipo de datos a las dos tarjetas FPGA, y de recibir el resultado final de las mismas.

La conexión a ésta se realizará mediante el uso de los diferentes conectores presentes en la tarjeta FPGA, siempre teniendo en cuenta las diferencias entre las versiones ML507 y ML510 en este aspecto, y las siguientes consideraciones:

- Ambas versiones poseen dos conectores DE-9, por lo que es posible utilizar conexiones RS-232 tanto para entrada de datos como para salida.
- La versión ML510 posee dos conectores RJ-45, por lo que es posible utilizar conexiones Ethernet tanto para entrada de datos como para salida. Sin embargo, la versión ML507 únicamente dispone de un conector de este tipo, por lo que este tipo de conexión sólo se puede utilizar o bien como entrada de datos, o bien como salida.
- La versión ML510 posee dos conectores PM, lo que ha posible la utilización de dos tarjetas puente con sendos conectores DB-25 de manera directa. De esta manera es posible utilizar conexiones RS-422 tanto para entrada de datos como para salida. Sin embargo, la versión ML507 únicamente dispone de un conector XGI para el mismo propósito, por lo que este tipo de conexión sólo se puede utilizar o bien como entrada de datos, o bien como salida. Por supuesto, es posible fabricar tarjetas puentes que permitan la adición de más de un conector DB-25, por lo que el número de conexiones se incrementarían. Sin embargo, ya se tratarían de tarjetas específicas que requerirían la modificación de la interfaz UART para contemplar la ampliación de conectores. Por este motivo, para el experimento se ha considerado una tarjeta puente con un único conector DB-25.

El hecho de que cada tarjeta FPGA posea diferentes conectores permite la posibilidad de utilizar diferentes configuraciones de conexiones, y por tanto, diferentes velocidades de transferencia de datos. Un estudio sobre cuáles son las configuraciones más apropiadas para el sistema propuesto en este experimento será realizado más adelante. Pero por el momento, y para detallar la manera en la que se realizan las comunicaciones, se considerarán las conexiones a un nivel general sin entrar en detalle los protocolos utilizados. De esta manera, la primera conexión en la primera tarjeta FPGA (*Smooth*) será utilizada para la recepción de datos (espectros y coordenadas), mientras que la segunda será utilizada para el envío del resultado generado (espectro suavizado). En cuanto a la segunda tarjeta FPGA (*Statistics*), la primera conexión será utilizada para la recepción de los resultados obtenidos por la primera tarjeta FPGA, mientras que la segunda conexión será utilizada para el envío de los cálculos realizados (valores máximo, mínimo, medio, desviación estándar, IR Ratio y NDVI), así como del espectro suavizado. Este procedimiento se encuentra detallado en la Sección 6.4.2.

Además, la comunicación de la computadora con las tarjetas FPGA se realizará a través de dos terminales que actuarán como interfaz, y que permitirá el envío de los datos originales (espectros y coordenadas) y la recepción de los resultados generados por el sistema SpaceCube simulado.

La Figura 6.18 muestra un diagrama de la conexión física de ambas tarjetas FPGA.

A un nivel general, la simulación del procesamiento a bordo se realizó de la siguiente manera:

1. A partir de los ficheros L0 e IGM correspondientes a un set de datos adquiridos por un avión o satélite, construir un fichero (al que denominaremos *SC*) en el que residirán todos los datos



Figura 6.18: Conexión física de las dos tarjetas FPGA emulando el sistema SpaceCube.

de los espectros y las coordenadas juntas, en ráfagas aleatorias pero siguiendo siempre el mismo orden. Es decir, un fichero SC podría estar formado por una secuencia de datos como la siguiente (donde S_i son todos los valores de los espectros juntos, y C_i la latitud y la longitud juntas), como por ejemplo: $S_1, S_2, C_1, S_3, S_4, C_2, S_5, C_3, C_4, C_5, C_6, C_7, S_6, S_7, \dots$. Para poder identificar el tipo de dato, cada parámetro C_i irá precedido por un byte con valor 0, y cada parámetro S_i por un byte 1. De esta forma, la primera tarjeta FPGA podrá distinguir los valores de espectro de los valores de coordenada. El motivo de realizarlo de esta manera, es porque la transferencia de datos computadora - primera tarjeta FPGA se realizan por el mismo conector y es necesario poder identificar el formato de cada valor. Si la conexión se realizara mediante dos puertos diferentes, entonces podría dedicarse uno a los valores de los espectros y otro al de las coordenadas, siendo la distinción desde la tarjeta FPGA más sencilla.

2. Ambas tarjetas FPGA son programadas con la metodología propuesta en la Sección 6.4.2, dedicándose la primera de ella al suavizado de los espectros (*Smooth*) y la segunda al cálculo de los valores estadísticos (*Statistics*). Una vez empiezan a trabajar, será posible distinguir tres fases:

- a) Fase de Inicio: Cálculo de los valores constantes por parte del procesador PowerPC. Estos valores serán $Bands$, $Bands_2$, $Bands2_1$, $Bands2_D$, $Bands2_D_1$ y UT_SQRT para el caso de la primera tarjeta, y $Bands$, $Bands_2$, $Bands2_1$, $Bands2_D$, $Bands2_D_1$, $BandIR$, y $BandR$ para el caso de la segunda tarjeta. Estos valores son enviados a los periféricos desarrollados para su correcto funcionamiento.
- b) Fase de Espera de Datos: Tras enviar los valores constantes a los periféricos, el procesador PowerPC crea dos hilos de ejecución (uno para envío de datos y otro para recepción), y queda a la espera de datos.
- c) Fase de Ejecución: Fase que transcurre desde la recepción de los datos, hasta la generación de resultados, y que se realiza de la siguiente manera:
 - 1) Desde la primera terminal se realiza el envío del fichero SC completo.
 - 2) El hilo de recepción del procesador de la primera tarjeta FPGA recibe progresivamente los datos, y los va almacenando en las memorias FIFO de entrada, de acuerdo al identificador del tipo de datos (0 para espectro, 1 para coordenadas).

- 3) En el momento en el que un espectro junto con su coordenada están disponibles (almacenados en sus correspondientes memorias FIFO), el periférico realiza el suavizado y almacena el resultado en las memorias FIFO de salida. Entonces activa la señal *Data_RDY*. Inmediatamente, sigue con el procesado de los siguientes datos (si hay disponibles en las memorias FIFO).
 - 4) El hilo de envío del procesador recibe la señal *Data_RDY*, lee el espectro suavizado de las memorias FIFO de salida y las envía a la otra tarjeta.
 - 5) El hilo de recepción del procesador de la segunda tarjeta FPGA recibe progresivamente los datos, y los va almacenando en las memorias FIFO de entrada.
 - 6) En el momento en el que un espectro junto con su coordenada están disponibles (almacenados en sus correspondientes memorias FIFO), el periférico realiza las estadísticas y almacena el resultado (junto con el espectro suavizado) en las memorias FIFO de salida. Entonces activa la señal *Data_RDY*.
 - 7) El hilo de envío del procesador recibe la señal *Data_RDY*, lee el los resultados estadísticos y el espectro suavizado de las memorias FIFO de salida y las envía a la computadora.
 - 8) La segunda terminal recibe los datos de la segunda tarjeta y los almacena como un fichero compuesto por una secuencia de registros DMR.
3. El resultado final es un fichero DMF (Sección 4.1.1.3) mediante el cual es posible cargar los datos en formato matriz DMF. Así, tras *el procesamiento a bordo y la recepción de los datos simulados*, se obtiene un producto georreferenciado con el cual ya es posible trabajar.

De esta manera, las cinco imágenes correspondientes al *dataset teórico* fueron utilizadas para el procesamiento del sistema SpaceCube simulado. Se requirió una previa preparación de los mismos, y a partir de los ficheros L0 e IGM se elaboraron los correspondientes ficheros *SC*, los cuales almacenaban ristas de espectros y coordenadas de manera aleatoria. Para una mejor comprensión de los datos proporcionados por cada una de las tarjetas, la Figura 6.19 muestra el resultado obtenido para un espectro cualquiera de la escena de Barrax.

Existen tres parámetros que deben ser controlados para el correcto y óptimo funcionamiento del sistema y la metodología propuestos: la ocupación de la FPGA, los tiempos de ejecución de los circuitos, y los tiempos de comunicación entre los diferentes dispositivos y módulos presentes en el diseño.

En cuanto a la ocupación de la FPGA, en un principio todos los módulos hardware presentados con anterioridad fueron diseñados e implementados de manera paralela. Sin embargo, esto requería un número demasiado grande de componentes que hacían que el sistema completo superara la capacidad de la FPGA. Diseños de módulos como FIFO_E, Mean, StdDev, FilterAdder, Maximum y Minimum tuvieron que ser realizados de nuevo, pero en versión secuencial, y mediante el uso de componentes adicionales como contadores, multiplexores, demultiplexores, comparadores y latches para conseguir este propósito. De esta manera, el número de recursos de la FPGA se redujo considerablemente, y fue posible mantener otros módulos como ToSingle256, Threshold128 y ToInt256 en su forma paralela, debido a su sencillez y a la liberación de espacio conseguido mediante la secuencialización de módulos más complejos. A pesar de la reducción del consumo de espacio en la FPGA, la secuencialización de los módulos más complejos conlleva tiempos de ejecución mayores. Mientras que en las versiones paralelas era posible realizar determinadas funciones para espectros de n bandas en una única iteración con un consumo de m ciclos de reloj, en las versiones secuenciales es necesario realizar n iteraciones que conllevan un consumo total de $n \times m$ ciclos de reloj. Mediante la herramienta Xilinx ISE se comprobó que los diseños propuestos, una vez compilados e implementados en ambas tarjetas FPGA, tenían una ocupación real de 75 % para la primera metodología (*Smooth*) y de 83 % para la segunda metodología (*Statistics*).

Respecto a los tiempos de ejecución, al realizar una secuencialización de los módulos más

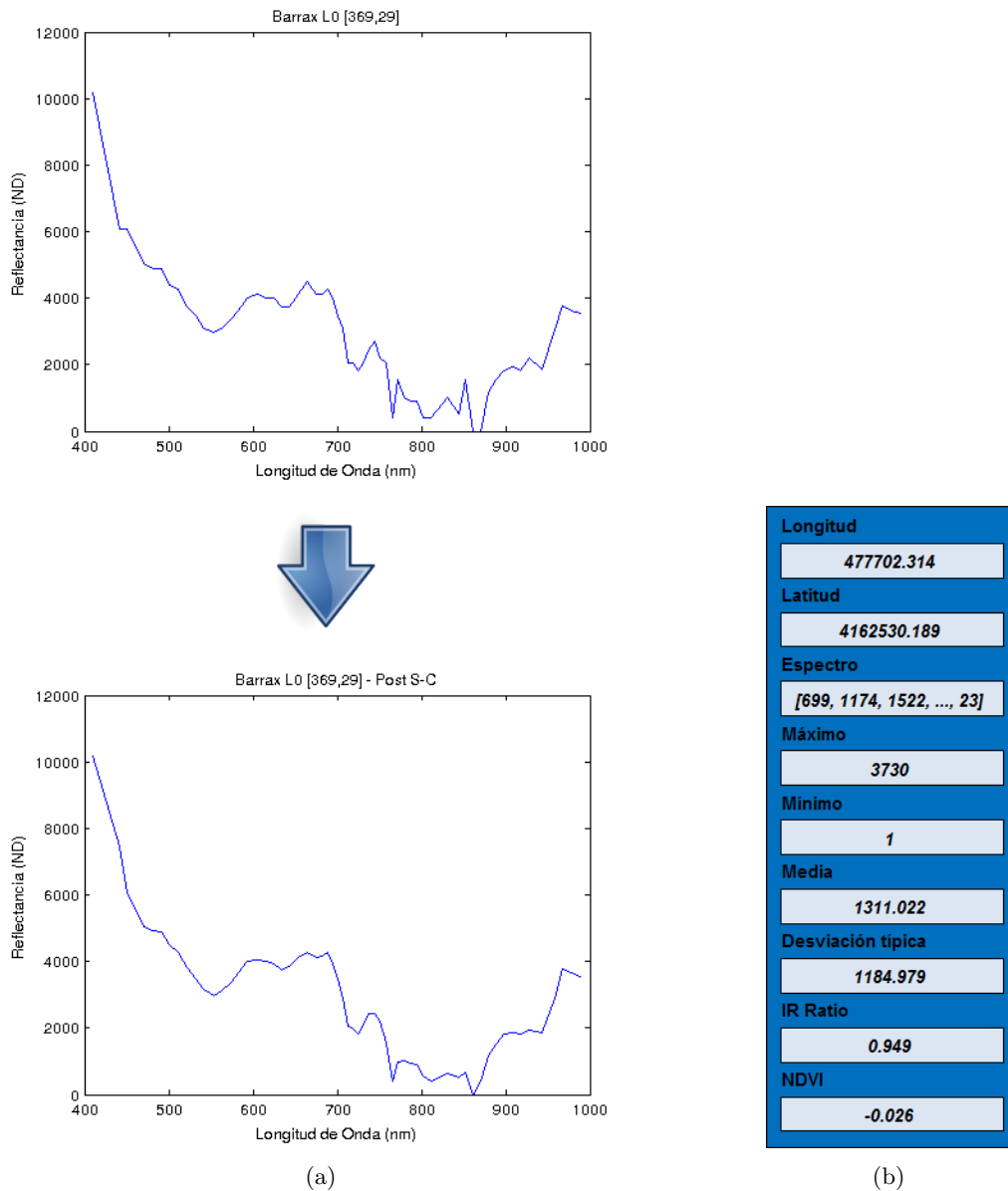


Figura 6.19: Ejemplo de resultado proporcionado por cada una de las tarjetas FPGA (Espectro (369,29) de la escena de Barrax).

(a) Suavizado del espectro realizado por la primera tarjeta FPGA (Smooth). (b) Cálculo de datos estadísticos en formato DMR realizado por la segunda tarjeta FPGA (Statistics).

complejos, éstos dependerán directamente del número de bandas que componen los espectros. La Tabla 6.5 presenta los tiempos de ejecución de cada uno de los módulos presentes en la metodología hardware en función del número de bandas que conforman el espectro (n). En ella, se han considerado módulos de diferentes niveles, de acuerdo a los niveles de abstracción propuestos. De esta manera, en el nivel más bajo, se situarían los componentes básicos, mientras que los módulos de niveles de abstracción superiores englobarían a los de niveles inferiores.

Mediante un análisis realizado con la herramienta Xilinx ISE, fue posible comprobar que la frecuencia de reloj máxima a la que puede ejecutarse las dos metodologías (Smooth y Statistics) era 230.787 MHz. Este aspecto es muy importante ya que frecuencias muy bajas implicarían sistemas más lentos, y repercutirían en los tiempos de procesamientos totales. Las frecuencias máximas adquiridas al reloj global utilizado por el sistema hardware propuesto, 100 MHz, por lo que el diseño de la metodología podría ser ejecutada en futuras actualizaciones o en versiones posteriores y más rápidas de tarjetas FPGA.

Componentes Básicos	
<i>Comparador (CMP)</i>	1 cr
<i>Comparador (CMP, single 32 bits)</i>	1 cr
<i>Counter</i>	1 cr
<i>Multiplexor (MUX)</i>	1 cr
<i>Demultiplexor (DEMUX)</i>	1 cr
<i>Latch</i>	1 cr
<i>Adder (single 32 bits)</i>	8 cr
<i>Subtractor</i>	1 cr
<i>Subtractor (single 32 bits)</i>	6 cr
<i>Multiplier (single 32 bits)</i>	10 cr
<i>Divider (single 32 bits)</i>	57 cr
<i>Raíz cuadrada (SQRT)</i>	57 cr
<i>Memoria FIFO</i>	Lectura: 1 cr, Escritura: 1 cr
<i>Registro (R)</i>	Lectura: 1 cr, Escritura: 1 cr
<i>IntToSingle</i>	6 cr
<i>SingleToInt</i>	6 cr
Módulos de primer nivel	
<i>ToDouble256</i>	6 cr
<i>ToInt256</i>	6 cr
<i>Kernel</i>	26 cr
<i>Filter</i>	$28 \times n + 1$ cr
<i>Mean</i>	$10 \times n + 57$ cr
<i>StdDev</i>	$28 \times n + 114$ cr
<i>Threshold</i>	2 cr
<i>Threshold128</i>	2 cr
<i>Filter Adder</i>	$10 \times n + 1$ cr
<i>Maximum</i>	$3 \times n + 1$ cr
<i>Minimum</i>	$3 \times n + 1$ cr
<i>IR_Ratio</i>	57 cr
<i>NDVI</i>	65 cr
Módulos de segundo nivel	
<i>FIFO_E</i>	$4 + n$ cr
<i>FIFO_S</i>	1 cr
<i>Smooth</i>	$76 \times n + 207$ cr
<i>Statistics</i>	$38 \times n + 177$ cr
Módulos de tercer nivel	
<i>Smooth (Periférico)</i>	$77 \times n + 212$ cr
<i>Statistics (Periférico)</i>	$39 \times n + 182$ cr

Tabla 6.5: *Tiempos de ejecución de cada uno de los módulos de la metodología hardware propuesta.*

Respecto a los tiempos de comunicación entre los diferentes dispositivos y módulos, este parámetro depende directamente de la manera en la que se realizan los tipos de conexiones entre las tarjetas y el sistema de computación, así como de los protocolos utilizados, siendo todos éstos ya definidos de manera general con anterioridad. Como se comentó previamente, el número de conectores de las tarjetas FPGA Virtex-5 varían de la versión ML507 a la versión ML510. Mientras que ambas poseen dos conectores DE-9 para conexiones RS-232, únicamente la versión ML510 posee dos conectores RJ-45 para conexiones Ethernet y dos conectores PM para conexiones RS-422. La versión ML507 únicamente posee un conector para cada tipo (siendo el conector XGI el análogo del PM de la versión ML510). Como para la elaboración del estudio se ha utilizado una tarjeta FPGA Virtex-5 ML507, el número de conectores será la principal limitación a la hora de establecer diferentes combinaciones de conexiones para la transferencia de los datos.

Cada una de las tarjetas FPGA posee varios puntos críticos que es necesario sincronizar para el correcto funcionamiento del sistema (Figura 6.20). El primero (PC1) es la frecuencia de llegada de los datos, que dependerá directamente del tipo de conexión utilizado. El segundo punto (PC2)

corresponde a la velocidad de transmisión del procesador a la FPGA, y viceversa, a través del bus central PLBv46. El tercer punto (PC3) está relacionado con el tiempo de procesamiento y frecuencia en las que trabajarán los periféricos propuestos (*Smooth* y *Statistics*), y que dependerá directamente del número de bandas que componen el espectro. El cuarto punto (PC4) corresponde a la frecuencia de salida de los datos, que dependerá directamente, tal y como ocurría para la frecuencia de entrada de datos, del tipo de conexión utilizado.

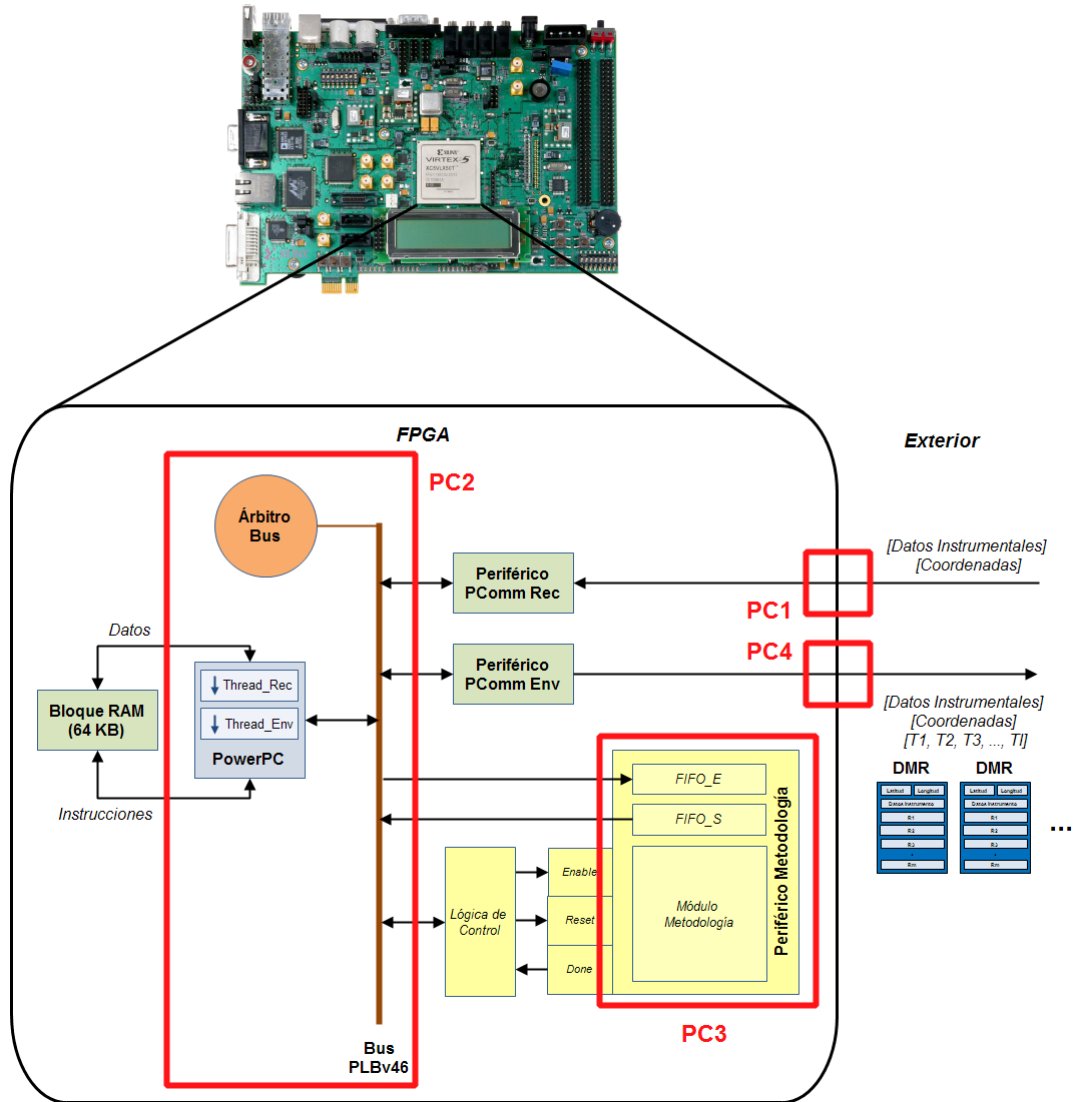


Figura 6.20: Señalización de los puntos críticos que son necesarios sincronizar para el correcto funcionamiento del sistema propuesto.

De esta forma, es posible determinar el número de registros DMR que cada uno de los puntos críticos es capaz de procesar mediante una serie de ecuaciones, en donde n representa el número de bandas del espectro, y siempre teniendo en cuenta que los valores de espectro están representados por 2 bytes, los valores de las coordenadas por 4 bytes, y que la entrada de datos al sistema (por la primera tarjeta FPGA) se realizan de manera separada para las coordenadas y los espectros, cada uno de ellos acompañado de un byte identificador:

1. *Frecuencia de entrada de datos (f_E)*: Representado en B/s, e incluyendo los datos de cabecera del paquete (según tipo de conexión). Para la metodología implementada en la primera tarjeta FPGA (*Smooth*) y para las conexiones RS-232 y RS-422 (en donde hay que considerar 10 bits por cada byte de dato real, debido a los bits de inicio y parada, es decir, un incremento de $10/8 = 1.25$ de información) estará determinado por:

$$N^{\circ}DMRs/seg(RS - 232, RS - 422, FPGA1) = \frac{f_E}{(1 + n \times 2) \times 1.25 + (1 + 2 \times 4) \times 1.25} \quad (6.13)$$

Donde el parámetro $(1 + n \times 2) \times 1.25$ representa la recepción de paquetes de espectro, mientras que el parámetro $(1 + 2 \times 4) \times 1.25$ representa la recepción de paquetes de coordenada.

Para esta misma tarjeta FPGA (*Smooth*), y para la conexión Ethernet, la frecuencia de entrada de datos queda determinada por paquetes individuales para espectros y coordenadas, al poder tener ambos un origen y orden de llegada diferentes:

$$N^{\circ}DMRs/seg(Ethernet, FPGA1) = \frac{f_E}{26 + 1 + n \times 2 + 26 + 1 + 2 \times 4} \quad (6.14)$$

Donde el parámetro $26 + 1 + n \times 2$ representa la recepción de paquetes de espectro, mientras que el parámetro $26 + 1 + 2 \times 4$ representa la recepción de paquetes de coordenada.

Igualmente, para la metodología implementada en la segunda tarjeta FPGA (*Statistics*), las ecuaciones que definen la frecuencia de entrada de datos son iguales que para la primera tarjeta FPGA, pero considerando que la transmisión de la información se realiza ya de manera ordenada y que no se incluyen los bytes adicionales de identificación del tipo de dato. Para las conexiones RS-232 y RS-422, dicha frecuencia quedará definida por:

$$N^{\circ}DMRs/seg(RS - 232, RS - 422, FPGA2) = \frac{f_E}{(n \times 2) \times 1.25 + (2 \times 4) \times 1.25} \quad (6.15)$$

Mientras que para esta misma tarjeta FPGA (*Statistics*), y para la conexión Ethernet, hay que tener en cuenta que el campo *Datos* de la trama correspondiente a este protocolo es variable, con un máximo de 1500 B. Toda la información referente al espectro y coordenadas, que ya ha sido sincronizada y procesada en conjunto por la primera tarjeta FPGA, puede ser recibida como un único paquete al tener un tamaño menor. Por tanto, la frecuencia de entrada estará determinada por:

$$N^{\circ}DMRs/seg(Ethernet, FPGA2) = \frac{f_E}{26 + n \times 2 + 2 \times 4} \quad (6.16)$$

2. *Frecuencia de transmisión del bus (f_B)*: Representado en MHz y considerando que el tiempo medio de transferencia de un dato de 32 bits es de 9 cr. Hay que tener en cuenta que la transferencia de un espectro, a pesar de que tienen un tamaño de 16 bits, se realiza de manera independiente (utilizando 32 bits y 9 cr por cada valor de espectro), mientras que las coordenadas, al tener un tamaño de 32 bits, se realiza en una iteración (9 cr) por cada valor de coordenada. Para la metodología implementada en la primera tarjeta FPGA (*Smooth*), la cantidad de datos de entrada y salida del periférico es la misma, por lo que número de DMRs capaz de transmitir por segundo será el mismo tanto antes de procesar, como después. En este caso, dicha frecuencia de transmisión estará determinados por:

$$N^{\circ}DMRs/seg(FPGA1) = \frac{f_B}{(n + 2) \times 9 + (n + 2) \times 9} \quad (6.17)$$

Donde el parámetro $(n + 2) \times 9$ representa la transmisión de una medición (espectro y coordenadas). Éste parámetro aparece repetido ya que hay que tener en cuenta la transmisión de la medición al periférico antes y después del procesado.

Para el caso de la segunda tarjeta FPGA (*Statistics*), el razonamiento es básicamente el mismo, pero hay que tener en cuenta que se envía el espectro a la Metodología, y se reciben valores estadísticos. Por tanto, la tasa de mediciones capaz de procesar está definida por:

$$N^{\circ}DMRs/seg(FPGA2) = \frac{f_B}{(n+2) \times 9 + (n+2) \times 9 + (1+1+1+1+1+1) \times 9} \quad (6.18)$$

Donde el parámetro $(n+2) \times 9 + (n+2) \times 9$ representa la doble transmisión de una medición (espectro y coordenadas), y donde el parámetro $(1+1+1+1+1+1) \times 9$ representa la transmisión de los resultados procedentes de los cálculos estadísticos (máximo, mínimo, media, desviación estándar e índices IR Ratio y NDVI, respectivamente), todos considerados como 32 bits por el ancho de banda del bus.

3. *Frecuencia de procesamiento (f_P)*: Representado en cr, consiste en el tiempo de procesamiento de cada una de las metodologías desarrolladas. Depende directamente de las latencias de los módulos que las conforman, las cuales fueron detalladas en la Tabla 6.5. De esta forma, la tasa de procesamiento de mediciones de la primera tarjeta FPGA (*Smooth*), de acuerdo a los tiempos de ejecución determinados, queda definida por:

$$N^{\circ}DMRs/seg(FPGA1) = \frac{f_P}{77 \times n + 212} \quad (6.19)$$

Mientras que para la segunda tarjeta FPGA (*Statistics*), la tasa de procesamiento de mediciones queda definida por:

$$N^{\circ}DMRs/seg(FPGA2) = \frac{f_P}{39 \times n + 182} \quad (6.20)$$

4. *Frecuencia de salida de datos (f_S)*: Representado en B/s, e incluyendo los datos de cabecera del paquete (según tipo de conexión). Para la metodología implementada en la primera tarjeta FPGA (*Smooth*) y para las conexiones RS-232 y RS-422 estará determinado por:

$$N^{\circ}DMRs/seg(RS - 232, RS - 422, FPGA1) = \frac{f_S}{(n \times 2) \times 1.25 + (2 \times 4) \times 1.25} \quad (6.21)$$

Donde el parámetro $(n \times 2) \times 1.25$ representa el envío de de paquetes de espectro, mientras que el parámetro $(2 \times 4) \times 1.25$ representa el envío de paquetes de coordenada.

Para esta misma tarjeta FPGA (*Smooth*), y para la conexión Ethernet, hay que tener en cuenta que el campo *Datos* de la trama correspondiente a este protocolo es variable, con un máximo de 1500 B. Toda la información referente al espectro y coordenadas, que ya ha sido sincronizada y procesada en conjunto por la esta tarjeta FPGA, puede ser enviada como un único paquete al tener un tamaño menor. Por tanto, la frecuencia de salida estará determinada por:

$$N^{\circ}DMRs/seg(Ethernet, FPGA1) = \frac{f_S}{26 + n \times 2 + 2 \times 4} \quad (6.22)$$

Donde el parámetro $26 + n \times 2$ representa el envío de paquetes de espectro, mientras que el parámetro $26 + 2 \times 4$ representa el envío de paquetes de coordenada.

Igualmente, para la metodología implementada en la segunda tarjeta FPGA (*Statistics*), las ecuaciones que definen la frecuencia de salida para las conexiones RS-232 y RS-422 quedan representadas por:

$$\begin{aligned} N^{\circ}DMRs/seg(RS - 232, RS - 422, FPGA2) = \\ = \frac{f_S}{(n \times 2) \times 1.25 + (2 \times 4) \times 1.25 + (2 + 2 + 4 + 4 + 4 + 4) \times 1.25} \end{aligned} \quad (6.23)$$

Donde el parámetro $(n \times 2) \times 1.25 + (2 \times 4) \times 1.25$ representa el envío de una medición (espectro y coordenadas), y donde el parámetro $(2 + 2 + 4 + 4 + 4 + 4) \times 1.25$ representa el envío de los resultados procedentes de los cálculos estadísticos (máximo, mínimo, media, desviación estándar e índices IR Ratio y NDVI, respectivamente).

Para esta misma tarjeta FPGA (*Statistics*), y para la conexión Ethernet, hay que tener en cuenta que el campo *Datos* de la trama correspondiente a este protocolo es variable, con un máximo de 1500 B. Toda la información referente al espectro y coordenadas, que ya ha sido sincronizada y procesada en conjunto por la esta tarjeta FPGA, junto con la información de los cálculos estadísticos, pueden ser enviadas como un único paquete al tener un tamaño menor. Por tanto, la frecuencia de salida estará determinada por:

$$N^{\circ}DMRs/seg(Ethernet, FPGA2) = \frac{f_s}{26 + n \times 2 + 2 \times 4 + 2 + 2 + 4 + 4 + 4 + 4} \quad (6.24)$$

Donde el parámetro $n \times 2 + 2 \times 4$ representa el envío de una medición (espectro y coordenadas), y donde el parámetro $2 + 2 + 4 + 4 + 4 + 4$ representa el envío de los resultados procedentes de los cálculos estadísticos (máximo, mínimo, media, desviación estándar e índices IR Ratio y NDVI, respectivamente).

A partir de las ecuaciones anteriores es posible realizar un cálculo de la capacidad de procesamiento de mediciones del sistema SpaceCube propuesto, en función del número de bandas que conforman el espectro. Dicho cálculo se encuentra de manera gráfica para la primera tarjeta FPGA en la Figura 6.21(a), y para la segunda tarjeta FPGA en la 6.21(b). El cálculo de la capacidad para cada uno de los puntos críticos del sistema es muy importante, ya que permite establecer qué tipos de conexiones (tanto de entrada como de salida) es posible utilizar, así como encontrar las limitaciones respecto a las comunicaciones entre los dispositivos y módulos.

Para la realización de dicho cálculo no se han utilizado todas las conexiones con todas las velocidades posibles, sino que se han considerado únicamente las más representativas:

- *RS-232 9600 Baud*: Conexión de baja velocidad usando el protocolo RS-232 a 9600 Baud, velocidad estándar.
- *RS-232 19200 Baud*: Conexión de baja velocidad usando el protocolo RS-232 a 19200 Baud, velocidad máxima soportada por algunos dispositivos y terminales.
- *RS-232 128000 Baud*: Conexión de baja velocidad usando el protocolo RS-232 a 128000 Baud, de las máximas velocidades soportadas por este protocolo.
- *RS-232 230400 Baud*: Conexión de baja velocidad usando el protocolo RS-232 a 203400 Baud, de las máximas velocidades soportadas por este protocolo.
- *Ethernet 10 Mbps*: Conexión de alta velocidad usando el protocolo Ethernet a 10 Mbps, mínima velocidad.
- *Ethernet 100 Mbps*: Conexión de alta velocidad usando el protocolo Ethernet a 100 Mbps.
- *Ethernet 1000 Mbps*: Conexión de alta velocidad usando el protocolo Ethernet a 1000 Mbps, máxima velocidad.
- *RS-422 10 Mbps*: Conexión de alta velocidad usando el protocolo RS-422 a 10 Mbps, máxima velocidad.

El cálculo a través del protocolo RS-232 a velocidades superiores (460800 Baud y 921600 Baud) ha quedado excluido debido a incompatibilidades física entre el reloj del sistema (100 MHz) y el de la interfaz UART, al ser ésta asíncrona [MI03].

De esta manera, en la Figura 6.21, los diferentes puntos críticos del sistema y los tipos de conexiones utilizados han sido clasificados en diferentes esquemas de colores para una mejor visualización:

- *Conexiones de baja velocidad (RS-232)*: Color rojo para la tasa de entrada de datos; color naranja para la tasa de salida de datos.
- *Conexiones de alta velocidad (Ethernet)*: Color verde oscuro para la tasa de entrada de datos; color verde claro para la tasa de salida de datos.
- *Conexiones de alta velocidad (RS-422)*: Color púrpura para la tasa de entrada de datos; color rosa para la tasa de salida de datos.
- *Transferencias a través del bus central*: Color marrón.
- *Procesamiento de la metodología*: Color azul oscuro con línea continua para frecuencia de 100 MHz (caso real); color azul oscuro con línea punteada para frecuencia de 230 MHz (frecuencia máxima soportada, para futuras ampliaciones).

A la hora de seleccionar las conexiones de entrada y salida del sistema hay que tener especial cuidado en la velocidad de procesamiento de los demás componentes. Para garantizar un correcto funcionamiento, la frecuencia de recepción de datos nunca debe ser superior a la frecuencia de salida de los mismos (ya procesados), ni a la capacidad de procesamiento de los demás componentes implicados (transmisión a través del bus y velocidad de procesamiento del periférico). De esta manera, es posible comprobar, para ambas tarjetas FPGA, que conexiones Ethernet a 100 Mbps y 1000 Mbps son totalmente incompatibles con el sistema propuesta, pues la frecuencia de llegada de datos sería tan elevado que tanto el bus central como el periférico se saturarían (los buffers de entrada se llenarían completamente). Sin embargo, sí que sería posible utilizar el resto de conexiones, ya que la frecuencia de datos sería menor y tanto el bus, como el periférico, tendrían suficiente tiempo para procesar los datos a medida que llegan.

- *Selección del tipo de conexión para entrada de datos*: Para la selección de un tipo de conexión de entrada (excluyendo por supuesto las conexiones Ethernet a 100 y 1000 Mbps), es necesario tener en cuenta que dependiendo de la velocidad proporcionada, el bus y el periférico se encontrarán más o menos ociosos entre la finalización del procesamiento o transferencia de una medición y la recepción de la siguiente. Básicamente, este tiempo de ocio, representado como la diferencia de mediciones a procesar o transferir respecto a la entrada de datos, puede representarse de manera gráfica en función del número de componentes que conforman el espectro (Figura 6.22).

De esta manera, es posible comprobar que las conexiones de baja velocidad no aprovechan el rendimiento del sistema, pues el bus central y la metodología están ociosas casi al 100 % durante la ejecución (a la espera de datos). Este porcentaje de espera se decreta ligeramente cuando se utilizan conexiones de alta velocidad, aunque no nunca es menor de 90 % para el caso del bus, ni del 80 % - 55 % para el caso del periférico. Respecto a los tiempos de espera del bus, ambos son similares para ambas tarjetas FPGA, decreciente a medida que el número de componentes que conforman el espectro se incrementa. Sin embargo, este comportamiento es totalmente contrario, ascendente, si se utiliza una conexión RS-232 o RS-422 en la segunda tarjeta FPGA, debido a que el tipo de medición resultante es de mayor tamaño (posee campos adicionales), y que el tiempo dedicado a la transferencia de la misma a través del bus, a medida que se incrementa el número de componentes del espectro, es menor que el tiempo dedicado en su recepción. En cuanto a los tiempos de espera del periférico, para la metodología implementada en la segunda

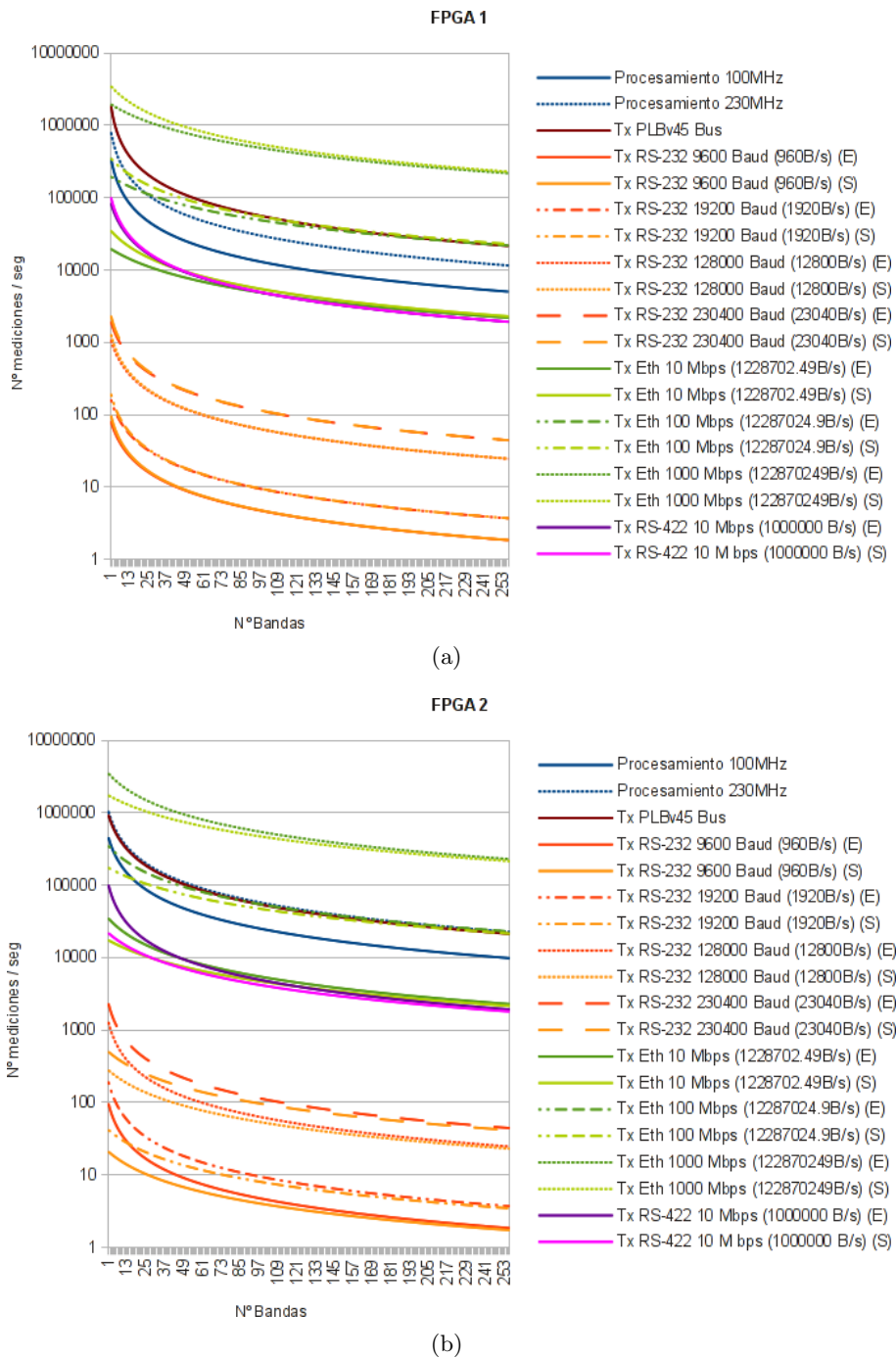


Figura 6.21: *Tasas de procesamiento y transmisión de mediciones para los puntos críticos del sistema. (a) Primera tarjeta FPGA (Smooth). (b) Segunda tarjeta FPGA (Statistics).*

tarjeta FPGA son mayores que en la primera tarjeta FPGA, pues el circuito implementado es más sencillo y con una ejecución más rápida.

- Selección del tipo de conexión para salida de datos: La selección del tipo de conexión de salida es bastante delicado en este caso, debido principalmente a una determinada circunstancia derivada de la segunda tarjeta FPGA.

Si se considera la primera tarjeta FPGA (Figura 6.21(a)), es posible comprobar que los tiempos de entrada de datos siempre van a ser superiores a los de salida, debido a que el formato de la medición a transmitir es el mismo, pero con campos adicionales de identificación de tipo de dato para la entrada. Por tanto, es posible utilizar el mismo tipo de conexión tanto para la entrada de datos, como para la salida.

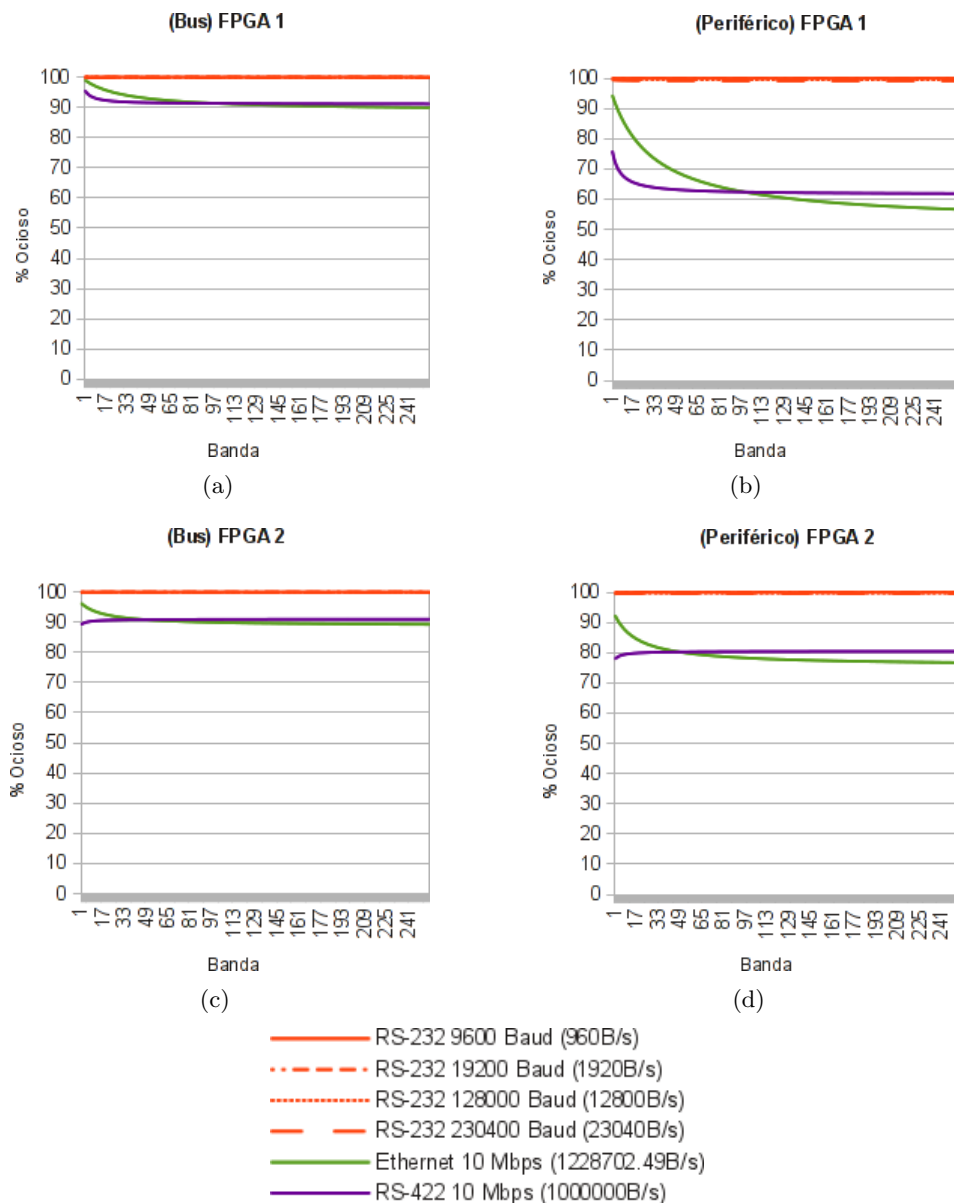


Figura 6.22: Porcentaje de tiempo ocioso para cada uno de los dispositivos del sistema en función del tipo de conexión de entrada de datos seleccionado.

(a) Bus (FPGA 1). (b) Periférico (FPGA 1). (c) Bus (FPGA 2). (d) Periférico (FPGA 2).

Sin embargo, y respecto a la segunda tarjeta FPGA (Figura 6.21(b)), la frecuencia de salida de datos es mayor que la de entrada, debido a que el resultado del procesamiento implica campos adicionales (máximo, mínimo, media, desviación estándar, e índices IR Ratio y NDVI). Esto produce una desincronización respecto a la frecuencia de entrada de datos, ya que no es admisible que la frecuencia con la que se reciben dichos datos sea mayor que con la que se envían ya procesados (en el sistema no pueden entrar más datos de los que salen, se saturarían los buffers de entrada en caso contrario). Por tanto, para el caso especial de esta tarjeta FPGA, no se puede utilizar las mismas velocidades de conexión para las entradas y las salidas, sino que siempre la velocidad de transferencia de datos de salida debe ser mayor que el de entrada. Por tanto, si se supone que se ha utilizado un tipo de conexión en la primera tarjeta FPGA, la segunda tarjeta FPGA debe utilizar el mismo tipo de conexión para la entrada de datos. Por tanto, a la hora de seleccionar un tipo de conexión para la salida de datos de esta segunda tarjeta FPGA, existen dos posibilidades:

1. Disminuir las velocidades de transmisión del sistema SpaceCube entero (entrada de la primera y segunda tarjeta FPGA, y salida de la primera tarjeta FPGA), manteniendo

la velocidad de transmisión de datos de salida de la segunda tarjeta FPGA.

2. Mantener las velocidades de transmisión del sistema SpaceCube entero (entrada de la primera y segunda tarjeta FPGA, y salida de la primera tarjeta FPGA) tal cual, y aumentar la velocidad de transmisión de datos de salida de la segunda tarjeta FPGA. Ésta es la posibilidad recomendada y considerada en el estudio, aunque es necesario remarcar que el dispositivo receptor de los resultados del sistema SpaceCube debe soportar la frecuencia de transmisión de datos de salida derivada de la segunda tarjeta FPGA.

Por tanto, el número de combinaciones que es posible realizar con los tipos de conexiones considerados, para la entrada y salida de datos, es diverso, pero también limitado debido a las consideraciones anteriores. De todas éstas, la limitación más importante a la hora de realizar combinaciones es el hecho de determinar una frecuencia de salida de datos menor que el de entrada. Esto implica la permanente existencia de un determinado porcentaje de tiempo ocioso para la conexión de salida, cuya duración dependerá directamente de la combinación seleccionada. La Tabla 6.6 muestra la relación entre la frecuencia de entrada de datos y la de salida. Los valores numéricos representan cuántas veces la frecuencia de salida es más rápida que la frecuencia de entrada. Celdas sin valores numéricos indican que la frecuencia de entrada de la combinación correspondiente sería más rápida que la salida, lo que imposibilitaría el correcto funcionamiento del sistema al saturar los buffers. Los porcentajes marcados en azul indican las combinaciones más adecuadas, ya que el tiempo de espera de la conexión de salida sería menor. Además, se ha añadido como conexión de salida la Ethernet 100 Mbps, ya que puede combinar correctamente con las conexiones de entrada Ethernet 10 Mbps y RS-422 10 Mbps. Sin embargo, este tipo de conexión presenta velocidades de transferencia de datos muy elevadas, por lo que es necesario determinar si el sistema o dispositivo receptor de los mismos soporta tales frecuencias.

En cuanto a las combinaciones de conexiones posibles, existe el caso de que con una conexión de entrada RS-232 230400 Baud, tanto la conexión de salida Ethernet 10 Mbps como la RS-422 10 Mbps serían adecuadas y permitirían un correcto funcionamiento del sistema. Como ambas presentan un porcentaje de ocio similar, se optó por seleccionar como salida la conexión Ethernet 10 Mbps en el resto del análisis, ya que se trata de un tipo de conexión más extendido y además se encuentra incluida en las tarjetas. Por tanto, las combinaciones de conexiones de entrada (E) y salida (S) seleccionadas para el resto del análisis son las siguientes:

- *Combinación 1 (RS232-9600-19200):*
 - FPGA 1 - (E): RS-232 9600 Baud, (S): RS-232 9600 Baud.
 - FPGA 2 - (E): RS-232 9600 Baud, (S): RS-232 19200 Baud.
- *Combinación 2 (RS232-19200-128000):*
 - FPGA 1 - (E): RS-232 19200 Baud, (S): RS-232 19200 Baud.
 - FPGA 2 - (E): RS-232 19200 Baud, (S): RS-232 128000 Baud.
- *Combinación 3 (RS232-128000-230400):*
 - FPGA 1 - (E): RS-232 128000 Baud, (S): RS-232 128000 Baud.
 - FPGA 2 - (E): RS-232 128000 Baud, (S): RS-232 230400 Baud.
- *Combinación 4 (RS232-230400-Eth-10):*
 - FPGA 1 - (E): RS-232 230400 Baud, (S): RS-232 230400 Baud.
 - FPGA 2 - (E): RS-232 230400 Baud, (S): Ethernet 10 Mbps.

Cuprite								
Salida		RS-232				Ethernet		RS-422
		9600 Baud	19200 Baud	128000 Baud	230000 Baud	10 Mbps	100 Mbps	10 Mbps
RS-232	9600 Baud	—	1.916	12.773	22.992	1182.769	11827.689	997.899
	19200 Baud	—	—	6.387	11.496	591.384	5913.845	498.950
	128000 Baud	—	—	—	1.724	88.708	887.077	74.842
	230400 Baud	—	—	—	—	49.282	492.820	41.579
Ethernet	10 Mbps	—	—	—	—	—	9.602	—
RS-422	10 Mbps	—	—	—	—	1.135	11.355	—
Barrax								
Salida		RS-232				Ethernet		RS-422
		9600 Baud	19200 Baud	128000 Baud	230000 Baud	10 Mbps	100 Mbps	10 Mbps
RS-232	9600 Baud	—	1.737	11.579	20.842	965.590	9655.899	904.605
	19200 Baud	—	—	5.789	10.421	482.795	4827.949	452.303
	128000 Baud	—	—	—	1.563	72.419	724.192	67.845
	230400 Baud	—	—	—	—	40.233	402.329	37.692
Ethernet	10 Mbps	—	—	—	—	—	8.876	—
RS-422	10 Mbps	—	—	—	—	—	9.270	—
Mediterráneo								
Salida		RS-232				Ethernet		RS-422
		9600 Baud	19200 Baud	128000 Baud	230000 Baud	10 Mbps	100 Mbps	10 Mbps
RS-232	9600 Baud	—	1.444	9.630	17.333	690.901	6909.014	752.315
	19200 Baud	—	—	4.815	8.667	345.451	3454.507	376.157
	128000 Baud	—	—	—	1.300	51.818	518.176	56.424
	230400 Baud	—	—	—	—	28.788	287.876	31.346
Ethernet	10 Mbps	—	—	—	—	—	7.959	—
RS-422	10 Mbps	—	—	—	—	—	6.633	—
GMéxico								
Salida		RS-232				Ethernet		RS-422
		9600 Baud	19200 Baud	128000 Baud	230000 Baud	10 Mbps	100 Mbps	10 Mbps
RS-232	9600 Baud	—	1.916	12.773	22.992	1182.769	11827.689	997.899
	19200 Baud	—	—	6.387	11.496	591.384	5913.845	498.950
	128000 Baud	—	—	—	1.724	88.708	887.077	74.842
	230400 Baud	—	—	—	—	49.282	492.820	41.579
Ethernet	10 Mbps	—	—	—	—	—	9.602	—
RS-422	10 Mbps	—	—	—	—	1.135	11.355	—
Cantera								
Salida		RS-232				Ethernet		RS-422
		9600 Baud	19200 Baud	128000 Baud	230000 Baud	10 Mbps	100 Mbps	10 Mbps
RS-232	9600 Baud	—	1.740	11.602	20.883	969.329	9693.287	906.385
	19200 Baud	—	—	5.801	10.442	484.664	4846.644	453.193
	128000 Baud	—	—	—	1.566	72.700	726.997	67.979
	230400 Baud	—	—	—	—	40.389	403.887	37.766
Ethernet	10 Mbps	—	—	—	—	—	8.889	—
RS-422	10 Mbps	—	—	—	—	—	9.306	—

Tabla 6.6: Tiempos de ocio de cada una de las conexiones de salida de acuerdo a la conexión de entrada seleccionada.

- *Combinación 5 (Eth-10-100):*
 - FPGA 1 - (E): Ethernet 10 Mbps, (S): Ethernet 10 Mbps.
 - FPGA 2 - (E): Ethernet 10 Mbps, (S): Ethernet 100 Mbps.
- *Combinación 6 (RS422-10-Eth-100):*
 - FPGA 1 - (E): RS-422 10 Mbps, (S): RS-422 10 Mbps.
 - FPGA 2 - (E): RS-422 10 Mbps, (S): Ethernet 100 Mbps.

Por supuesto, es posible utilizar conexiones de salida que impliquen una mayor velocidad. El sistema continuaría funcionando correctamente, sólo que dichas conexiones de salida implicarían mayores tiempo de espera entre la salida completa del dato, y la recepción del siguiente.

A partir de las Ecuaciones 6.13 - 6.20 es posible determinar los tiempos de ejecución de cada una de las imágenes consideradas. Pero hay que tener en cuenta que el procesamiento dentro del sistema SpaceCube es solapado. Es decir, a medida que se transmite una medición por la conexión de entrada y por el bus central, existe otra siendo procesada por el periférico (por la metodología correspondiente, según la tarjeta FPGA), y otra siendo transferida a través del bus central y la conexión de salida. De esta misma manera, a medida que ciertas mediciones están siendo transferidas y procesadas en la primera tarjeta FPGA, otras que ya han sido consideradas por ésta están siendo transferidas y procesadas también, y a la misma vez, por la segunda tarjeta FPGA. De ahí que el tiempo de ejecución total del sistema SpaceCube completo no sea la suma directa de los tiempos de ejecución de ambas tarjetas de manera independiente.

La Tabla 6.7 muestra, para cada una de las imágenes, el tiempo en el que una única medición es procesada. Es decir, desde que se recibe el primer byte en la primera tarjeta FPGA hasta que el último byte sale de ésta (primera columna); desde que se recibe el primer byte en la segunda tarjeta FPGA hasta que el último byte sale de ésta (segunda columna); y, desde un punto de vista global, desde que se recibe el primer byte en el sistema SpaceCube (desde que se recibe en la primera tarjeta FPGA) hasta que el último byte sale de ésta (hasta que el último byte sale de la segunda tarjeta FPGA). La Tabla 6.8 muestra esta misma información, pero referentes a las imágenes completas. Es decir, los tiempos empleados por cada una de las tarjetas FPGA, y por el sistema SpaceCube completo, en procesar todas las mediciones de cada una de las imágenes consideradas.

Para poder determinar en qué aspecto las diferentes combinaciones de conexiones influyen en los tiempos de procesamiento del sistema SpaceCube, se realizó un cálculo del porcentaje de ejecución global empleado tanto en la transmisión de datos, como en el procesamiento de las mediciones empleado por la metodología. La Tabla 6.9 presenta dichos porcentajes respecto a la transferencia de mediciones por parte de las conexiones, tanto de entrada como de salida (T. Conexiones); respecto a la transferencia de mediciones (pre-procesadas y post-procesadas) por parte del bus central (T. Bus); respecto a las comunicaciones en general, es decir, por parte de las conexiones y del bus central (T. Comunicaciones = T. Conexiones + T. Bus); y respecto al procesamiento por parte de la metodología (T. Procesamiento). Estos porcentajes reflejan el tiempo global dedicado a cada una de las actividades, a un nivel general de sistema SpaceCube, es decir, implicando ambas tarjetas FPGA.

Finalmente, tras realizar el estudio de las comunicaciones implicadas entre las dos tarjetas FPGA, se muestran los datos obtenidos tras el procesamiento de las metodologías (Figuras 6.23, 6.24, 6.25, 6.26 y 6.27). En ellas se visualizan de manera gráfica los resultados recibidos del sistema SpaceCube, en donde se representan, respectivamente y como matrices DMF independientes, los valores máximo, mínimo, media, desviación estándar, IR Ratio y NDVI de cada registro DMR, siempre siguiendo la política de visualización DMF (Sección 4.1.1.2). Por supuesto, cabe recordar que estos resultados, proporcionados de manera directa por la segunda tarjeta FPGA, fueron previamente suavizados mediante la técnica S-C, implementada en la primera tarjeta FPGA.

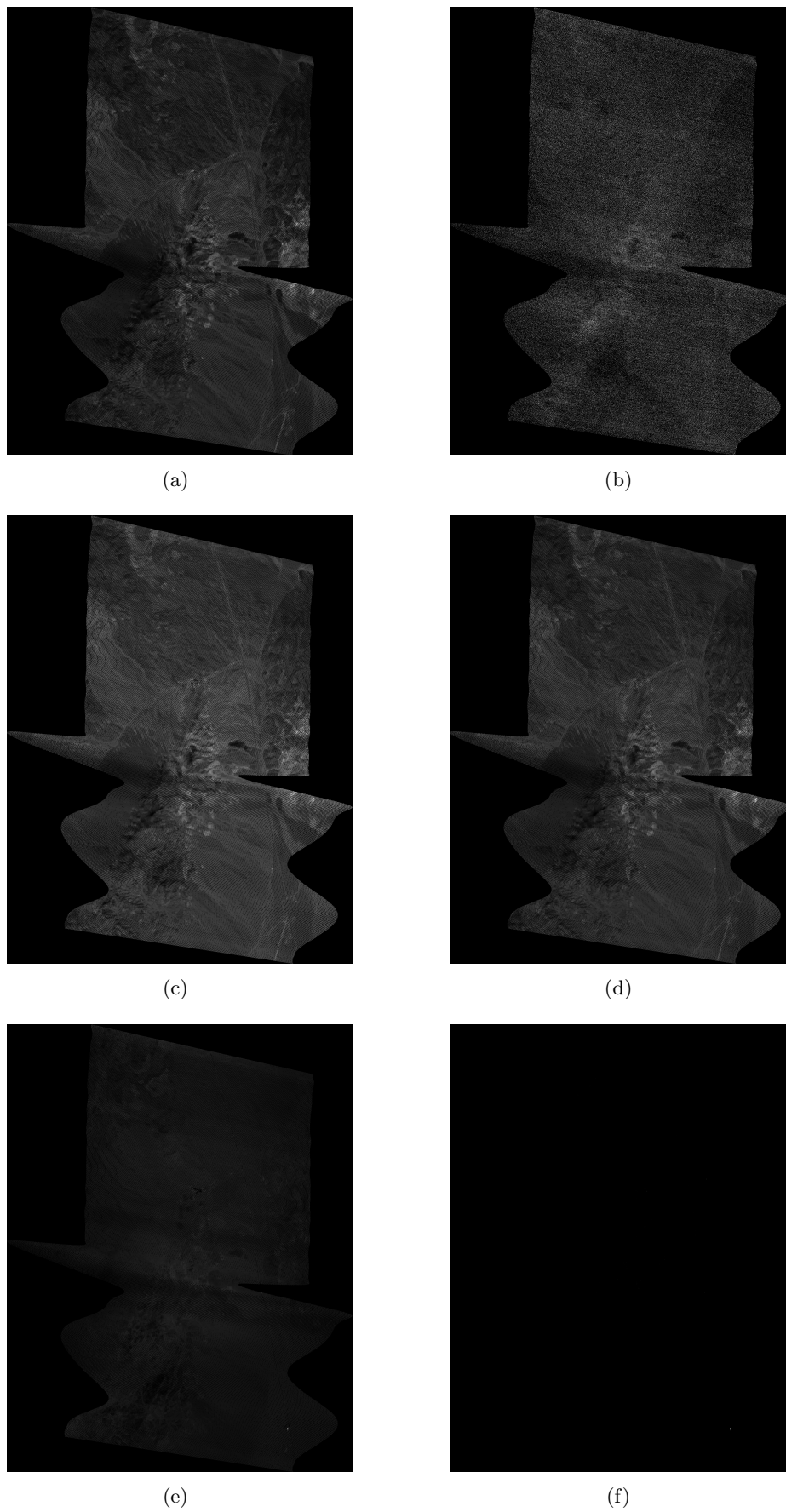


Figura 6.23: Visualización en formato DMF de los resultados obtenidos con el sistema SpaceCube para la escena de Cuprite. (a) Máximo espectral. (b) Mínimo espectral. (c) Media espectral. (d) Desviación típica espectral. (e) Índice IR Ratio. (f) Índice NDVI.

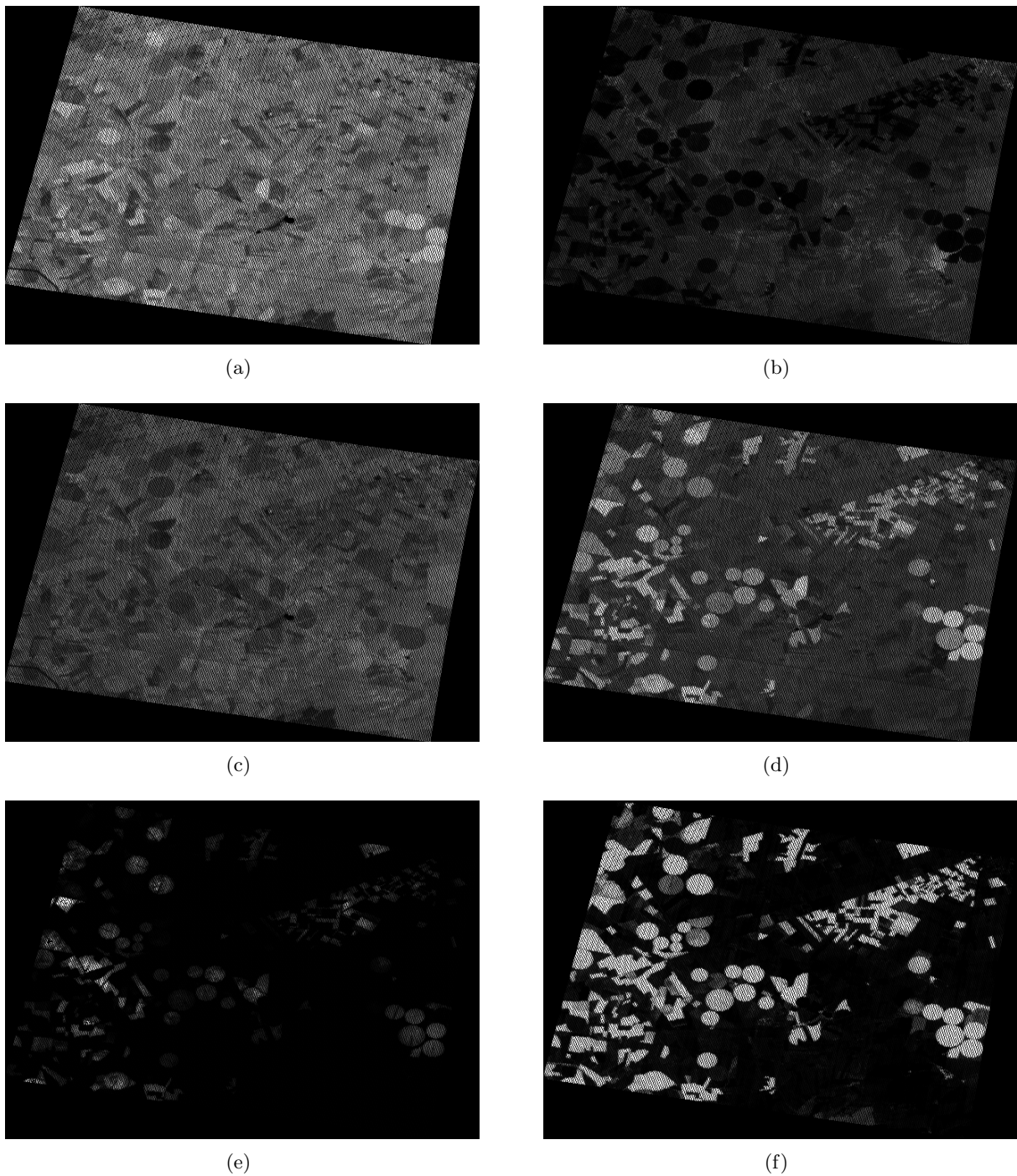


Figura 6.24: Visualización en formato DMF de los resultados obtenidos con el sistema SpaceCube para la escena de Barrax. (a) Máximo espectral. (b) Mínimo espectral. (c) Media espectral. (d) Desviación típica espectral. (e) Índice IR Ratio. (f) Índice NDVI.

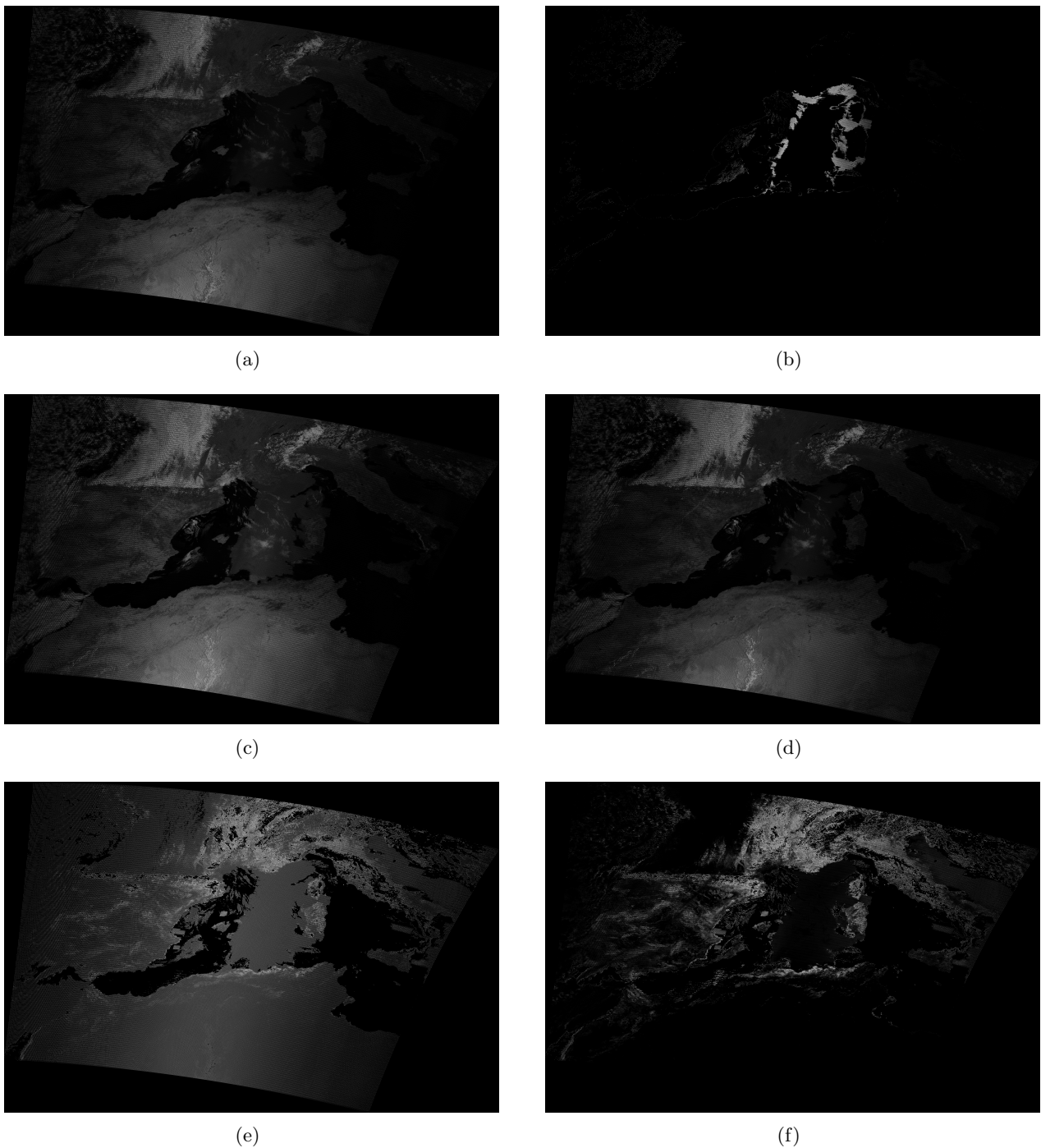


Figura 6.25: Visualización en formato DMF de los resultados obtenidos con el sistema SpaceCube para la escena de Mediterráneo. (a) Máximo espectral. (b) Mínimo espectral. (c) Media espectral. (d) Desviación típica espectral. (e) Índice IR Ratio. (f) Índice NDVI.

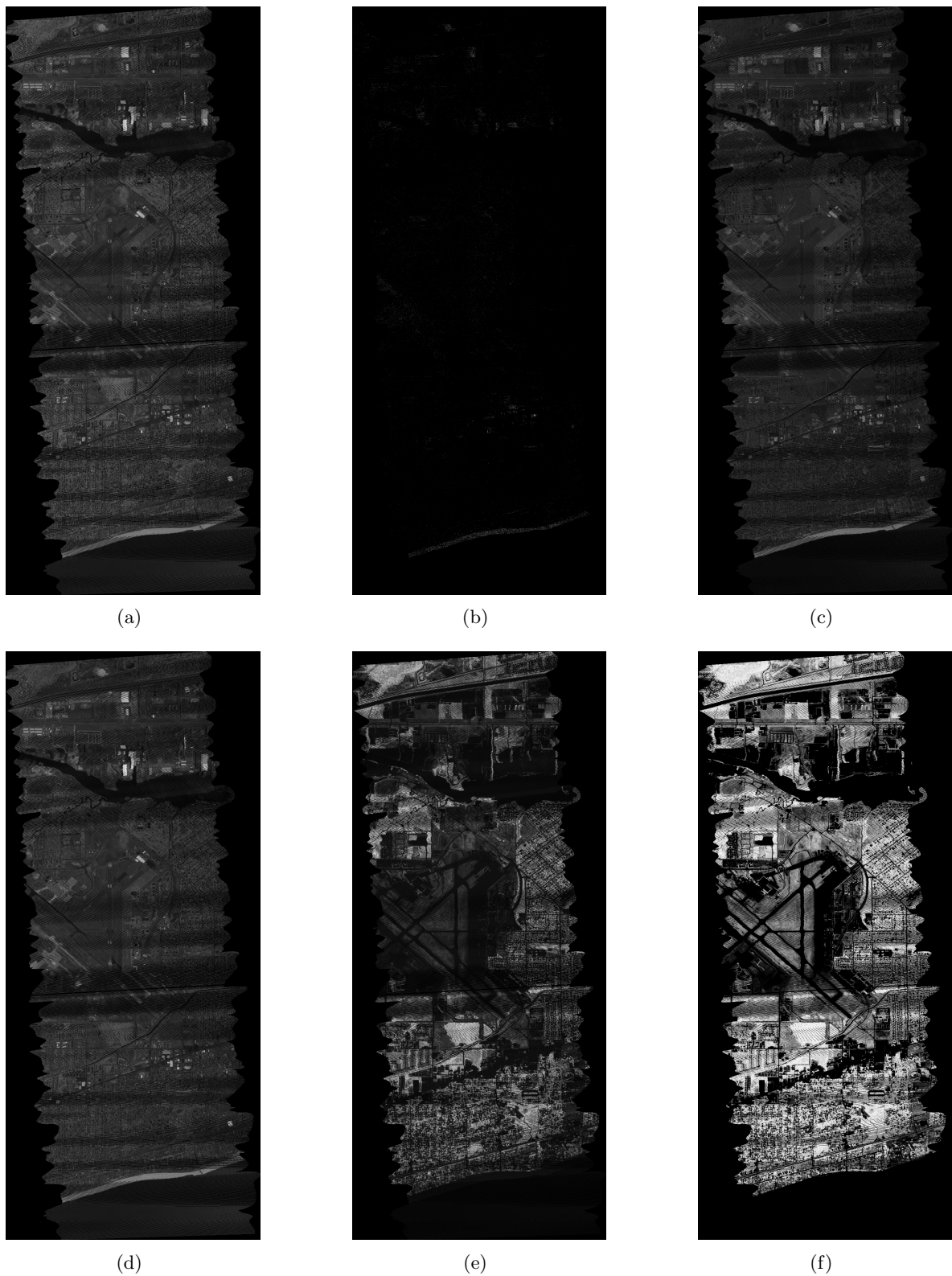
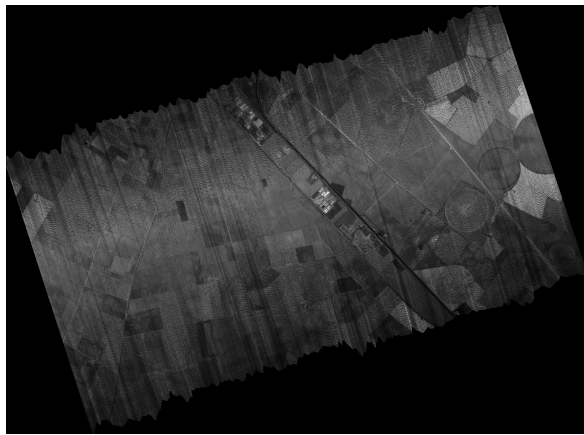
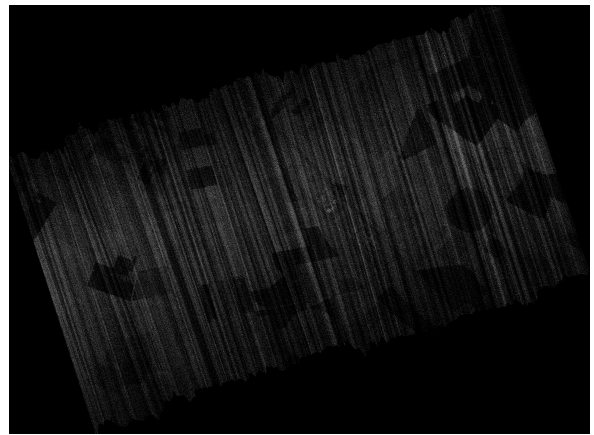


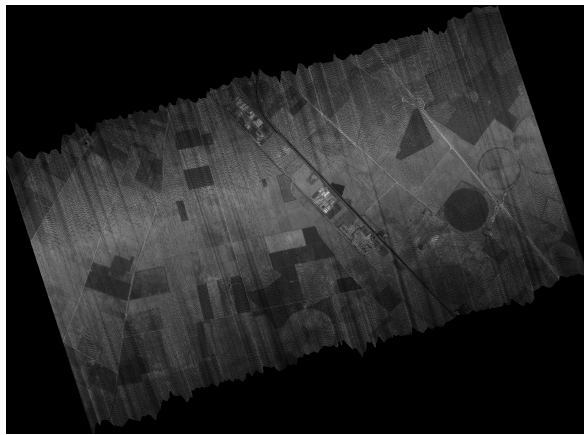
Figura 6.26: Visualización en formato DMF de los resultados obtenidos con el sistema SpaceCube para la escena de GMéxico. (a) Máximo espectral. (b) Mínimo espectral. (c) Media espectral. (d) Desviación típica espectral. (e) Índice IR Ratio. (f) Índice NDVI.



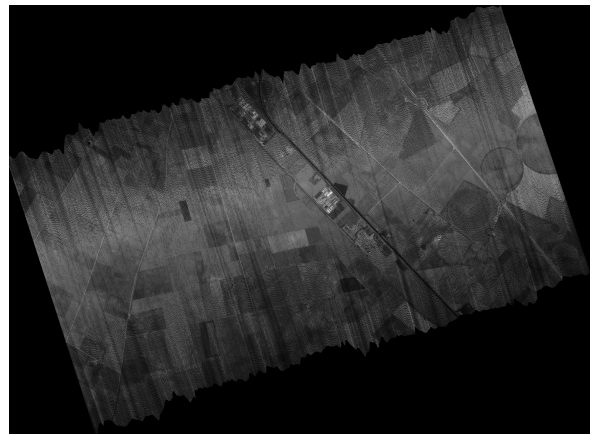
(a)



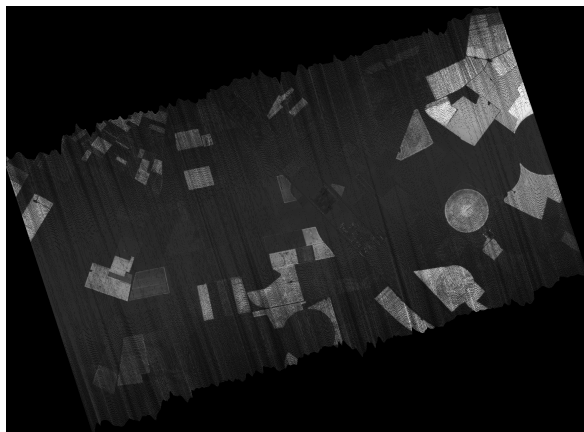
(b)



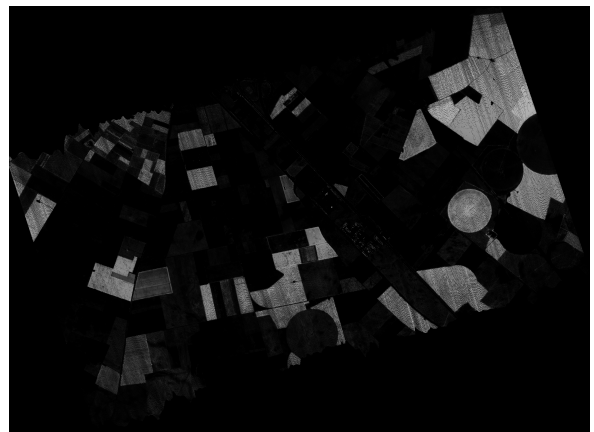
(c)



(d)



(e)



(f)

Figura 6.27: Visualización en formato DMF de los resultados obtenidos con el sistema SpaceCube para la escena de Cantera.

(a) Máximo espectral. (b) Mínimo espectral. (c) Media espectral. (d) Desviación típica espectral. (e) Índice IR Ratio. (f) Índice NDVI.

Cuprite			
	<i>FPGA 1</i>	<i>FPGA 2</i>	<i>SpaceCube</i>
<i>RS232-9600-19200</i>	0.952 s	0.723 s	1.2 s
<i>RS232-19200-128000</i>	0.476 s	0.275 s	0.513 s
<i>RS232-128000-230400</i>	0.072 s	0.056 s	0.092 s
<i>RS232-230400-Eth-10</i>	0.04 s	0.02 s	0.04 s
<i>Eth-10-100</i>	0.00095 s	0.00051 s	0.001 s
<i>RS422-10-100</i>	0.0011 s	0.00059 s	0.0012 s
Barrax			
	<i>FPGA 1</i>	<i>FPGA 2</i>	<i>SpaceCube</i>
<i>RS232-9600-19200</i>	0.277 s	0.217 s	0.356 s
<i>RS232-19200-128000</i>	0.136 s	0.081 s	0.15 s
<i>RS232-128000-230400</i>	0.021 s	0.017 s	0.027 s
<i>RS232-230400-Eth-10</i>	0.0116 s	0.0059 s	0.0118 s
<i>Eth-10-100</i>	0.0003 s	0.00017 s	0.00034 s
<i>RS422-10-100</i>	0.00032 s	0.00017 s	0.00036 s
Mediterráneo			
	<i>FPGA 1</i>	<i>FPGA 2</i>	<i>SpaceCube</i>
<i>RS232-9600-19200</i>	0.11 s	0.092 s	0.148 s
<i>RS232-19200-128000</i>	0.055 s	0.033 s	0.061 s
<i>RS232-128000-230400</i>	0.0083 s	0.0072 s	0.0114 s
<i>RS232-230400-Eth-10</i>	0.0046 s	0.0023 s	0.0047 s
<i>Eth-10-100</i>	0.00015 s	0.000081 s	0.00016 s
<i>RS422-10-100</i>	0.00013 s	0.00007 s	0.00014 s
GMéxico			
	<i>FPGA 1</i>	<i>FPGA 2</i>	<i>SpaceCube</i>
<i>RS232-9600-19200</i>	0.952 s	0.723 s	1.2 s
<i>RS232-19200-128000</i>	0.476 s	0.275 s	0.513 s
<i>RS232-128000-230400</i>	0.072 s	0.056 s	0.092 s
<i>RS232-230400-Eth-10</i>	0.04 s	0.02 s	0.04 s
<i>Eth-10-100</i>	0.00095 s	0.00051 s	0.00108 s
<i>RS422-10-100</i>	0.00109 s	0.00059 s	0.00122 s
Cantera			
	<i>FPGA 1</i>	<i>FPGA 2</i>	<i>SpaceCube</i>
<i>RS232-9600-19200</i>	0.281 s	0.22 s	0.362 s
<i>RS232-19200-128000</i>	0.141 s	0.082 s	0.153 s
<i>RS232-128000-230400</i>	0.021 s	0.017 s	0.028 s
<i>RS232-230400-Eth-10</i>	0.0118 s	0.006 s	0.012 s
<i>Eth-10-100</i>	0.00031 s	0.00017 s	0.00035 s
<i>RS422-10-100</i>	0.00032 s	0.00017 s	0.00036 s

Tabla 6.7: Tiempo de procesamiento de una única medición en el sistema SpaceCube simulado.

6.4.4. Análisis de los resultados del experimento.

En esta sección, se ha diseñado e implementado una metodología hardware teniendo como referencia el dispositivo SpaceCube, y en el cual los datos procesados se ofrecen en formato DMF. Está compuesto por dos tarjetas FPGA Virtex-5 interconectadas, en la que cada una de ellas cumplen una determinada función. La primera FPGA realiza un suavizado del espectro de acuerdo al algoritmo de Savitzky-Golay (S-C), mientras que la segunda realiza cálculos estadísticos de cada espectro procesado por la primera tarjeta FPGA, como son el máximo, mínimo, media, desviación típica, índice IR Ratio e índice NDVI. Tras realizar una ejecución con las imágenes del *dataset real*, se obtuvieron los siguientes resultados:

1. Capacidades de transmisión y procesamiento de cada una de las tarjetas FPGA, de acuerdo a los tipos de conexión (Figura 6.21).

	Cuprite		
	<i>FPGA 1</i>	<i>FPGA 2</i>	<i>SpaceCube</i>
<i>RS232-9600-19200</i>	4.063 días	4.046 días	4.063 días
<i>RS232-19200-128000</i>	2.032 días	2.023 días	2.032 días
<i>RS232-128000-230400</i>	7.314 horas	7.282 horas	7.314 horas
<i>RS232-230400-Eth-10</i>	4.063 horas	4.046 horas	4.063 horas
<i>Eth-10-100</i>	4.749 min	4.729 min	4.749 min
<i>RS422-10-100</i>	5.617 min	5.593 min	5.617 min
	Barrax		
	<i>FPGA 1</i>	<i>FPGA 2</i>	<i>SpaceCube</i>
<i>RS232-9600-19200</i>	5.365 horas	5.285 horas	5.365 horas
<i>RS232-19200-128000</i>	2.683 horas	2.643 horas	2.683 horas
<i>RS232-128000-230400</i>	24.145 min	23.784 min	24.145 min
<i>RS232-230400-Eth-10</i>	13.414 min	13.213 min	13.414 min
<i>Eth-10-100</i>	17.713 s	17.491 s	17.713 s
<i>RS422-10-100</i>	18.543 s	18.266 s	18.543 s
	Mediterráneo		
	<i>FPGA 1</i>	<i>FPGA 2</i>	<i>SpaceCube</i>
<i>RS232-9600-19200</i>	1.789 días	1.723 días	1.789 días
<i>RS232-19200-128000</i>	21.474 horas	20.678 horas	21.474 horas
<i>RS232-128000-230400</i>	3.221 horas	3.102 horas	3.221 horas
<i>RS232-230400-Eth-10</i>	1.789 horas	1.723 horas	1.789 horas
<i>Eth-10-100</i>	2.932 min	2.859 min	2.932 min
<i>RS422-10-100</i>	2.474 min	2.382 min	2.474 min
	GMéxico		
	<i>FPGA 1</i>	<i>FPGA 2</i>	<i>SpaceCube</i>
<i>RS232-9600-19200</i>	9.290 días	9.249 días	9.290 días
<i>RS232-19200-128000</i>	4.645 días	4.625 días	4.645 días
<i>RS232-128000-230400</i>	16.721 horas	16.648 horas	16.721 horas
<i>RS232-230400-Eth-10</i>	9.290 horas	9.249 horas	9.290 horas
<i>Eth-10-100</i>	10.857 min	10.812 min	10.857 min
<i>RS422-10-100</i>	12.842 min	12.786 min	12.842 min
	Cantera		
	<i>FPGA 1</i>	<i>FPGA 2</i>	<i>SpaceCube</i>
<i>RS232-9600-19200</i>	2.459 días	2.423 días	2.459 días
<i>RS232-19200-128000</i>	1.230 días	1.212 días	1.230 días
<i>RS232-128000-230400</i>	4.427 horas	4.362 horas	4.427 horas
<i>RS232-230400-Eth-10</i>	2.459 horas	2.423 horas	2.459 horas
<i>Eth-10-100</i>	3.240 min	3.200 min	3.240 min
<i>RS422-10-100</i>	3.400 min	3.350 min	3.400 min

Tabla 6.8: *Tiempos de procesamiento de las imágenes completas en el sistema SpaceCube simulado.*

- Porcentaje de tiempos de ocio de cada uno de los dispositivos de transmisión de datos para cada una de las tarjetas FPGA, de acuerdo a los tipos de conexión (Figura 6.22).
- Porcentaje de tiempos de ocio de cada uno de los dispositivos de transmisión de datos para cada una de las tarjetas FPGA, de acuerdo a la combinación de conexiones seleccionada (Tabla 6.6).
- Tiempos de procesamiento de una única medición para cada una de las imágenes (Tabla 6.7).
- Tiempos totales de procesamiento de cada una de las imágenes (Tabla 6.8).
- Porcentaje de dedicación a procesamiento y comunicaciones para cada una de las imágenes (Tabla 6.9).
- Análisis visual de los resultados generados por el sistema hardware propuesto, para cada una de las imágenes (Figuras 6.23, 6.24, 6.25, 6.26 y 6.27).

Cuprite				
	<i>T. Conexiones</i>	<i>T. Bus</i>	<i>T. Comunicaciones (Total)</i>	<i>T. Procesamiento</i>
<i>RS232-9600-19200</i>	99.971 %	0.007 %	99.978 %	0.022 %
<i>RS232-19200-128000</i>	99.933 %	0.016 %	99.949 %	0.051 %
<i>RS232-128000-230400</i>	99.626 %	0.089 %	99.715 %	0.285 %
<i>RS232-230400-Eth-10</i>	99.145 %	0.203 %	99.347 %	0.653 %
<i>Eth-10-100</i>	70.165 %	7.069 %	77.234 %	22.766 %
<i>RS422-10-100</i>	73.406 %	6.301 %	79.707 %	20.293 %
Barrax				
	<i>T. Conexiones</i>	<i>T. Bus</i>	<i>T. Comunicaciones (Total)</i>	<i>T. Procesamiento</i>
<i>RS232-9600-19200</i>	99.972 %	0.007 %	99.979 %	0.021 %
<i>RS232-19200-128000</i>	99.934 %	0.016 %	99.950 %	0.050 %
<i>RS232-128000-230400</i>	99.638 %	0.086 %	99.724 %	0.276 %
<i>RS232-230400-Eth-10</i>	99.156 %	0.200 %	99.356 %	0.644 %
<i>Eth-10-100</i>	72.984 %	6.406 %	79.390 %	20.610 %
<i>RS422-10-100</i>	73.810 %	6.210 %	80.020 %	19.980 %
Mediterráneo				
	<i>T. Conexiones</i>	<i>T. Bus</i>	<i>T. Comunicaciones (Total)</i>	<i>T. Procesamiento</i>
<i>RS232-9600-19200</i>	99.974 %	0.006 %	99.980 %	0.020 %
<i>RS232-19200-128000</i>	99.937 %	0.015 %	99.952 %	0.048 %
<i>RS232-128000-230400</i>	99.662 %	0.080 %	99.743 %	0.257 %
<i>RS232-230400-Eth-10</i>	99.181 %	0.195 %	99.376 %	0.624 %
<i>Eth-10-100</i>	77.649 %	5.310 %	82.959 %	17.041 %
<i>RS422-10-100</i>	74.659 %	6.020 %	80.679 %	19.321 %
GMéxico				
	<i>T. Conexiones</i>	<i>T. Bus</i>	<i>T. Comunicaciones (Total)</i>	<i>T. Procesamiento</i>
<i>RS232-9600-19200</i>	99.971 %	0.007 %	99.978 %	0.022 %
<i>RS232-19200-128000</i>	99.933 %	0.016 %	99.949 %	0.051 %
<i>RS232-128000-230400</i>	99.626 %	0.089 %	99.715 %	0.285 %
<i>RS232-230400-Eth-10</i>	99.145 %	0.203 %	99.347 %	0.653 %
<i>Eth-10-100</i>	70.165 %	7.069 %	77.234 %	22.766 %
<i>RS422-10-100</i>	73.406 %	6.301 %	79.707 %	20.293 %
Cantera				
	<i>T. Conexiones</i>	<i>T. Bus</i>	<i>T. Comunicaciones (Total)</i>	<i>T. Procesamiento</i>
<i>RS232-9600-19200</i>	99.972 %	0.007 %	99.979 %	0.021 %
<i>RS232-19200-128000</i>	99.934 %	0.016 %	99.950 %	0.050 %
<i>RS232-128000-230400</i>	99.638 %	0.086 %	99.724 %	0.276 %
<i>RS232-230400-Eth-10</i>	99.156 %	0.200 %	99.356 %	0.644 %
<i>Eth-10-100</i>	72.930 %	6.419 %	79.349 %	20.651 %
<i>RS422-10-100</i>	73.801 %	6.212 %	80.013 %	19.987 %

Tabla 6.9: Porcentajes de dedicación a comunicaciones y procesamiento en el sistema SpaceCube simulado.

La simulación del sistema SpaceCube diseñado demuestra la importancia del desarrollo de los sistemas reconfigurables y del procesamiento hardware a bordo de los satélites. A las ventajas que estos tipos de sistemas presentan por su posibilidad de ser reprogramados desde Tierra, hay que sumarle aquéllas derivadas de aplicarles formatos de datos específicos como el formato DMF. Este formato no sólo permite una precisión espacial de los datos adquiridos por los instrumentos reducidos a las limitaciones del sistema de posicionamiento (tal y como se ha demostrado en los experimentos anteriores), sino que además permite un almacenamiento y transferencia de la información de una manera organizada y eficaz. Además, tal y como se ha demostrado, mediante el uso de la matriz DMF existe la posibilidad de recibir un producto recién adquirido por los instrumentos totalmente georreferenciado y disponible para el análisis, junto con información adicional y diferentes niveles de procesamiento procedentes de las metodologías implementadas, eliminando los tiempos de espera característicos de las misiones y potenciando el análisis en tiempo real a bordo de la nave.

El desarrollo de un sistema satélite modular como SpaceCube, basado en módulos de dos tarjetas FPGA, permite la ejecución de algoritmos de manera eficiente, en donde el componente hardware representa un gran papel. Esto permite el diseño de metodologías basadas en el paralelismo, sacando el máximo partido al rendimiento del sistema. Sin embargo, para el diseño de un sistema de este tipo es necesario considerar varios puntos críticos derivados de los tiempos de procesamiento de la metodología implementada y de los tiempos de transferencia de los datos, tanto de entrada y salida del sistema, como en el interior del mismo (entre los diferentes módulos que lo conforman). En este caso particular, en el cual el sistema SpaceCube simulado está compuesto por dos tarjetas FPGA interconectadas mediante diferentes tipos de conexiones, es posible remarcar cuatro puntos críticos, los cuales es necesario considerar e implementar de manera adecuada para un correcto funcionamiento del sistema:

- *Frecuencia de procesamiento*: Capacidad de procesamiento de la metodología hardware implementada. Hay que tener en cuenta que el algoritmo implementado debe caber completamente en la FPGA, y que la frecuencia de reloj soportada no debe ser baja, o ralentizará el resto del sistema. Es decir, la metodología implementada debe ser tan pequeña y rápida como posible. Además, es necesario considerar su dependencia con el tipo de datos de entrada (como en el caso de las metodologías implementadas, con el espectro de la medición), ya que su rapidez y rendimiento dependerá directamente del tamaño del dato. Por supuesto, la metodología hardware puede ser implementada de múltiples maneras, y aunque aquí se haya presentado un esquema en el cual todas las tareas residían en un periférico conectado mediante un bus central al procesador, y donde la única tarea de éste era recibir y enviar datos, los diferentes componentes pueden ser utilizados y combinados de diferentes formas según las necesidades de la misión, y también de acuerdo a las versiones de tarjetas FPGA involucradas. De esta manera, es posible repartir las diferentes funciones de la metodología en uno o varios periféricos, e incluso utilizar el procesador (o procesadores) como dispositivo auxiliar. O bien considerar un punto de vista totalmente contrario, implementar principalmente la metodología en el procesador (o procesadores) y el resto en uno o varios periféricos auxiliares. También cabe la posibilidad de prescindir completamente del procesador e implementar todo (incluidos las interfaces de conexión) a un nivel de hardware.
- *Frecuencia de transmisión del bus central*: Capacidad del bus central de enviar los datos del procesador al periférico, y viceversa. Hay que tener en cuenta que, en este estudio, un único bus central ha sido utilizado para las transmisiones, y debe ser compartido tanto para los datos pre-procesados como para los post-procesados. Por tanto, hay que considerar los tiempos de transferencia de este componente siempre dos veces. Por supuesto, su existencia depende de la arquitectura hardware seleccionada y de las versiones de tarjetas FPGA utilizadas. Es posible prescindir del bus central y realizar directamente las comunicaciones mediante interfaces hardware dedicadas (como periféricos), o bien incrementar su número (mediante sub-buses) o dedicando un bus por procesador, en caso de que en la versión de la tarjeta FPGA presente más de un procesador.

- *Frecuencia de entrada de datos*: Capacidad del sistema de recibir datos del exterior. Depende directamente del tipo de conexión utilizado. La selección de dicho tipo de conexión debe realizarse con cuidado, ya que debe ser la menor frecuencia de transferencia del sistema, y los tiempos de ejecución serán totalmente dependientes de la misma. Esto es debido a que el sistema no debe tardar en procesar y enviar un determinado tipo de dato (en el caso del sistema propuesto, una medición) más de lo que se tarda en recibir el siguiente, ya que implicaría la saturación de los buffers de entrada, impidiendo el correcto funcionamiento del mismo.
- *Frecuencia de salida de datos*: Capacidad del sistema de enviar datos al exterior. Depende directamente del tipo de conexión utilizado. La frecuencia con la que los datos son enviados al exterior, una vez procesados, nunca debe ser inferior a la frecuencia de datos de entrada, pues esto provocaría la saturación de los buffers de entrada, impidiendo el correcto funcionamiento del mismo. Existen ocasiones en la que el procesamiento de datos implica la adición de nueva información, por lo que la cantidad de datos a enviar puede ser mayor a la recibida (en el caso del sistema propuesto, ocurre en la segunda tarjeta FPGA, en la que se reciben espectros junto con sus coordenadas, y han de enviarse esta misma información, pero junto a cálculos estadísticos como el máximo, el mínimo, media y desviación estándar de los espectros, y los índices IR Ratio y NDVI). Si ocurre esta circunstancia, no es posible utilizar la misma frecuencia de datos de entrada que de salida (ya que se saturarían, del mismo modo, los buffers de entrada), y sería necesario, o bien decrementar la frecuencia de entrada de datos de todo el sistema (manteniendo la frecuencia de salida), o bien aumentar la frecuencia de salida de datos.

De esta manera, la selección de los tipos de conexiones de entrada y salida de datos es determinante para el correcto funcionamiento del sistema. La frecuencia de llegada de los datos de entrada no debe ser mayor al tiempo de procesamiento de las mediciones, ni a la capacidad de envío de los resultados obtenidos al exterior. En el sistema SpaceCube simulado, formado por dos tarjetas FPGA Xilinx Virtex-5, el número de conexiones disponibles es amplio (incluyendo baja y alta velocidad), lo que proporciona la posibilidad de realizar diferentes combinaciones de acuerdo al número de conectores de cada tipo de conexión (en la versión ML510 existen dos tipos de conectores para cada tipo de conexión, mientras que en la versión ML507 sólo uno de cada tipo, excepto para el protocolo RS-232). Teniendo en cuenta la metodología propuesta, las combinaciones posibles fueron seleccionadas proporcionando diferentes velocidades, siempre cuidando que el tiempo de ocio de la conexión de salida sea lo menor posible.

Tras analizar los tiempos de ejecución para cada una de las imágenes (Tablas 6.7, 6.8 y 6.9), se encontró que el principal problema es el tiempo dedicado a la transmisión de los datos, el cual puede variar según las combinaciones de conexiones utilizadas:

- *Combinaciones de conexiones de baja velocidad (RS-232)*: El procesamiento de una imagen completa puede conllevar desde varios días a varias horas (utilizando las velocidades máximas posibles), siendo el porcentaje del tiempo de transferencia de datos del 99.5 al 99.9 % del tiempo de computación global. Esto implica que, si se usa este tipo de conexión, el procesamiento a bordo del satélite es bastante limitado, y únicamente estaría destinado a cálculos de regiones específicas (previamente seleccionadas) que no requieran un tiempo de respuesta inmediato. Este tipo de cálculos podrían ser ejecutados cada cierto tiempo (varias veces al día, o cada varios días), o cuando el satélite se encuentre más libre de trabajo. Varias posibilidades podría ser la calibración de los instrumentos, o correcciones de órbita, a partir de imágenes referencia.
- *Combinaciones de conexiones de alta velocidad (Ethernet y RS-422)*: El procesamiento de una imagen completa puede conllevar varios minutos, siendo el porcentaje del tiempo de transferencia de datos del 80 al 90 %. Estas combinaciones de conexiones son las más apropiadas si se desean desarrollar algoritmos que requieran un tiempo de respuesta rápido, cercano al tiempo real. Ejemplos de metodologías, aparte de la propuesta en los experimentos, podrían ser la

monitorización y alerta de incendios forestales, detección y eliminación de nubes, o la elaboración de nuevos productos adicionales, como resultados de clasificaciones espectrales.

Sin embargo, aunque se utilicen conexiones de alta velocidad para la transferencia de los datos, el porcentaje dedicado a esta actividad es aún bastante elevado. Gracias a los conectores XGI (en la versión ML507) y PM (en la versión ML510), que ofrecen la posibilidad de transmitir información mediante tarjetas puente personales, es posible diseñar y proponer conexiones dedicadas de alta velocidad acorde a las necesidades de las misiones de campaña o espaciales, ampliando el repertorio de conectores convencionales ofrecidos por Xilinx en estos dos tipos de tarjetas.

Por otro lado, el porcentaje dedicado a la transferencia de datos por parte del bus central, el cual se demostró que implica un incremento de los tiempos de ejecución totales en la Sección 6.2, presenta unos porcentajes insignificantes comparados con los correspondientes a las entradas y salidas de las tarjetas. Sería posible realizar las mismas mejoras que las presentadas en dicha sección, como la utilización memorias compartidas y conexiones dedicadas, pero para unos porcentajes tan bajos los tiempos de procesamiento de las imágenes serían prácticamente los mismos. Estos porcentajes, como son mayores para conexiones de baja velocidad, sí que se decrementarían si las mejoras propuestas en dicha sección fueran aplicadas. Incluso podrían desaparecer completamente si tanto el procesador como el bus central del sistema son eliminados. De esta manera, la gestión de las transferencias podrían hacerse de manera directa mediante módulos hardware que hicieran de intermediarios entre el periférico de comunicaciones y el periférico que implementa la metodología.

Por otro lado, y respecto a la metodología hardware implementada en las tarjetas FPGA, los tiempos de procesamiento de las mediciones adquiridas (sin contar los tiempos de transferencias de los datos) muestran una ejecución óptima y rápida, con capacidad de poder ser ejecutada hasta a frecuencias de 230 MHz. Esto permite su utilización en nuevas versiones o actualizaciones de la tarjeta que permitan el diseño de sistemas hardware más rápidos. Otra limitación importante es la capacidad de la FPGA, la cual impidió la paralelización de la mayoría de los módulos presentes debido al gran consumo de recursos, por lo que fue necesario la secuencialización implicando un incremento considerable en los tiempos de procesamiento.

Existen diferentes mejoras u optimizaciones que pueden ser consideradas en un futuro:

- Si no se desea tanta precisión en los resultados, sería posible utilizar el binario natural como tipo de datos, en vez de *Single*. De esta manera, se podría prescindir de los módulos de cambio de formato ToSingle256 y ToInt256, y el resto de módulos, al no trabajar en punto flotante, podrían trabajar a mayores frecuencias y decrementar el tiempo de cómputo.
- El resultado generado por la primera tarjeta FPGA se convierte a binario natural y es proporcionado en este tipo de datos a la segunda, la cual tiene que realizar de nuevo la conversión a *Single*. Aunque se perdería la abstracción de la metodología implementada en la tarjeta FPGA como unidad de procesamiento independiente, sería posible que la primera tarjeta FPGA enviara ya el resultado convertido a tipo *Single*, y que la segunda la recibiera y procesara directamente en este formato. De esta manera el módulo ToInt256 de la primera tarjeta FPGA, y el módulo ToDouble256 de la segunda desaparecerían. Sin embargo, hay que tener en cuenta que todos los datos serán ahora de 32 bits (incluyendo los espectros) por lo que el tiempo de transferencia de una tarjeta FPGA a la otra se incrementará considerablemente. De acuerdo al tipo de conexión utilizado, y a la velocidad máxima proporcionada por la misma, esta opción puede suponer una optimización o no.
- Puede existir parte de la comunidad científica que muestre cierta desconfianza al hecho de trabajar con un espectro modificado (suavizado en la primera tarjeta FPGA) respecto al adquirido realmente por el sensor, y que desee también disponer de los datos originales. De esta manera, aparte del suavizado y el estudio estadístico realizado por el sistema, también habría

que enviar al exterior el espectro original, implicando un incremento de la cantidad de datos a transferir en la conexión de salida. Existen varias posibilidades para afrontar esta circunstancia:

- No utilizar la plataforma SpaceCube para esta tarea. Es decir, el sistema funcionaría del mismo modo, pero la transmisión del espectro original podría delegar en otro componente del satélite, como el Sistema de Computación o algún otro dispositivo adicional.
- Utilizar una única conexión de salida para transferir los datos: Tal y como se ha estado realizando en los experimentos. Esto implicaría que la carga de transmisión del puerto de salida sería bastante superior, por lo que habría que escoger una frecuencia de conexión de entrada que menor, que permita la completa transmisión del resultado al exterior y que no sature los buffers de entrada.
- Utilizar otro conector adicional para transmitir al exterior el espectro de salida (preferiblemente en la primera tarjeta FPGA, para evitar ser transmitido a la segunda). Como en este caso la cantidad de datos de salida es inferior o igual (si se transmiten también las coordenadas), la frecuencia utilizada puede ser igual a la de entrada. No obstante, hay que tener en cuenta que existirían dos puertos de salida diferentes, y, a no ser que se utilice un mecanismo de sincronización entre ambos mediante módulos hardware dedicados, el concepto del formato DMF podría perderse. Es decir, en este caso el formato DMF de salida del sistema SpaceCube podría estar constituido por coordenadas (latitud y longitud), espectro original, espectro suavizado, y cálculos estadísticos (máximo, mínimo, media, desviación estándar, e índices IR Ratio y NDVI). Sin embargo, si cada uno de los campos es transferido por un puerto diferente, se hace necesario un dispositivo (que podría ser externo a SpaceCube) que permita la correcta generación del DMR. Otra solución para no perder el concepto del formato DMF consistiría en considerar dos tipos de productos diferentes, representados por dos tipos de DMR también diferentes: Uno representado por las coordenadas y el espectro original, y otro representado por las coordenadas, el espectro suavizado y los cálculos estadísticos.

Finalmente, si se analizan en detalle los resultados visuales obtenidos (Figuras 6.23, 6.24, 6.25, 6.26 y 6.27), llama especialmente la atención dos circunstancias. La primera de ellas es la mancha situada en el centro de la imagen correspondiente a la escena de Mediterráneo, la cual queda patente al realizar el cálculo de la media (Figura 6.25(c)), la desviación típica (Figura 6.25(d)), el índice IR Ratio (Figura 6.25(e)), y especialmente en el cálculo del mínimo (Figura 6.25(b)), caso en el que aparece potenciada indicando valores espectrales muy elevados. Dicha mancha corresponde a la reflexión directa de los rayos solares sobre el Mar Mediterráneo (*sunglint*). La segunda circunstancia son los resultados correspondientes a la detección de vegetación mediante las técnicas de IR Ratio y NDVI en el caso de la escena de Cuprite (Figuras 6.23(e) y 6.23(f)). Para esta escena, el resultado aparece casi completamente en negro, indicando un escaso índice de vegetación. Tras un estudio del área que abarca la imagen, fue posible comprobar que se trata de un área correspondiente al condado de Esmeralda, Nevada (USA), una zona desértica localizada en el suroeste del país [Cla96, Abr77, Ken99]. Es por este motivo por el cual los índices de vegetación presentados son tan bajos. De hecho, si se realiza un zoom cerca de la esquina inferior derecha (Figura 6.28), se puede comprobar que sí hay una pequeña área con alta vegetación. A partir de las coordenadas ofrecidas por las mediciones, se identificó dicha vegetación como los árboles de la zona de aterrizaje Lida Junction Airstrip.

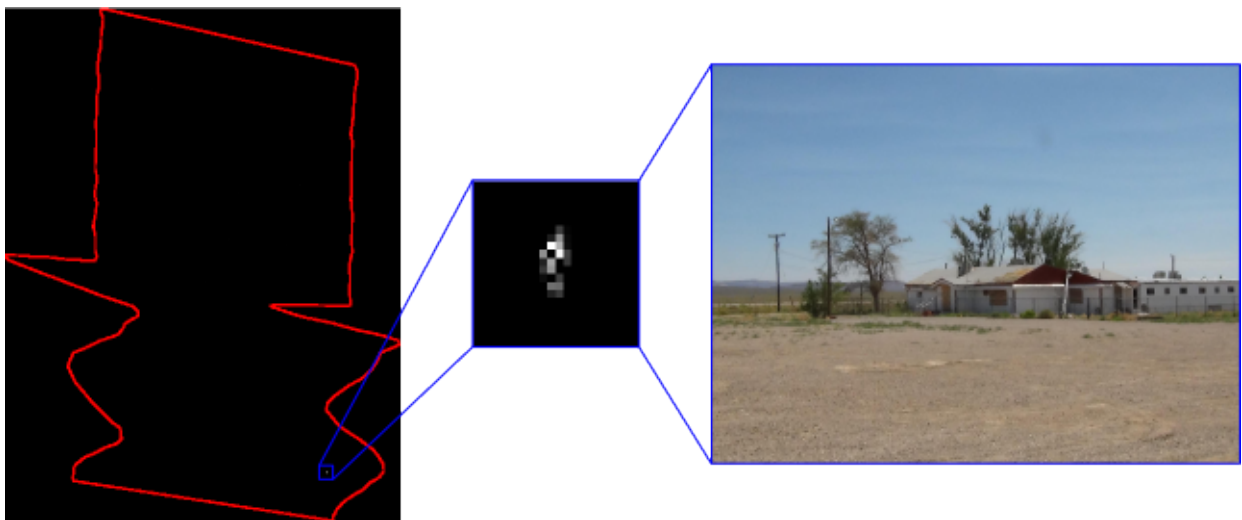


Figura 6.28: Localización de la zona de aterrizaje Lida Junction Airstrip, único foco de alta vegetación en el desierto perteneciente a la escena de Cuprite.

El siguiente capítulo analiza los resultados obtenidos en este capítulo y en el anterior, y elabora una serie de conclusiones que pueden ser extraídas tras el estudio completo realizado.

Capítulo 7

Conclusiones / Conclusions

En la presente memoria se ha diseñado un nuevo formato, Diffused Matrix Format (DMF), para el almacenamiento, visualización y procesamiento de los datos obtenidos por plataformas de sensores en el ámbito de la teledetección. Este formato busca la sencillez y rapidez que ofrecen las estructuras matriciales, con el objetivo de minimizar los problemas principales derivados de las imprecisiones generadas durante el proceso de geocorrección de la información, así como la fácil integración a metodologías más complejas y a las necesidades y requisitos de las misiones. Se basa en la consideración de la medición como unidad básica de representación, en donde la información instrumental (proporcionada por los sensores) y la información espacial (proporcionada por el sistema de posicionamiento) conforman el Diffused Matrix Record (DMR). Los registros DMR son posicionados sobre una rejilla matricial, denominada *matriz DMF* de acuerdo a su localización, de tal forma que en cada celda DMF pueden existir ninguno, uno, o varios registros DMR. El almacenamiento de los registros DMR dentro de una misma celda DMF puede ser diferente, variando desde listas simples, a listas ordenadas o árboles Kd-tree. Con esta política, se limita el error espacial al introducido por el sistema de posicionamiento, se evita el duplicado de la información al no tener que desplazar mediciones vecinas para rellenar píxeles no asignados, se garantiza el uso y la integridad del 100 % de los datos recogidos por los sensores, y se minimiza el consumo de memoria al considerar las celdas DMF situadas fuera de la escena (y de subáreas en las que no se ha adquirido medición por la geometría del sensor o los movimientos de la plataforma) como *Null*. Además, de manera adicional, al estar el registro DMR compuesto tanto por la información instrumental como por la espacial, es posible salvar la región de interés como una ristra secuencial de dichos registros DMR, minimizando el consumo de espacio en disco y a la vez facilitando su distribución.

Para comprobar el rendimiento del formato DMF, se realizaron una serie de experimentos tanto con imágenes sintéticas, como con imágenes reales adquiridas por sensores a bordo de aviones y satélites. Los experimentos siguieron dos líneas bien diferenciadas: la primera de ellas pretendían realizar un estudio del comportamiento del formato DMF como estructura de datos aplicado a problemas convencionales; mientras que la segunda se centraba en la adaptación de dicho formato y de los sistemas de computación para realizar procesamiento a bordo de la plataforma en tiempo real, dedicando especial atención al dispositivo hardware reconfigurable SpaceCube, el cual hace uso de chips FPGA.

Los resultados muestran que el formato DMF supone una gran alternativa a las estructuras y formatos de datos convencionales, y que su diseño supone una serie de ventajas a la hora de visualizar, almacenar y procesar las mediciones recogidas por los instrumentos utilizados en campañas de campo, aviones o satélites.

Las numerosas ventajas que el formato DMF proporciona se detallan a continuación:

- Presenta un diseño no muy complejo de implementar, potenciando su integración a cualquier

tipo de sistema de adquisición de datos, plataforma, necesidad o requerimiento.

- Versátil y funcional en cualquier tipo de plataforma, ya que los datos mantienen su integridad dentro de la estructura independientemente de los movimientos, turbulencias o dirección de vuelo presentados, así como de la configuración de los sensores a bordo.
- Al almacenar las coordenadas originales para cada medición, el error espacial se minimiza al máximo, quedando limitada a las imprecisiones introducidas por el sistema de posicionamiento. Aunque cada registro DMR esté situada en una determinada celda DMF, la cual abarca un pequeño subárea de la escena de interés, su localización exacta siempre está registrada.
- El formato DMF considera únicamente la información real adquirida por los sensores, sin la adición de datos adicionales o promediados, como los introducidos en el proceso de geocorrección. Con esto se consigue que los resultados obtenidos por las metodologías sean mucho más fiables y precisos en comparación a los obtenidos con las imágenes análogas geocorregidas por técnicas convencionales.
- Aprovecha la principal cualidad de las estructuras matriciales, el acceso directo a una determinada posición de la escena, potenciando los tiempos de procesamiento y cómputo.
- El tamaño de celda DMF es adaptable, manteniendo en todo momento la integridad la información. Es posible utilizar, de esta manera, diferentes tamaños de matriz DMF de acuerdo a las necesidades de la misión. En caso de desconocimiento de los parámetros del vuelo, es posible determinar el tamaño de celda DMF de manera automática mediante un método adicional, el cual se basa en el estudio de las distancias en latitud y longitud entre las mediciones.
- A la hora de visualizar los datos, el registro DMR más cercano al centro de su celda DMF correspondiente es seleccionado para la representación a píxel. Los valores instrumentales se escalan a un gradiente de azul para el mínimo, y blanco para el máximo, de manera equivalente al realizado por las imágenes convencionales. Regiones situadas fuera del área de interés, o no medidas por los instrumentos, permanecen como píxeles sin información de color negro. De esta manera, se consigue una visualización de la matriz DMF y una representación de la escena muy fiables desde un punto de vista computacional, al no existir imprecisiones espaciales como en las imágenes geocorregidas por métodos convencionales.
- El hecho de almacenar únicamente la información real adquirida por los sensores implica una reducción considerable del consumo de tamaño en disco. Además presenta tres políticas de almacenamiento diferentes con sus correspondientes características, lo que pueden proporcionar diferentes ventajas dependiendo de los requisitos de las misiones. Las dos primeras utilizan cabeceras adicionales para mapear la localización de los registros DMR dentro de la matriz DMF, suponiendo un ligero incremento en el consumo de tamaño en disco pero optimizando los tiempos de carga de datos. La tercera política, en cambio, almacena la información almacenada en la matriz DMF como una secuencia de registros DMR, por lo que el tamaño en disco se minimiza al máximo, convirtiéndose en la mejor alternativa para la distribución de los datos a la comunidad científica, pero suponiendo un ligero aumento en los tiempos de carga.
- Aunque se trate de un formato matricial a alto nivel, existe la posibilidad de habilitar estructuras más complejas para un óptimo acceso y procesamiento de las mediciones situadas en una determinada celda DMF. Estas estructuras abarcan desde árboles Kd-tree (que dispone los datos de manera jerárquica), listas dinámicas (que dispone los datos de manera ordenada según un determinado criterio, por ejemplo, la distancia de la medición al centro de la celda DMF que lo contiene), o acceso directo secuencial a los registros DMR almacenados en la matriz DMF (evitando el análisis de celdas DMF vacías). Estas metodologías potencian aún más el tratamiento y procesamiento de la información, optimizando los tiempos de ejecución.
- Al basarse en la medición como unidad básica de almacenamiento, ampliando los campos de información correspondientes al mismo se puede adaptar el formato DMF a cualquier tipo de

situación o necesidad de manera fácil y sencilla. De esta forma, metodologías más avanzadas como la fusión de datos o la detección de cambios reducen su complejidad considerablemente, al existir la posibilidad de integrar las mediciones correspondientes a diferentes sensores o vuelos (o ambos) en una misma estructura de datos. Esto potencia los niveles de computación, ya que mediciones próximas con orígenes distintos estarán situadas en la misma celda DMF, o en su vecindad, reduciendo los tiempos dedicados a la identificación de información análoga.

- Cuando el formato DMF se integra en una plataforma de adquisición en tiempo real, considerándolo como tipo de salida del sistema de computación principal, se potencian las funcionalidades del mismo:
 - Las metodologías se simplifican considerablemente al tratar la medición como unidad básica de procesamiento.
 - Utilizar el formato DMF como tipo de salida proporciona la disponibilidad de los datos de una manera sincronizada y la generación de unos resultados ya georreferenciados desde el momento en el que la adquisición finaliza, potenciando su tratamiento y distribución a la comunidad científica.
 - El tráfico de datos entre el avión / satélite y Tierra, así como el almacenamiento temporal de los datos recogidos por los instrumentos, pueden minimizarse al considerar únicamente mediciones reales adquiridas por los instrumentos. Incluso es posible realizar el descarte de mediciones en tiempo real (de acuerdo a una determinada condición) sin afectar a la estructura ni al formato DMF.

Sin embargo, el formato DMF tiene también implícita una serie de limitaciones y dificultades a tener en cuenta cuando es utilizada como estructura de datos principal, con el objetivo de minimizar los problemas y asegurar un correcto funcionamiento del sistema global.

Dichas limitaciones y consideraciones se detallan a continuación:

- El formato DMF necesita de unas coordenadas para localizar cada medición adquirida por los instrumentos. Esto implica una total dependencia del sistema de posicionamiento y requiere una correcta y perfecta sincronización con todos los dispositivos situados en la plataforma.
- Aunque el formato DMF se basa en estructuras matriciales, es difícil gestionar los registros DMR cuando la densidad de mediciones de las celdas DMF es alta, requiriendo estructuras de datos más avanzadas para un óptimo tratamiento (como árboles Kd-tree o listas dinámicas) incrementando la complejidad general del formato. Además, aunque dichas estructuras siempre suponen una mejora en los tiempos de procesamiento, ésta se hace realmente notable en casos demasiado extremos (matrices DMF muy pequeñas con alta densidad de registros DMR, o matrices DMF muy grandes con baja densidad de registros DMR).
- Gracias al diseño del formato DMF, la adaptación de cualquier tipo de plataforma es rápida y sencilla, pero es necesario dedicar un tiempo mínimo para comprobar su correcta integración. Las fases de implementación y testeo puede suponer un incremento del esfuerzo y coste económico por parte de la empresa o institución.
- El hecho de almacenar cada medición con las coordenadas proporcionadas por el sistema de posicionamiento puede suponer un consumo de tamaño en memoria y disco ligeramente superior a los formatos geocorregidos convencionales, los cuales estiman las localizaciones de todos los píxeles mediante referencias. Sin embargo, al considerar únicamente las mediciones reales adquiridas por los instrumentos, evitando la duplicidad de información y la adición de espectros a 0 para representar zonas situadas fuera de la escena de interés, el tamaño final de la imagen en formato DMF se reduce considerablemente, por lo que el almacenamiento de esta información adicional no supone realmente un problema.

- El tamaño de celda DMF puede ser variable e independiente de la integridad de la información. Esto proporciona varias ventajas, pero la consideración de un tamaño de celda DMF incorrecto puede implicar matrices DMF demasiado pequeñas o demasiado grandes que pueden perjudicar al procesamiento global. Sin embargo, en caso de desconocimiento de este parámetros y de las condiciones del vuelo, es posible determinar el tamaño de celda DMF de manera automática.
- La visualización de la matriz DMF puede llegar a resultar un poco incómoda en determinadas ocasiones, sobre todo si la plataforma ha sido sometida a turbulencias de gran intensidad, ya que subregiones no medidas por el sistema de adquisición aparecen como píxeles negros, los cuales pueden entorpecer el análisis visual. Además, no se representa el 100 % de la información presente en la matriz DMF, pues al existir una correspondencia celda DMF - píxel, únicamente se muestra uno de los registros DMR almacenados por celda DMF (generalmente el más cercano al centro).
- Al no tratarse de un formato convencional, se requiere la completa adaptación de todas las metodologías convencionales del procesamiento de imágenes, desde las más sencillas (erosión, dilatación, etc.) a las más avanzadas (fusión de datos, detección de cambios, etc.).
- Cuando el formato DMF se integra en un sistema de adquisición en tiempo real es recomendable tener en cuenta las siguientes consideraciones:
 - Para establecer el formato DMF como tipo de salida es necesario aportar las mediciones y resultados generados junto con sus coordenadas, por lo que se requiere una perfecta sincronización por parte del sistema de computación (y de las metodologías implementadas) con el sistema de posicionamiento.
 - Los registros DMR proporcionados albergan la localización y valor instrumental para cada medición, además de posibles resultados generados a bordo y en tiempo real, lo que implica un ligero incremento del tamaño de información final a enviar. Se requiere la comprobación de las tasas y frecuencias máximas de recepción y envío de datos proporcionados por los puertos utilizados para un correcto funcionamiento global, así de la manera en la que se realizan las comunicaciones entre los componentes y módulos implementados en el sistema de computación.

En el siguiente capítulo se proponen varias soluciones y mejoras a estas limitaciones o dificultades presentadas por el formato DMF en determinadas situaciones. Se tratan de trabajos y proyectos en los que se trabajará próximamente, por lo que pueden quedar enmarcadas como líneas futuras.

This thesis document presents the design of a new format, the Diffused Matrix Format (DMF), for the storage, display and processing of data obtained by sensor platforms in the field of remote sensing. This format seeks to exploit the simplicity and speed of matrix structures with the aim of minimising problems arising from inaccuracies produced during the information geocorrection process and ensuring easy integration with more complex methodologies and mission needs and requirements. It is based on considering measurements as basic units of representation, in which instrumental information (provided by the sensors) and spatial information (provided by the positioning system) make up the Diffused Matrix Record (DMR). DMRs are positioned on a matrix grid, known as a *DMF matrix*, as per its location, in such a way that, in each DMF cell, zero, one or several DMRs can exist. Storage of the DMRs within the same DMF cell can be different, ranging from simple lists, sorted lists or k-d trees. With this policy, spatial errors are limited to that introduced by the positioning system, duplication of information is avoided by not having to move neighbouring measurements to fill unallocated pixels, usage and the integrity of 100 % of the data collected by the sensors is guaranteed and memory consumption is minimised by considering the DMF cells situated outside the scene (and sub-areas in which measurement has not been acquired by the geometry of the sensor or the movement of the platform) as *Null*. In addition, since the DMR consists of instrumental and spatial information,

it is possible to save the region of interest as a sequential string of DMRs, thereby minimising the consumption of disk space while facilitating distribution.

To check the performance of the DMF format, a series of experiments was performed with synthetic and real images acquired by sensors on board aircraft and satellites. The experiments followed two distinct lines: the first was a study of the behaviour of the DMF format as a data structure applied to conventional problems, while the second focused on the adaptation of this format and the computation systems to perform real-time processing on the platform with particular attention paid to the SpaceCube reconfigurable hardware device, which uses FPGA chips.

The results show that the DMF format is a great alternative to conventional data structures and formats with a design that provides a number of advantages when it comes to display, storage and processing measurements collected by the instruments used in field campaigns and on aircraft and satellites.

The numerous advantages provided by the DMF format are detailed below:

- It features a not too complex design to implement, enhancing its integration with any type of data acquisition system, platform, need or requirement.
 - It is versatile and functional on any kind of platform as the data maintains its integrity within the structure regardless of movement, turbulence or flight direction, as well as the configuration of on-board sensors.
 - By storing the original coordinates for each measurement, spatial errors are minimised to the maximum and limited to inaccuracies introduced by the positioning system. Although each DMR is located in a particular DMF cell, which covers a small sub-area of the area of interest, its exact location is always recorded.
 - The DMF format only considers actual information acquired by the sensors, without the addition of other or averaged data, like that introduced in the geocorrection process. This ensures that the results obtained by the methodologies are much more reliable and accurate compared to those obtained with geocorrected analogue images through conventional techniques.
 - It exploits the main quality of matrix structures: direct access to a certain position in the scene, thereby enhancing processing and computing times.
 - The DMF cell size is adaptable, always maintaining the integrity of the information. It is possible to use, in this way, different DMF matrix sizes depending on the needs of the mission. In the event of not knowing the flight parameters, it is possible to determine the DMF cell size automatically by means of an additional method, which is based on studying latitude and longitude distances between measurements.
 - When displaying the data, the DMR closest to the centre of its corresponding DMF cell is selected for pixel representation. The instrumental values are scaled to a blue gradient for the minimum and white for the maximum, in a way that is equivalent to that carried out for conventional images. Regions located outside the area of interest, or not measured by the instruments, remain as pixels without information in black. By doing so, what is achieved is a very reliable display of the DMF matrix and representation of the scene from a computational point of view, as there is an absence of spatial inaccuracies like those produced in geocorrected images achieved by conventional methods.
 - By storing only the actual data acquired by the sensors, there is a considerable reduction in disk space consumption. It also has three different storage policies with their corresponding features, which can provide different advantages depending on mission requirements. The first two use additional headers to map the location of DMRs within the DMF matrix, which produces a slight increase in disk space consumption but optimises data load time. The third policy, conversely,
-

stores the information in the DMF matrix as a sequence of DMRs, thereby minimising to the maximum disk space consumption and making it the best alternative for the distribution of data to the scientific community, albeit with a slight increase in load times.

- Although it is a high-level matrix format, there exists the possibility of enabling more complex structures for optimum access and processing of measurements situated in a certain DMF cell. These structures range from k-d trees (which arrange data hierarchically) and dynamic lists (which sort data as per certain criteria, for example, the measurement distance at the centre of the DMF cell that contains it) to sequential direct access to the DMRs stored in the DMF matrix (avoiding analysis of empty DMF cells). These methodologies further enhance the treatment and processing of information, thereby optimising execution times.
- Since it is based on measurement as the basic storage unit, expanding the information fields corresponding to it, it is possible to easily and simply adapt the DMF format to any type of situation or need. This way, more advanced methodologies, such as data fusion or change detection, considerably reduce its complexity, as there is the possibility of integrating the measurements corresponding to different sensors or flights (or both) in the same data structure. This enhances computation levels since nearby measurements with different origins are located in the same DMF cell, or in its vicinity, thereby reducing the time taken up identifying similar information.
- When the DMF format is integrated with an acquisition platform in real-time and considered as a type of output of the main computation system, its functionalities are enhanced:
 - Methodologies are considerably simplified when measurements are treated as basic processing units.
 - Using the DMF format as a type of output provides data availability in a synchronised manner and the generation of certain results that have already been georeferenced from the moment that the acquisition is complete, thereby enhancing its processing and distribution to the scientific community.
 - The data traffic between the aircraft/satellite and Earth, as well as the temporary storage of the data collected by the instruments, can be minimised by only considering actual measurements acquired by the instruments. It is even possible to discard measurements in real time (as per a certain condition) without affecting the DMF structure or format.

The DMF format does, however, have a number of implicit limitations and difficulties that need to be considered when it is used as the main data structure in order to minimise problems and ensure proper functioning of the overall system.

These limitations and considerations are detailed below:

- The DMF format requires certain coordinates to locate each measurement acquired by the instruments. This results in total dependence on the positioning system and requires correct and perfect synchronisation with all of the devices on the platform.
 - Although the DMF format is based on matrix structures, it is difficult to manage DMRs when the density of DMF cell measurements is high, resulting in the need for more advanced data structures for optimum processing (such as k-d trees or dynamic lists), which increases the overall complexity of the format. Moreover, although these structures always produce an improvement in processing times, it is particularly notable in extreme cases (very small DMF matrices with a high DMR density or very large DMF matrices with low DMR density).
 - The design of the DMF format ensures that adaptation to any type of platform is rapid and easy, but it is necessary to spend some time verifying its correct integration. The implementation and testing phases can lead to increased effort and cost by the company or institution.
-

- Storing each measurement with the coordinates provided by the positioning system can result in slightly higher memory and disk space consumption than conventional geocorrected formats, which estimate the locations of all of the pixels by means of references. However, by considering only the actual measurements acquired by the instruments, thereby avoiding duplication of information and the addition of spectra at 0 to represent areas located outside the scene of interest, the final size of the image in DMF format is reduced considerably, meaning that storage of this additional information does not actually represent a problem.
- The DMF cell size can be variable and independent from the integrity of the information. This provides several advantages, but using an incorrect DMF cell size can result in DMF matrices that are too small or too large and can harm the overall processing. In the event of not knowing these parameters or the flight conditions, it is possible to determine DMF cell size automatically.
- Displaying the DMF matrix can prove difficult in certain situations, especially if the platform has been subjected to highly intensive turbulence, since sub-regions not measured by the acquisition system appear as black pixels, which can hamper visual analysis. Moreover, not all of the information present in the DMF matrix is represented because, as there is a DMF cell - pixel correspondence, only one of the DMRs stored per DMF cell (usually the closest to the centre) is shown.
- Since it is not a conventional format, complete adaptation of all conventional image processing methodologies is required, from the most simple (erosion, expansion, etc.) to the most advanced (data fusion, change detection, etc.).
- When the DMF format is integrated with a real-time acquisition system, it is advisable to take into account the following considerations:
 - To set the DMF format as a type of output, it is necessary to provide the measurements and results generated together with their coordinates. To do so, perfect synchronisation is required between the computation system (and the implemented methodologies) and the positioning system.
 - The DMRs provided hold the location and instrumental value for each measurement, as well as possible results generated on board and in real time, which produces a slight increase in the size of the final information to transmit. The maximum rates and frequencies for the receipt and transmission of data provided by the ports used for correct overall functioning need to be checked, as does the way in which communications are performed between the components and modules implemented in the computation system.

In the next chapter, several solutions and improvements with respect to the DMF format's limitations and difficulties in certain situations are proposed. They include work and projects that will be carried out shortly and, therefore, can be considered as forming part of future lines of research.

Capítulo 8

Líneas futuras

El formato DMF proporciona una integración del 100% de la información recolectada por los instrumentos presentes en la plataforma, limitando la precisión espacial a la correspondiente del dispositivo de posicionamiento, y reduciendo los tiempos de cómputo y de almacenamiento, lo que permite una más fácil y rápida distribución.

Todas estas características han sido estudiadas mediante la realización de varios experimentos en el Capítulo 5 y en el Capítulo 6, en donde se ha demostrado que el formato DMF es una gran alternativa a los formatos de datos convencionales. Gracias al diseño propuesto, basado en la medición como registro DMR, y en el cual es posible añadir nuevos campos, el trabajo realizado queda abierto a futuros análisis, modificaciones y adaptaciones que permitirían la integración del formato DMF a nuevos campos científicos, y que reduciría considerablemente las limitaciones existentes.

Este conjunto de tareas, las cuales se pretenden desarrollar en un futuro próximo, pueden agruparse en las siguientes líneas:

- *Consideración de la morfología del footprint* (Sección 8.1).
- *Visualización dinámica de la matriz DMF* (Sección 8.2).
- *Adaptación del formato DMF para procesamiento paralelo* (Sección 8.3).
- *Utilización del formato DMF para aislamiento de áreas de interés* (Sección 8.4).
- *Optimización del módulo de procesamiento y de las comunicaciones* (Sección 8.5).

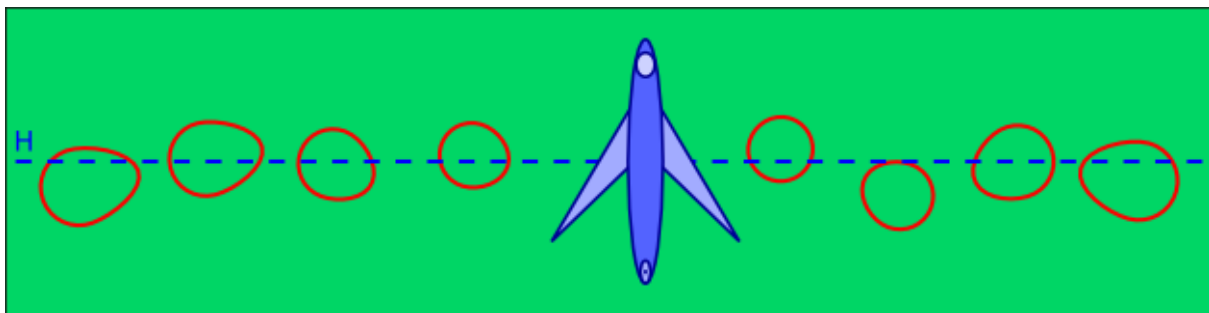
Las siguientes secciones de este capítulo plantean en detalle estas posibilidades que el formato DMF deja abiertas.

8.1. Consideración de la morfología del footprint.

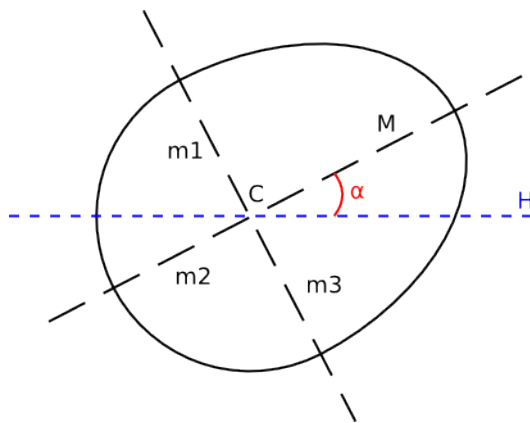
El área de integración sobre la corteza terrestre delimitado por el sensor en un instante t , y que conformará el valor instrumental de la medida, se denomina Ground Instantaneous Field Of View (GIFOV) o *footprint* (Sección 3.2.1), y suele considerarse como un círculo cuyo tamaño está directamente relacionado con la altitud de la plataforma, de tal manera que a mayores alturas, mayor área de *footprint*, y viceversa. El centro de dicha proyección circular se considera el mismo centro de la medición correspondiente, la cual es georreferenciada por el sistema de posicionamiento. Sin embargo, esta morfología sólo es válida para mediciones cerca del Nadir y sobre una superficie completamente plana.

Mediciones que se alejan del Nadir y que se proyectan sobre superficies irregulares suelen definirse como curvas cerradas de naturaleza ovoide, cuyo área será mayor cuanto mayor sea la altitud de la plataforma (Figura 8.1(a)). De acuerdo a la definición de una curva ovoide, ésta está compuesta por un centro de medición pero con un área de integración consistente en tres ejes menores de igual tamaño m , y un eje mayor M (Figura 8.1(b)). La curva delimitada por dichos ejes constituye una morfología del *footprint* más cercana a la realidad, la cual puede ser determinada de manera automática conociendo el ángulo IFOV y la altitud de la plataforma en el instante t .

La adaptación del formato DMF para la integración de la morfología del *footprint* es relativamente sencilla. Al estar basado en el registro DMR como unidad básica, y al ser éste un parámetro adjunto a la medición, bastaría con añadir nuevos campos para delimitar la morfología correspondiente. En un primer lugar, es posible pensar en la adición de dos nuevos campos m y M que definirían los ejes menor y mayor del ovoide. No obstante, sería una mejor opción considerar los cuatro ejes como independientes ($m1$, $m2$, $m3$ y M), ya que en terrenos irregulares la forma del ovoide podría ser también irregular (Figura 8.1(c)). Además, sería necesario un campo adicional para indicar el ángulo de rotación del *footprint* (α) respecto a la perpendicular del sentido del vuelo (H).



(a)



(b)

Latitud	Longitud
Datos Instrumento	
Ángulo	
M	
m1	
m2	
m3	
Siguiente	

(c)

Figura 8.1: Consideración de la morfología del *footprint* en el formato DMF.

(a) Adquisición de mediciones considerando una morfología ovoide del *footprint*. (b) Definición de la curva cerrada ovoide. (c) Adaptación del registro DMR a mediciones de morfología ovoide.

Esta adaptación permite un almacenamiento de la información mucho más precisa, aunque presenta un par de dificultades. La primera de ellas es la manera de visualizar la matriz DMF. Aunque se puede seguir con la política de corresponder la medición más cercana al centro de la celda DMF con un píxel en pantalla, sería interesante plasmar las características adicionales de los registros DMR respecto a la morfología del *footprint*. No obstante, es complicado plasmar irregularidades morfológicas sobre un patrón matricial regular. Una posible opción podría ser considerar un tamaño de píxel igual al mínimo diámetro del *footprint* con área menor. De esta forma se garantiza que la medición más pequeña ocupará un píxel completo, mientras que el resto necesitará más de uno para ser representado

correctamente, dejando patente de esta manera la diferencia de tamaños. Sin embargo, si la diferencia entre la distancia entre mediciones y el ancho mínimo de *footprint* es muy elevada, podrían resultar matrices DMF muy grandes (con un elevado porcentaje de celdas DMF a Null), con sus consecuencias correspondientes. La morfología es difícil de representar, y quizás lo más factible sería mostrar el aspecto del *footprint* de una medición determinada siempre de manera independiente.

El segundo problema deriva de la adición de nuevos campos al registro DMR, y es el incremento en consumo de memoria que requerirá tanto la matriz DMF como el formato en disco. Si se desea un mínimo de precisión, la información relativa a los cuatro ejes, más el ángulo, debe ser un tipo numérico real, que podríamos considerar del mismo tamaño que la latitud y la longitud: 4 Bytes. De esta forma, el tamaño total de un registro DMR pasaría de $n \times 2 + 8$ Bytes a $n \times 2 + 28$ Bytes. Esto supone un incremento variable dependiendo del número de bandas que conforman la imagen. Considerando el *dataset real*, para el caso con el mínimo de bandas (Mediterráneo, 22 bandas) el incremento de tamaño es aproximadamente un 140 %, comparado con el caso con el máximo de bandas (Cuprite y GMéxico, 224 bandas), donde el incremento de tamaño es de únicamente un 104 %.

8.2. Visualización dinámica de la matriz DMF.

La manera en la que la matriz DMF se visualiza en pantalla es mediante una correspondencia celda DMF - píxel, tal que la imagen RGB resultante tendrá las mismas dimensiones espaciales que la matriz DMF. El problema surge a la hora de seleccionar un registro DMR de la lista almacenada en cada celda DMF. La política considerada es visualizar el valor de banda correspondiente del registro DMR más cercano al centro de la celda DMF. Sin embargo, aunque se adapten diferentes y nuevas políticas, este problema seguirá existiendo mientras se considere esta correspondencia DMF - píxel. Al visualizar una única medición, el resto de registros DMR de la lista son ignorados, por lo que visualmente sólo se está representando un determinado porcentaje del total de mediciones capturado por el sensor.

Sería bastante interesante diseñar una metodología que permitiera la representación y visualización de todos los datos recogidos por los sensores. Debido a la dificultad existente de realizar esta tarea a través de la conversión a un sistema matricial RGB, la mejor solución se presenta como una visualización de la información interactiva. Dicha solución consiste en el muestreo de la información referente a los registros DMR almacenados en una determinada celda DMF cuando ésta es seleccionada. La manera de representar los registros DMR puede ser variada. Por un lado, se puede generar una vista visual de la lista de registros DMR correspondientes a la celda DMF, incluyendo toda la información conocida sobre ellos. Por otro lado, se puede generar un nuevo mapa a partir de dicha celda DMF, y situar las posiciones de cada uno de los registros DMR de manera relativa. El uso de un identificador permitiría un análisis detallado mediante una visualización dual (Figura 8.2).

8.3. Adaptación del formato DMF para procesamiento paralelo.

Las imágenes adquiridas por los sensores a bordo de aviones y satélites suelen implicar una cantidad inmensa de información que hace que las estructuras matriciales que las almacenan sean difíciles y costosas de manejar y procesar computacionalmente. Es por ello, por lo que el procesamiento paralelo adquiere un gran papel en este campo científico.

El procesamiento paralelo consiste en la utilización de p procesadores trabajando de forma conjunta para resolver un problema p veces teóricamente más rápido que un único procesador. Utilizando estructuras de datos matriciales, el particionado de la escena representada por una imagen geocorregida por métodos convencionales puede realizarse de manera regular, de acuerdo al número de hilos de procesamiento. Sin embargo, si la escena ha sido adquirida mediante mediciones distribuidas de manera irregular, y se utilizan estructuras de datos más complejas, dicho particionado puede variar,

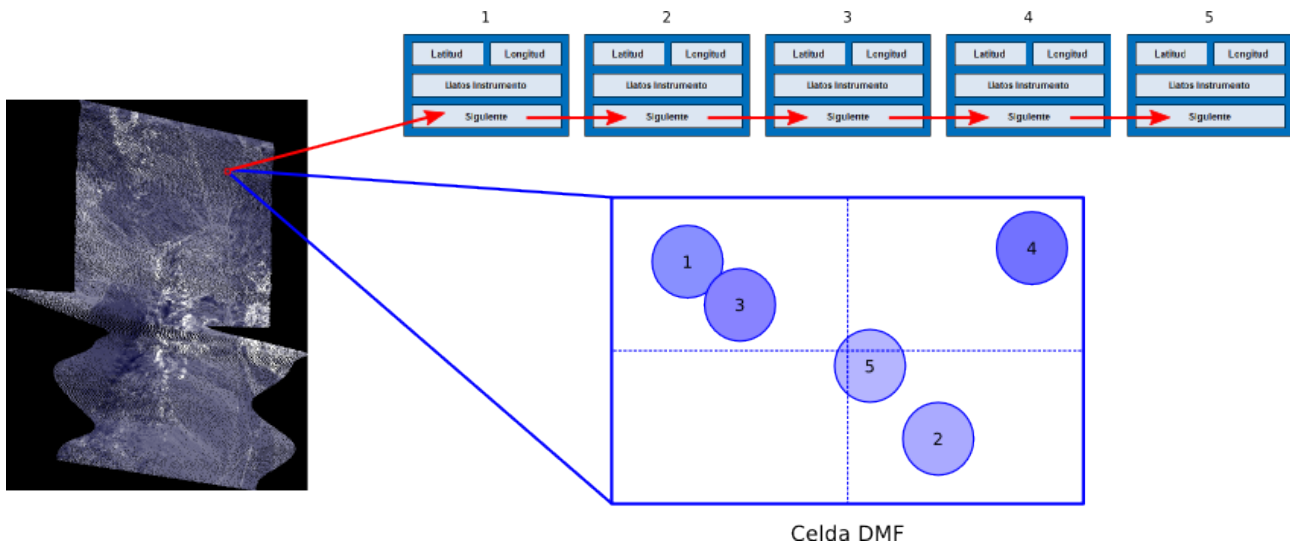


Figura 8.2: Visualización interactiva de una determinada celda DMF de la matriz DMF (vista dual).

seleccionando las subáreas a tratar por los hilos de ejecución acorde a la densidad de mediciones.

Las estructuras de datos arbóreas son las más adecuadas para este propósito, ya que permiten la división de la escena de manera irregular. Para el caso de procesamiento de imágenes de teledetección, el tipo arbóreo más adecuado es el Quad-tree. Esta estructura permite la partición de la escena de acuerdo a los puntos cardinales Noreste (NE), Noroeste (NW), Sureste (SE) y Suroeste (SW). Sus nodos almacenarán la información relativa a las mediciones adquiridas por los sensores (datos instrumentales y coordenadas espaciales (*longitud, latitud*)), y dispondrán de cuatro hijos. Dichos hijos representarán subáreas localizadas en posiciones NE, NW, SE o SW de acuerdo al nodo actual. Esto implica árboles resultantes más anchos, pero menos profundos, por lo que la localización de las mediciones se realizan de manera más sencilla, organizada y rápida.

La Figura 8.3 muestra la adaptación de la estructura arbórea Quad-tree para la representación de una escena con determinadas posiciones de mediciones localizadas.

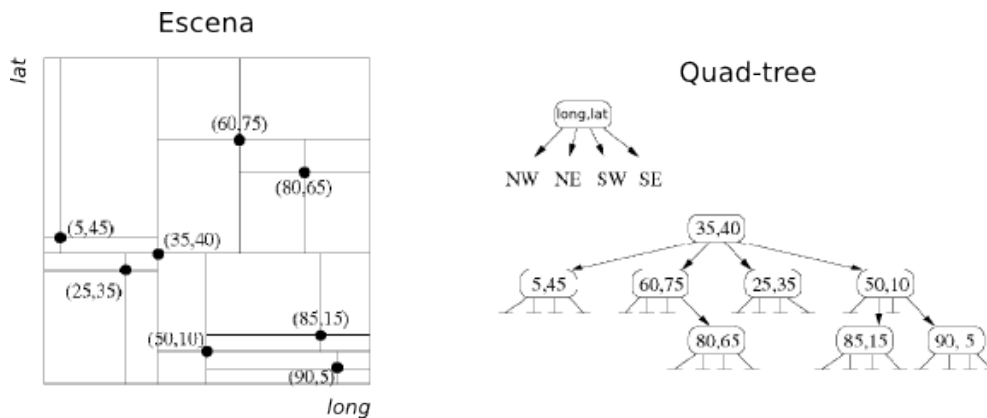


Figura 8.3: Representación de una escena simple mediante la estructura arbórea Quad-tree.

Es interesante combinar las características y ventajas del formato DMF con las que ofrece la estructura Quad-tree. De esta manera, se propone una estructura híbrida, denominada QTDMF, que utiliza la matriz DMF como almacenamiento de registros DMR, y el árbol Quad-tree como indexación de los datos. Se partirá de esta última estructura de datos, la cual estará compuesta por nodos, denominados *Quad-tree DMF Nodes* (QTDMFN), que no almacenarán la información instrumental y las coordenadas espaciales de las mediciones, sino que estarán compuestos por los siguientes campos (Figura 8.4(a)):

- Coordinadas del área que abarcan: Mínima longitud (m_L), máxima longitud (M_L), mínima latitud (m_l) y máxima latitud (M_l).
- Número total de mediciones insertadas como registros DMR (incluyendo a todos sus nodos descendientes).
- Cuatros hijos nodo NE, NW, SE y SW, correspondientes a las cuatro subáreas comprendidas en dichos puntos cardinales para el área abarcada:

$$NE = \left[\frac{M_L + m_L}{2}, M_l \right] : \left[M_L, \frac{M_l + m_l}{2} \right]$$

$$NW = [m_L, M_l] : \left[\frac{M_L + m_L}{2}, \frac{M_l + m_l}{2} \right]$$

$$SE = \left[\frac{M_L + m_L}{2}, \frac{M_l + m_l}{2} \right] : [M_L, m_l]$$

$$SW = \left[m_L, \frac{M_l + m_l}{2} \right] : \left[\frac{M_L + m_L}{2}, m_l \right]$$

Los nodos QTDMFN hojas serán nodos especiales cuyos hijos estarán compuestos por cuatro matrices DMF en vez de otros nodos QTDMFN (Figura 8.4(b)). Cada matriz DMF abarcará el subárea NE, NW, SE ó SW del nodo correspondiente y, siguiendo las pautas de esta metodología, almacenarán registros DMR en sus celdas DMF. Si no existen mediciones en alguna de las cuatro subáreas, la matriz DMF correspondiente permanecerá a *Null*. Con este diseño, los datos adquiridos siempre se encontrarán en las hojas del árbol en formato matriz DMF.

Además, la estructura QTDMF poseerá una variable global que almacenará un valor configurable y que indicará el número máximo de registros DMR en una matriz DMF. Si una matriz DMF supera este valor, se convertirá en un nodo QTDMFN convencional, y los registros DMR se distribuirán entre las cuatro matrices DMF NE, NW, SE y SW resultantes. Esta variable proporciona principalmente dos ventajas:

1. La georreferenciación de los datos y la creación de la estructura de una manera dinámica, y en tiempo real, según se van adquiriendo las mediciones sobre la zona de estudio.
2. Un completo control de la profundidad y complejidad de la estructura QTDMF, y del tamaño y densidad de las matrices DMF resultantes. De esta manera, la dinámica y gestión de la estructura es adaptable a las necesidades de la misión.

El diseño de la estructura QTDMF facilita el procesamiento paralelo de los datos, ya que éstos se encuentran aislados, en formato DMF, en los nodos hojas. Cada una de las matrices DMF de los nodos hojas implica un área de la escena independiente de determinado tamaño, que puede ser tratada por un procesador de manera individual.

Para acceder a los nodos hojas (y por tanto, a los datos), es necesario recorrer en profundidad el árbol. Sin embargo, es posible añadir una lista de punteros a cada una de las matrices DMF almacenadas para así conseguir un acceso directo a los mismos. Esta estructura adicional, denominada *Quad-tree DMF List* (QTDMFL) permite a cada procesador acceder directamente a sus datos correspondientes eliminando la necesidad de tener que descender por el árbol de acuerdo a las condiciones espaciales. Esto facilita considerablemente la división de tareas y la asignación de recursos.

La lista QTDMFL puede ser ordenada según diferentes parámetros, como puede ser el número de registros DMR almacenados en las matrices DMF. De esta forma, el primer elemento de la lista QTDMFL haría referencia, por ejemplo, a la matriz DMF con mayor número de registros DMR,

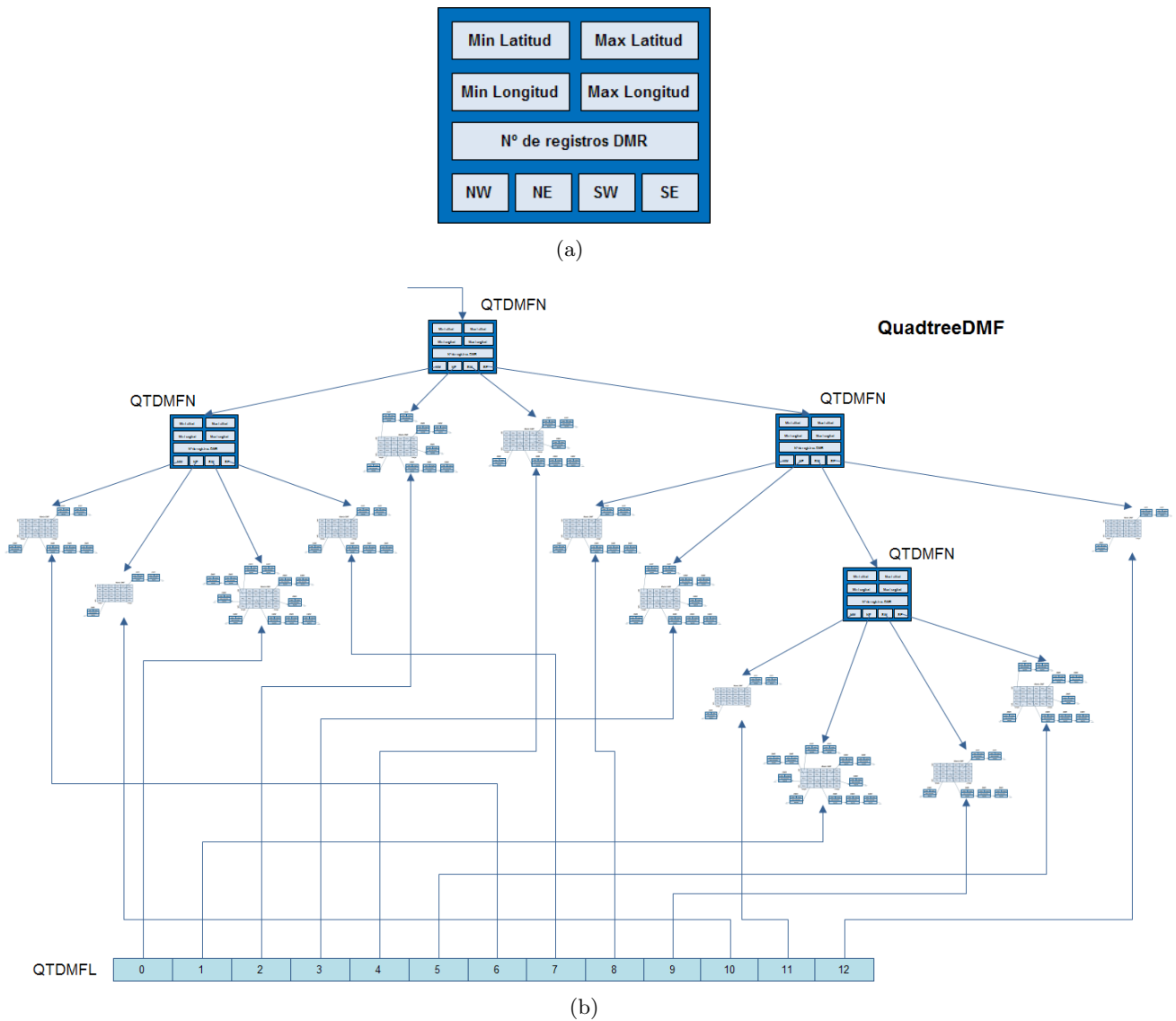


Figura 8.4: Adaptación del formato DMF para el procesamiento paralelo basado en el particionamiento irregular de la escena mediante variantes del árbol Quad-tree.
 (a) Nodo QTDMFN. (b) Estructura QTDMF.

mientras que el último elemento lo haría a la matriz con menor número de registros DMR. Esto es de mucha utilidad si el sistema de computación paralelo es heterogéneo, es decir, está compuesto por computadores con diferentes características y rendimiento, pudiendo ser las matrices DMF con mayor número de registros DMR (más información) asignadas a sistemas de cómputo más liberados o potentes, mientras que las matrices DMF con menor número de registros DMR (menos información) pueden ser asignadas a sistemas de cómputo con más carga de trabajo o menos potentes.

De esta forma, es interesante analizar siempre la densidad de registros DMR por elemento de la celda QTDMFL para realizar diferentes estrategias que optimicen la asignación de los recursos de acuerdo a las características del sistema de computación paralelo.

8.4. Utilización del formato DMF para aislamiento de áreas de interés.

En este documento de tesis, el formato DMF ha sido desarrollado con el objetivo principal de minimizar los errores espaciales cometidos en el proceso de geocorrección, así como la posibilidad de ofrecer un procesamiento rápido e íntegro de los datos recogidos por los instrumentos y una eficiente

distribución de los mismos a la comunidad científica. Su versatilidad permite que pueda ser adaptada de manera sencilla a situaciones más complejas, como pueden ser la fusión de datos y la detección de cambios, mediante la adición de nuevos campos al registro DMF.

El hecho de almacenar únicamente la información procedente de los dispositivos presentes en la plataforma, evitando la duplicidad de mediciones y espectros a 0 para rellenar píxeles cuyas áreas no han sido medidas o están localizadas fuera de la escena, hace del formato DMF una estructura liviana y menos costosa de manipular. Incluso es posible analizar cada medición adquirida y mediante una o varias determinadas condiciones, considerar si el registro DMR correspondiente es apto de ser considerado y almacenado en la matriz DMF. En la Sección 6.1 se presentó un experimento similar, en el cual las mediciones que superaban un determinado nivel de presencia de nubes eran ignorados. De esta manera, la matriz DMF resultante únicamente estaba conformada por datos que cumplían una serie de requisitos previos.

Esta situación puede ser extrapolada a otros casos en la que la información recogida por los instrumentos tengas sólo un interés parcial por parte de los expertos. Sirva como ejemplo analistas interesados únicamente en los datos y mediciones correspondientes a mares u océanos, en la monitorización de la vegetación terrestre en la que la información marítima puede no ser de interés, o en la presencia de gases y aerosoles en la atmósfera terrestre. Sería posible descartar las mediciones que no tienen lugar en el estudio (y que además pueden influir negativamente en los resultados finales) y, así, no tener que trabajar con una imagen completa. Mediante la eliminación de los registros DMR que no cumplan los requisitos, el área de estudio se reduce considerablemente. De esta forma, es posible, a partir de la misma información, generar matrices DMF únicamente correspondientes a mediciones marítimas y oceanográficas, a la superficie terrestre, o bien a los aerosoles presentes en la atmósfera. De la misma manera, el analista podría definir de manera interactiva un polígono que abarque el subárea de interés. Todos los registros DMR situados fuera de ese polígono pueden ser descartados, limitando espacialmente la información almacenada en la matriz DMF.

La Figura 8.5 muestra un ejemplo del aspecto visual de la matriz DMF resultante, en el cual el experto analista ha aislado diferentes áreas de interés de acuerdo a sus necesidades, conservando únicamente los registros DMR correspondientes a información marítima y oceanográfica (Figura 8.5(a)), a información correspondiente a la superficie terrestre (Figura 8.5(b)), o a información correspondiente a los aerosoles y gases presentes en la atmósfera (Figura 8.5(c)). De igual modo, el experto ha seleccionado un área de interés mediante la definición de un polígono de manera interactiva, siendo en este caso el norte de África el objeto de estudio (Figura 8.5(d)).

8.5. Optimización del módulo de procesamiento y de las comunicaciones.

El sistema desarrollado, el cual emulaba un módulo de procesamiento SpaceCube consistía en un par de tarjetas FPGA interconectadas, cada una de ellas albergando una metodologías que realizaba una serie de cálculos espectrales a un nivel de hardware. Ambas tarjetas FPGA trabajaban en paralelo y de manera solapada, pero las comunicaciones siempre se realizaban de manera unidireccional. Los datos entraban por la primera tarjeta, eran tratados por ésta, pasaban a la segunda tarjeta, se les sometía a cálculos adicionales, y los resultados generados se mostraban como salida de ésta última en formato DMF.

De esta forma, el sistema desarrollado podía ser considerado como un dispositivo de procesamiento en el que las mediciones eran procesadas en cadena. Para metodologías que no requieren vecindad, como las implementadas, este sistema es más que suficiente. Sin embargo, quedaría bastante limitado para metodologías que sí necesitaran de mediciones próximas para realizar sus cálculos. Esta no es una tarea sencilla cuando se tratan de sistemas a bordo en tiempo real, donde las mediciones llegan de manera consecutiva, no están posicionadas sobre ninguna estructura regular de carácter matricial, y además no se tiene una constancia determinada de cuándo va a estar disponible una vecindad

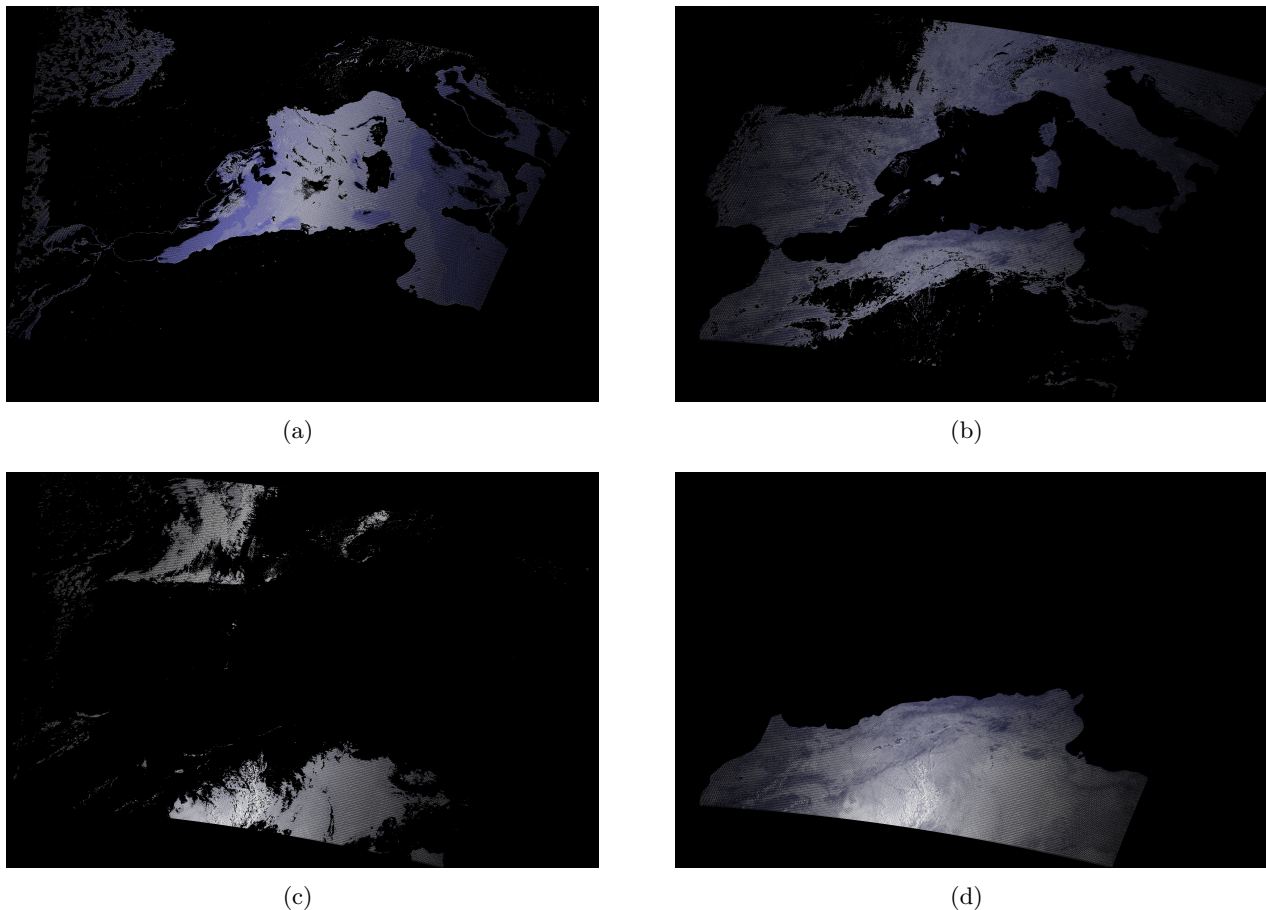


Figura 8.5: *Representación de áreas de interés mediante el formato DMF para la escena de Mediterráneo (Banda 4).*

(a) Restringido a superficies marinas y oceánicas. (b) Restringido a corteza terrestre. (c) Restringido a aerosoles. (d) Restringido a polígono personalizado (África).

completa (para iniciar el procesamiento de la medición central). Sería necesario, por tanto, modificar la arquitectura hardware para dar almacenar temporalmente las mediciones según van entrando al sistema. La mejor solución involucra a la memoria DDR2 como sistema de almacenamiento de dichas mediciones, y el desarrollo de un nuevo módulo que la gestione. Este módulo englobará al submódulo *FIFO_E*, garantizando la sincronización de los datos proporcionados por los instrumentos y el sistema de posicionamiento, y los almacenará en determinadas posiciones de memoria. Su complejidad sería bastante elevada, ya que debería llevar de alguna manera un registro de en qué posición de la memoria DDR2 se va almacenando cada medición recibida, y que podría ser implementado mediante listas de registros internos o bloques RAM. Submódulos independientes podrían estar monitorizando una entrada de registro cada uno, controlando en qué momento su medición correspondiente posee el número de vecinos (basándose en la distancia Euclídea) apto para realizar su procesamiento. Una vez una medida es apta de ser procesada, ésta y sus vecinas son leídas de la memoria DDR2 y enviadas a la metodología a través de $n \times n$ puertos (considerando un *kernel* de $n \times n$). El resultado generado por la metodología será proporcionado al exterior siguiendo el mismo procedimiento que el esquema original.

Otra optimización que sería posible implementar en el sistema de procesamiento propuesto sería la posibilidad de realizar comunicaciones entre las dos tarjetas FPGA de manera bidireccional, no únicamente unidireccional. Es decir, que cada tarjeta sea capaz de enviar y recibir datos de la otra. Esto implica la elaboración de un protocolo de comunicaciones adicional, que permita la correcta sincronización de la conexión utilizada con el objetivo de evitar el solapamiento de datos. También sería recomendable utilizar una política *maestro - esclavo*, en la que una realiza la gestión de las

transmisiones de los datos, así como del procesamiento. Esto además, abriría la posibilidad de incluir más tarjetas FPGA en el sistema, formando un *cluster*.

La tercera mejora que se le podría aplicar al diseño, aprovechando la anterior, sería la definición de un protocolo de comunicaciones que permitiera realizar una más elaborada detección (y corrección) de errores, muy importante sobre todo si el sistema está destinado a realizar sus labores en un entorno radiactivo como el Espacio. Por tanto, es interesante ir más allá de técnicas como la inserción del *bit de paridad* (como el utilizado en las conexiones RS-232), y empaquetar los datos en formatos específicos que gestionen *acuse de recibo* (ACK) y que hagan uso de campos adicionales como el *checksum*. Hay que considerar que se garantiza la integridad de los datos desde que sale de una tarjeta FPGA hasta que llega a la otra, pero en caso de fallo (y no recuperación de la información) es necesario enviar el paquete de nuevo, incrementando los tiempos empleados en comunicaciones. Como la aparición de fallos es una variable aleatoria, sería recomendable realizar un estudio exhaustivo sobre el entorno en que el será ejecutado el sistema, estableciendo un porcentaje de tolerancia y que debe ser considerado en los puntos críticos presentados en la Sección 6.1.3 para evitar la saturación los buffers de entrada por una incorrecta sincronización.

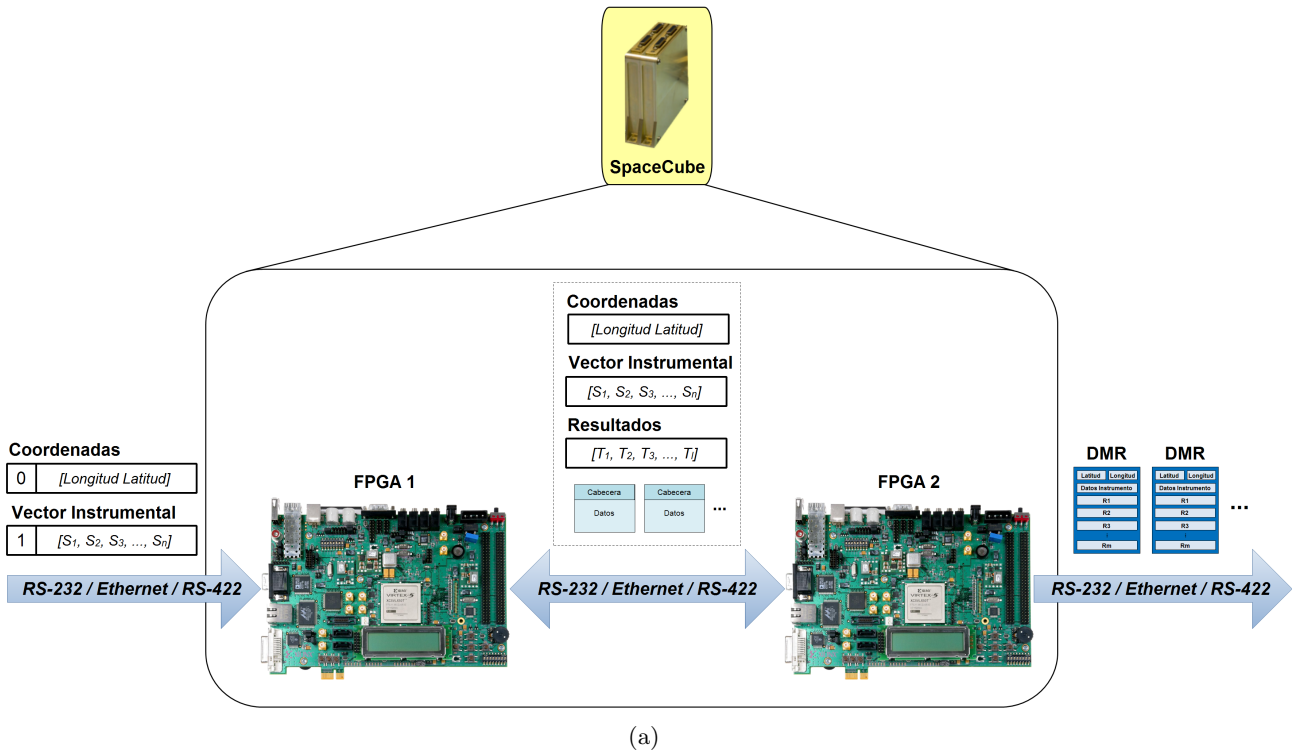
Una cuarta optimización implicaría la eliminación del procesador PowerPC y la arquitectura basada en un bus central. Las comunicaciones se realizarían a través de interfaces dedicados implementados directamente en la FPGA. Estos interfaces se encargarían tanto de recoger la información del exterior para suministrárselos a la metodología, como de enviar los resultados y los datos adicionales generados por ésta.

La Figura 8.6 representa un nuevo diseño de la conexión de las tarjetas FPGA y de los módulos hardware implementados en el chip reconfigurable, con el objetivo de dar cabida a las cuatro optimizaciones propuestas con anterioridad.

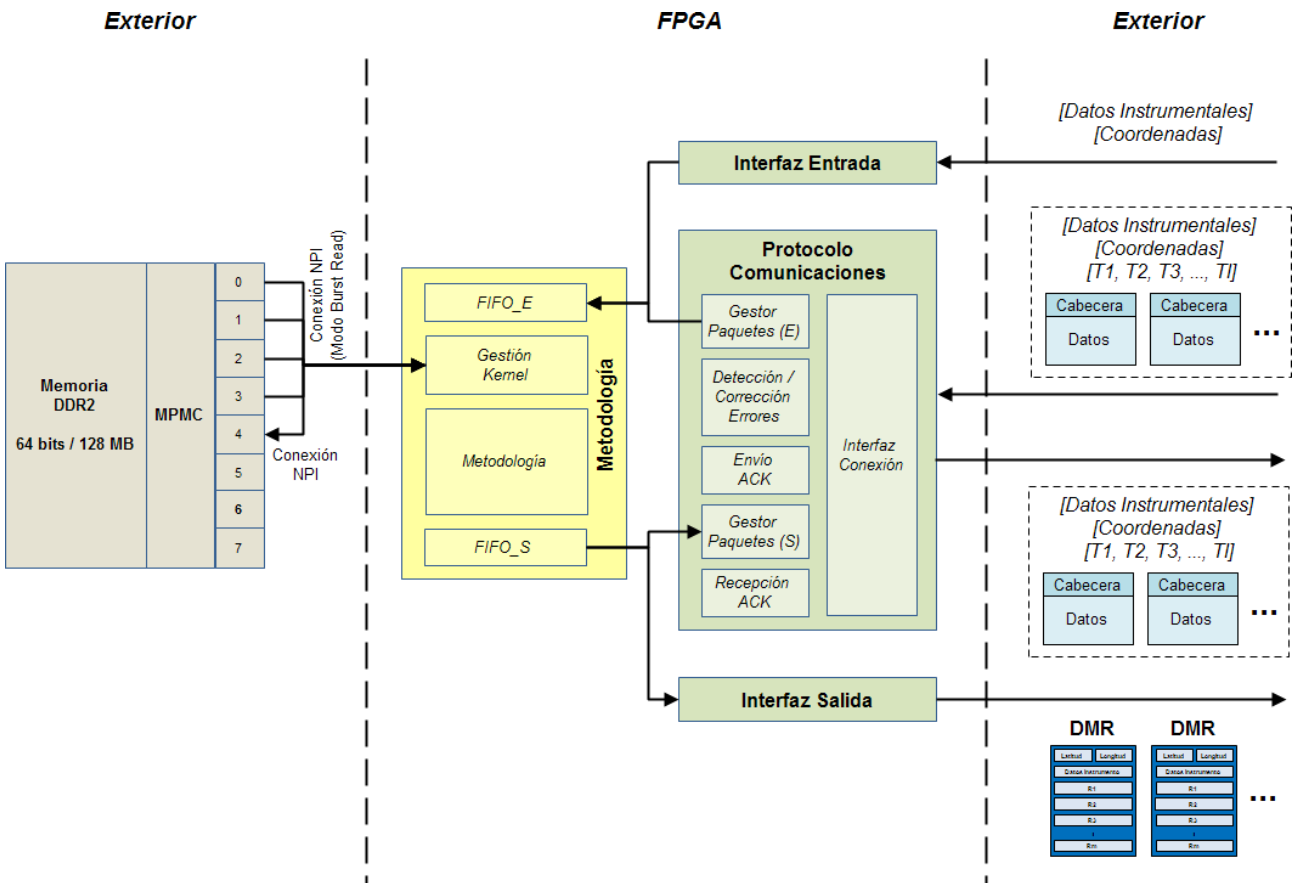
En un diseño a alto nivel (Figura 8.6(a)), se presenta una interconexión entre las dos tarjetas FPGA muy similar al original, pero considerando que para esta optimización la comunicación realizada entre ambas es bidireccional. Como se implementa un protocolo de comunicaciones más elaborado (que permite detección y corrección de errores), los datos no se envían y reciben tal cual, sino que van agrupados en paquetes de datos que poseen adjunta una cabecera (con información adicional dependiente del protocolo). Esta política se utilizaría únicamente en la interconexión entre ambas tarjetas. Las entradas (ráfagas alternas de coordenadas y valores instrumentales) y las salidas (en formato registro DMR) del sistema se mantendrían como en el diseño original, aunque cabría la posibilidad de implementar protocolos de comunicaciones adicionales, dependiendo de los dispositivos emisores y receptores de la información.

La implementación en los chips FPGA (Figura 8.6(b)) supone el desarrollo de nuevos módulos y submódulos, así como de la modificación de algunos de los ya existentes. El diseño hardware queda, de esta manera, constituido por cuatro componentes principales:

- *Interfaz Entrada*: Implementa la interfaz del conector utilizado para gestionar la entrada de datos del sistema general. Como dichos datos proceden del exterior del dispositivo propuesto, estarán compuestos por ráfagas de coordenadas y valores instrumentales, los cuales serán proporcionados al submódulo *FIFO_E* para su correcta sincronización y posterior almacenamiento.
- *Protocolo Comunicaciones*: Módulo que se encarga de gestionar las comunicaciones entre las tarjetas FPGA, recogiendo los datos de entrada, proporcionados por el exterior, y enviando los resultados generados por la metodología implementada.
 - *Interfaz Conexión*: Implementa la interfaz del conector utilizado para realizar las comunicaciones entre las tarjetas FPGA. Recoge los paquetes recibidos del exterior y se los pasa al submódulo *Gestor Paquetes (E)* para su interpretación; envía los resultados generados por la metodología tras su empaquetación por parte del submódulo *Gestor*



(a)



(b)

Figura 8.6: Optimización del diseño del sistema basado en SpaceCube propuesto.
 (a) Emulación de un slice de procesamiento SpaceCube. (b) Diagrama genérico del sistema hardware implementado en las tarjetas FPGA.

Paquetes (S); y gestiona las transferencias de *acuses de recibo (ACK)*, proporcionándoselos a los submódulos *Envío ACK* y *Recepción ACK*.

- *Gestor Paquetes (E)*: Implementa el protocolo de comunicaciones. Recibe el paquete del submódulo *Interfaz Conexión*, interpreta la cabecera y proporciona los datos al submódulo *Detección / Corrección Errores*. Si no se detecta ningún error en la transferencia (o se detecta pero se corrige con éxito), los datos se transmiten al módulo *Metodología* para su procesamiento. En caso contrario, el paquete es descartado.
 - *Detección / Corrección Errores*: Implementa el algoritmo de detección y corrección de errores correspondiente. Para ello, recibe los datos del paquete del submódulo *Gestor Paquetes (E)* y los somete a análisis. El resultado obtenido (paquete aceptado o rechazado) es proporcionado de vuelta a dicho submódulo, y también al submódulo *Envío ACK*.
 - *Envío ACK*: Este submódulo recibe el resultado del análisis realizado por el submódulo *Detección / Corrección Errores* y envía, a través del *Interfaz Conexión*, un paquete con valor 1 ($ACK=1$) si no se encontraron errores en la transmisión, o un 0 ($ACK=0$) en caso contrario.
 - *Gestor Paquetes (S)*: Implementa el protocolo de comunicaciones. Recibe los resultados generados por la metodología (y datos adicionales) y los empaqueta de acuerdo al protocolo establecido. Esto incluye la adición de una cabecera por paquete construido, que puede incluir información relativa al control de errores (por ejemplo, un *checksum*). Tras la generación del paquete, éste es enviado al exterior a través del *Interfaz Conexión* y comunicado al submódulo *Recepción ACK*, quedando a la espera de la respuesta de éste último. Si la respuesta no es favorable, el paquete se vuelve a enviar, siguiendo el mismo procedimiento. En caso contrario, el paquete se descarta y se prepara la elaboración del siguiente.
 - *Recepción ACK*: Este submódulo recibe una señal del submódulo *Gestor Paquetes (S)* (para indicar que se ha enviado un nuevo paquete), y queda a la espera de una respuesta de *acuse de recibo (ACK)* por parte de la otra tarjeta FPGA, y a través del *Interfaz Conexión*. Si la respuesta es un paquete con valor 1 ($ACK=1$), entonces la transmisión de los datos se realizaron correctamente, informando de este hecho al submódulo *Gestor Paquetes (S)*. Si la respuesta es un paquete con valor 0 ($ACK=0$), o bien pasa un tiempo determinado sin recibirla (por lo que hará uso de un temporizador), entonces se informará negativamente al submódulo *Gestor Paquetes (S)* para que vuelva a reenviar el paquete.
- *Módulo Metodología*: Módulo que se encarga de realizar el procesamiento de los datos de acuerdo a la metodología correspondiente. Está compuesto por los siguientes submódulos:
- *FIFO_E*: Este submódulo permanece inalterado respecto al diseño original (Sección 6.4.2.1), permitiendo la sincronización de los diferentes tipos de ráfagas de datos mediante un sistema basado en memorias FIFO.
 - *Gestión Kernel*: Es el submódulo más complejo del sistema, pues sobre él recae las principales responsabilidades para un correcto funcionamiento global. Por un lado, almacena los datos sincronizados proporcionados por el submódulo *FIFO_E* en la memoria DDR2 a través del componente MPMC. Un registro interno basado en bloques de memoria RAM permite indexar la localización espacial de la medición correspondiente, y la posición de memoria en la que es almacenada. Por otro lado, tras el almacenamiento, debe comprobar, usando dicho índice, que mediciones ya almacenadas en memoria son candidatas a ser vecinas de la recién recibida. Esto puede realizarse mediante la ejecución de procesos modulares que implementen, entre otros tipos de cálculos, la distancia Euclídea, incrementando su complejidad, y por supuesto los tiempos de procesamiento. Una vez se hayan encontrado un total de $n \times n$ candidatas (siendo $n \times n$ el tamaño del *kernel*), sus valores instrumentales correspondientes serán leídos de la memoria DDR2 mediante la conexión NPI (con el modo Burst Read activo) y enviados, a través de $n \times n$ puertos dedicados, al submódulo *Metodología*.

- *Metodología*: Implementa la metodología basada en vecindad correspondiente. Para ello, recibe todos los valores instrumentales pertenecientes al *kernel* del submódulo *Gestión Kernel* y, tras su procesamiento, envía el resultado obtenido al submódulo *FIFO_S*.
 - *FIFO_S*: Este submódulo permanece inalterado respecto al diseño original (Sección 6.4.2.1), permitiendo la sincronización de los diferentes tipos de resultados generados mediante un sistema basado en memorias FIFO.
- *Interfaz Salida*: Implementa la interfaz del conector utilizado para gestionar la salida de datos del sistema general. Estos datos se proporcionan al exterior del dispositivo propuesto en formato registro DMR.

Tal y como muestra el diseño a alto nivel, las comunicaciones entre ambas tarjetas FPGA se realizarán siguiendo un determinado protocolo, que no tiene por qué ser el mismo que el utilizado a la hora de enviar y recibir datos del exterior del dispositivo propuesto. Las entradas del sistema (ráfagas de coordenadas y valores instrumentales) se realizarán a través de la primera tarjeta FPGA, por lo que ésta deberá tener implementados los módulos *Interfaz Entrada* y *Protocolo Comunicaciones* obligatoriamente. Por el otro lado, las salidas del sistema (como registros DMR) se realizarán a través de la segunda tarjeta FPGA, por lo que ésta deberá tener implementados los módulos *Protocolo Comunicaciones* e *Interfaz Salida* obligatoriamente.

Con este nuevo diseño, se hace relativamente complicado realizar un estudio teórico sobre el rendimiento global, ya que su funcionamiento, así como los tiempos de ejecución totales, dependen directamente de diferentes parámetros externos de carácter aleatorio, como pueden ser el orden de llegada de las mediciones (que influye directamente sobre la búsqueda de la vecindad), o bien la frecuencia de errores. Sería interesante analizar cada uno de estos dos aspectos de manera aislada e independiente, para comprobar con cuál de ellos el sistema global se hace inestable más rápidamente y con más facilidad.

Anexo A

Módulos de circuitería detallados

Este capítulo tiene como objetivo mostrar en detalle todos los módulos de los circuitos de bajo nivel presentados a lo largo del documento de tesis.

Se divide en dos apartados principales. El primero de ellos (Sección [A.1](#), presenta los módulos internos correspondientes a la optimización de las comunicaciones realizadas dentro de la tarjeta FPGA a través del periférico MEDiC; mientras que el segundo (Sección [A.2](#)), muestra los módulos internos de las metodologías implementadas para la emulación del sistema de procesamiento SpaceCube.

A.1. Segunda aproximación a SpaceCube. Optimización del periférico MEDiC.

Este apartado muestra de manera detallada los módulos implementados para la optimización de las comunicaciones entre los periféricos situados en el interior de la tarjeta FPGA y el procesador PowerPC, tanto para la primera arquitectura hardware propuesta, la cual hacía uso de los 8 puertos del componente MPMC (Sección [A.1.1](#)), como para la segunda, la cual hacía uso de 5 de los puertos mediante conexión de alta velocidad NPI (Sección [A.1.2](#)).

A.1.1. *Primer diseño de la arquitectura hardware.*

A continuación se definen los módulos implementados para el desarrollo de la primera arquitectura hardware, el cual hace uso de la memoria DDR2 como dispositivo intermediario para la optimización de las comunicaciones internas entre los periféricos y el procesador PowerPC. Para ello, se dedican siete puertos del componente MPMC (a través de conexiones NPI) para el periférico MEDiC, y uno para el procesador.

El módulo MPMC Control Logic 1.

El módulo MPMC Control Logic 1 (Figura [A.1](#)) realiza la lectura de 10 valores de espectro de un determinado puerto de memoria. Ya que la conexión NPI trabaja con 64 bits, y que en cada lectura se obtienen dos valores de espectro al mismo tiempo, será necesario realizar un total de 5 lecturas. Está compuesto por los siguientes módulos:

- Un contador hardware (Counter) de 3 bits que se incrementa en uno cada vez que una lectura ha sido realizada. Ya que trabaja con 3 bits, puede contar hasta 8.

- Una lógica central compuesta por puertas NAND, OR, NOT y AND que controlan las lecturas de memoria. Se habilita cuando el contador es menor que 5, cuando el módulo Lógica de Control MPMC 1 se ejecuta por primera vez (la señal E estará activa por sólo un ciclo de reloj), y el módulo MPMC Controller ha terminado de leer (el módulo MPMC Control Logic 1 no debería estar funcionando mientras este módulo está en ejecución, es decir, mientras se está realizando la lectura). Una vez que el contador llega a 5 (entrada 100 para la puerta NAND), esta lógica central evita que el módulo MPMC Control Logic 1 continúe funcionando y la ejecución finaliza).
- Un decodificador (DEC) de 3 bits. Sólo una de sus señales de salida estará activa de acuerdo al contador. Ya que solo se realizarán 5 lecturas, las tres últimas salidas no serán utilizadas.
- Contador de Programa (PC). Un simple contador de 32 bits que incrementa en 8 bytes (64 bits) la dirección de lectura base ($Address_Read$). En cada ejecución (lectura), la dirección indica la posición de memoria en la que leer el próximo dato. Cada dirección de lectura es proporcionada al módulo MPMC Controller para que realice la lectura del dato correspondiente.
- Registros R0... R9 para almacenar los 10 valores leídos de memoria. Es necesario considerar que el dato leído consta de 64 bits, por lo que contiene dos valores de espectro de 32 bits. De esta manera, el dato leído es dividido en los dos valores. El primero es almacenado en el registro derecho correspondiente, y el segundo en el registro izquierdo correspondiente. La puerta AND habilita tanto los registros derecho e izquierdo correspondiente para su escritura. El registro RDone almacenará un 1 cuando los dos últimos valores de espectro hayan sido almacenados en R8 y R9 (la última lectura se haya realizado). Esto indicará al exterior del módulo MPMC Control Logic 1 que las lecturas de los 10 valores de espectro ya han sido realizados.

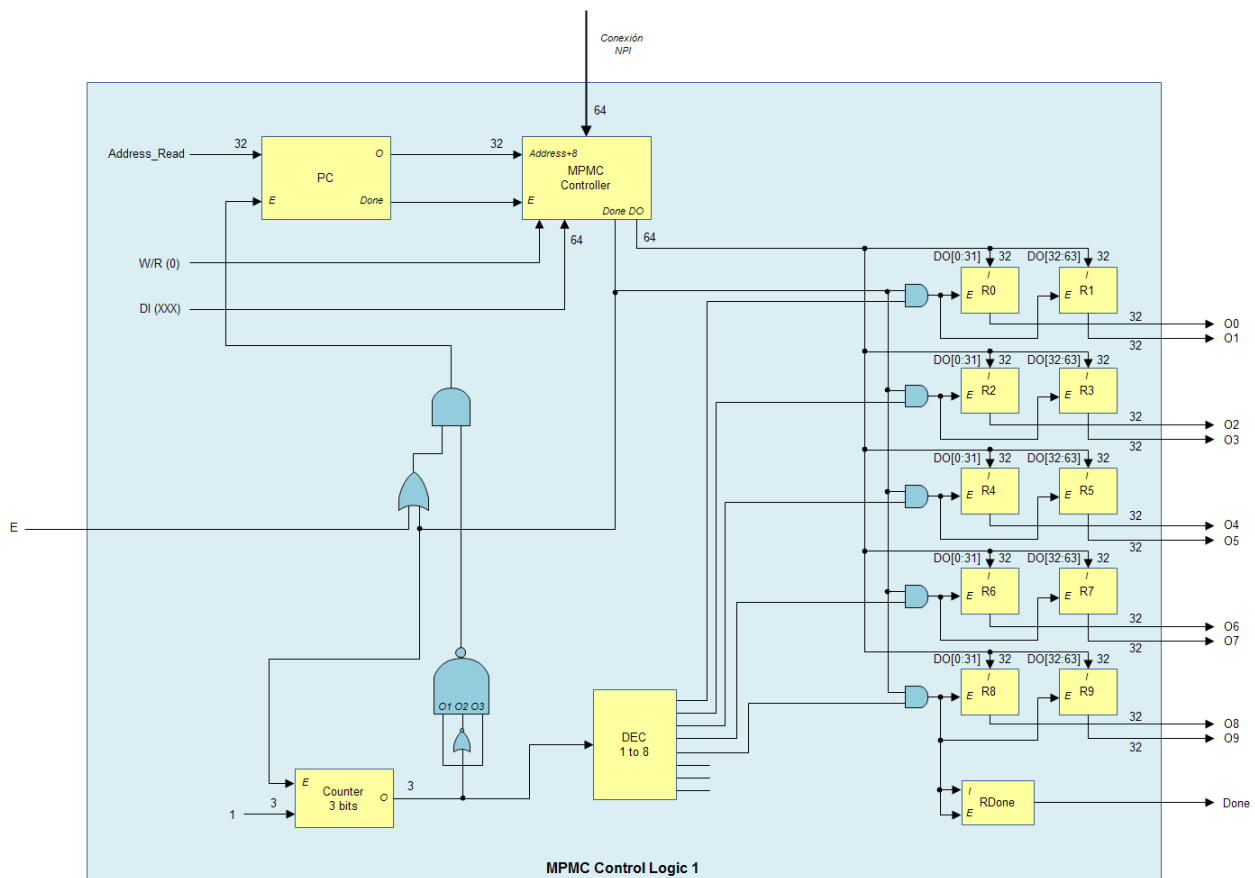


Figura A.1: Diagrama del módulo MPMC Control Logic 1, responsable de obtener de memoria los 10 valores de espectro a través de cada puerto MPMC.

El módulo MPMC Controller.

Este módulo gestiona el componente MPMC, y ha sido implementado como una máquina de estados que lee y escribe datos en memoria. El componente MPMC es un interfaz proporcionado por Xilinx para utilizar la memoria DDR2. Las señales necesarias para las acciones de escritura y lectura se muestran en las Figuras A.2(a) y A.2(b), respectivamente. El módulo MPMC Controller gestiona estas señales cuando se requiere la lectura o escritura de un dato en memoria.

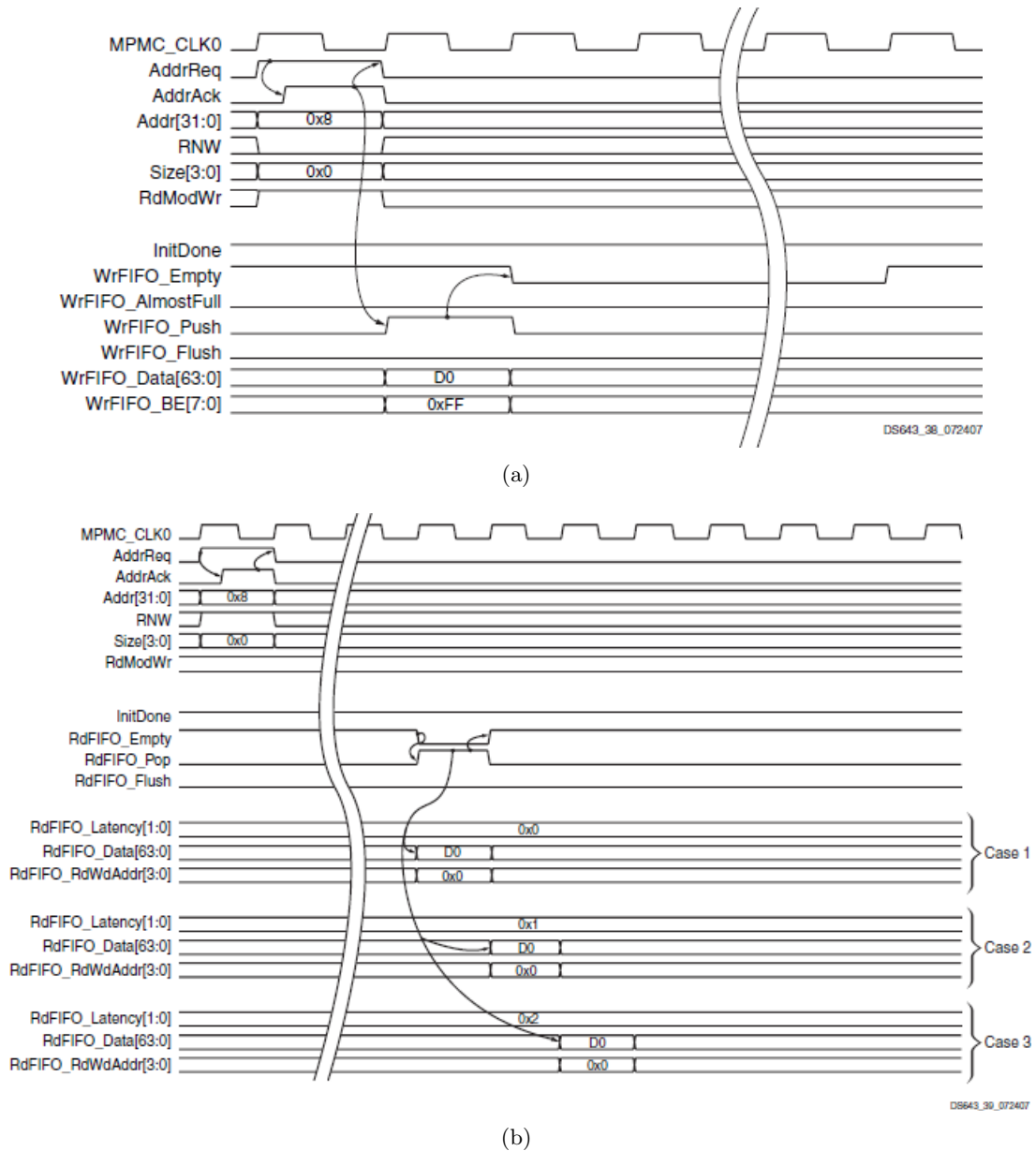


Figura A.2: Señales de control del componente MPMC.

(a) Señales para realizar la escritura de una palabra de 64 bits en memoria. (b) Señales para realizar la lectura de una palabra de 64 bits de memoria.

El módulo MPMC Control Logic 2.

El módulo MPMC Control Logic 2 (Figura A.3) realiza la lectura de 8 valores de espectro de un determinado puerto de memoria. Ya que la conexión NPI trabaja con 64 bits, y que en cada lectura se obtienen dos valores de espectro al mismo tiempo, será necesario realizar un total de 4 lecturas.

Este módulo tiene el mismo diseño que el módulo MPMC Control Logic 1 (Figura A.1), pero con las siguientes modificaciones:

- Un contador hardware (Counter) de 2 bits que se incrementa en uno cada vez que una lectura ha sido realizada. Ya que trabaja con 2 bits, puede contar hasta 3.
- Una lógica central compuesta por puertas NAND, OR y AND que controlan las lecturas de memoria. Se habilita cuando el contador es menor que 4, cuando el módulo MPMC Control Logic 2 se ejecuta por primera vez (la señal E estará activa por sólo un ciclo de reloj), y el módulo MPMC Controller ha terminado de leer (el módulo MPMC Control Logic 2 no debería estar funcionando mientras este módulo está en ejecución, es decir, mientras se está realizando la lectura). Una vez que el contador llega a 4 (entrada 11 para la puerta NAND), esta lógica central evita que el módulo MPMC Control Logic 2 continúe funcionando y la ejecución finaliza).
- Un decodificador (DEC) de 2 bits. Sólo una de sus señales de salida estará activa de acuerdo al contador.
- Registros R0... R7 para almacenar los 8 valores leídos de memoria.

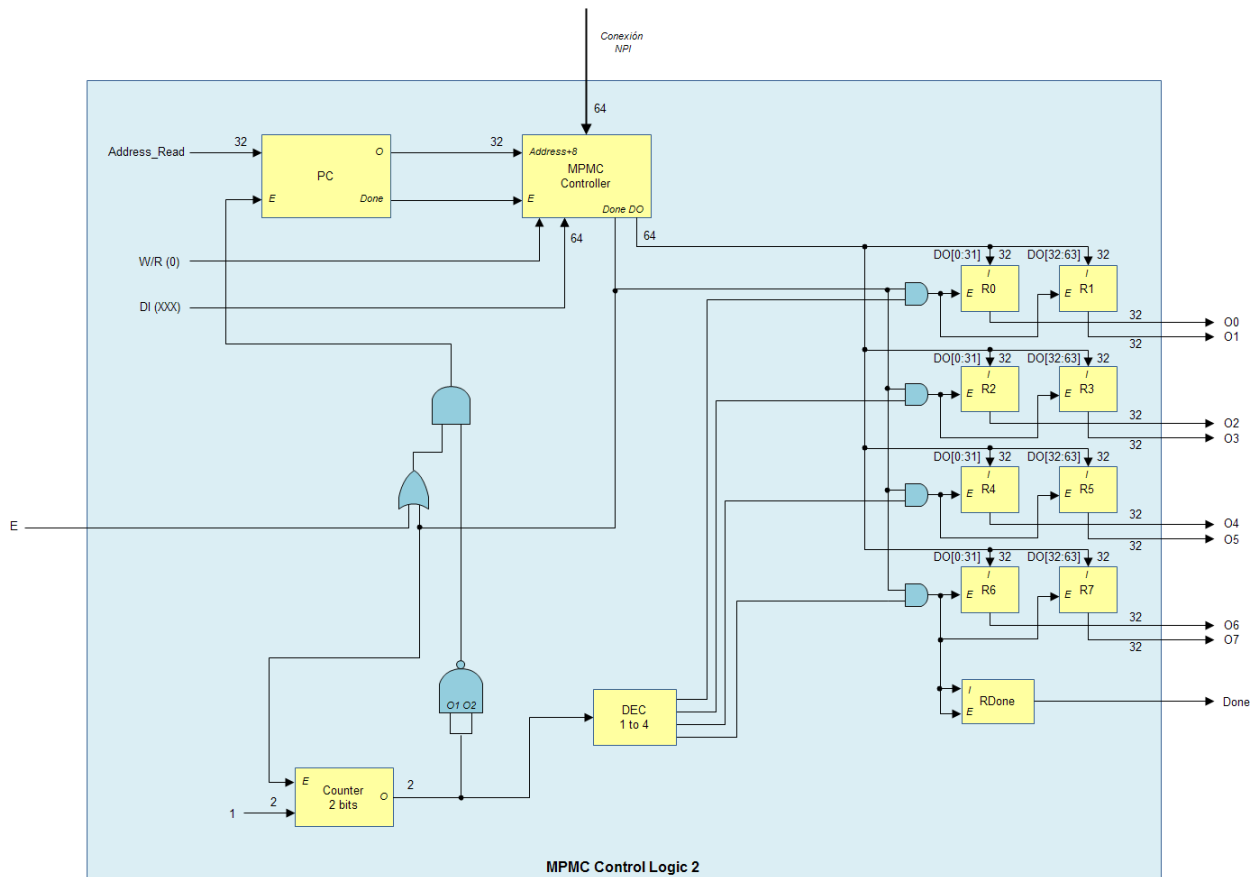


Figura A.3: Diagrama del módulo MPMC Control Logic 2, responsable de obtener de memoria los 8 valores de espectro a través de cada puerto MPMC.

El módulo MPMC Control Logic 3.

El módulo MPMC Control Logic 3 (Figura A.4) realiza la lectura de 8 valores de espectro de memoria, y la escritura del resultado de la distancia Euclídea multi-espectral. Ya que la conexión NPI trabaja con 64 bits, y que en cada lectura se obtienen dos valores de espectro al mismo tiempo,

será necesario realizar un total de 4 lecturas. El resultado del cálculo será almacenado, igualmente, en un formato de 64 bits. Este módulo tiene el mismo diseño que el módulo MPMC Control Logic 2 (Figura A.3), pero con las siguientes modificaciones:

- Dos multiplexores (MUX) de 32 bits que seleccionan entre la dirección de memoria proporcionado por el PC, para el caso de lectura ($W/R=1$), o una dirección de memoria predefinida en la cual escribir el resultado, para el caso de escritura ($W/R=0$).
- Un registro WDone que almacena un 1 cuando la escritura del resultado haya sido realizada. De esta manera, se comunica al exterior del módulo MPMC Control Logic 3 que el resultado ya está disponible en memoria.

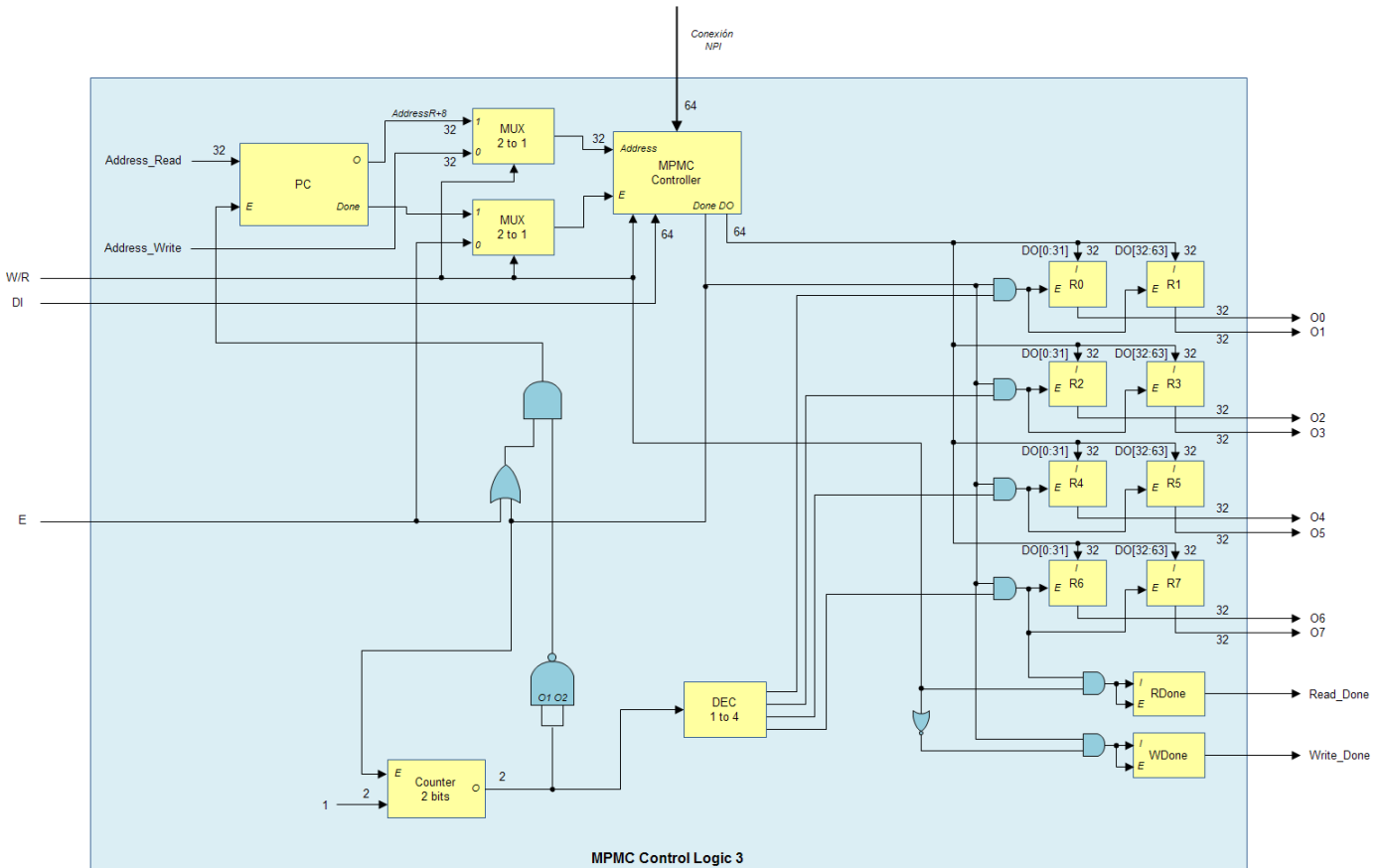


Figura A.4: Diagrama del módulo MPMC Control Logic 3, responsable de obtener de memoria los 8 valores de espectro a través de cada puerto MPMC, y de escribir el resultado del cálculo de la distancia Euclídea multi-espectral.

El módulo Parallel MEDiC.

El módulo Parallel MEDiC (Figura A.5) realiza el cálculo de la distancia Euclídea multi-espectral de dos espectros A y B. Posee el mismo diseño que el circuito original (Figura 6.4(b)), pero optimizado debido a la siguiente modificación:

- El módulo SQRT ha sido mejorado y tiene una latencia de 10 cr en vez de 17 cr.

El número de ciclos de reloj requeridos por este módulo optimizado para realizar el cálculo de la distancia Euclídea multi-espectral de manera paralela ($P_MEDiC_opt_{cc}$), para dos espectros de n bandas, es definido por:

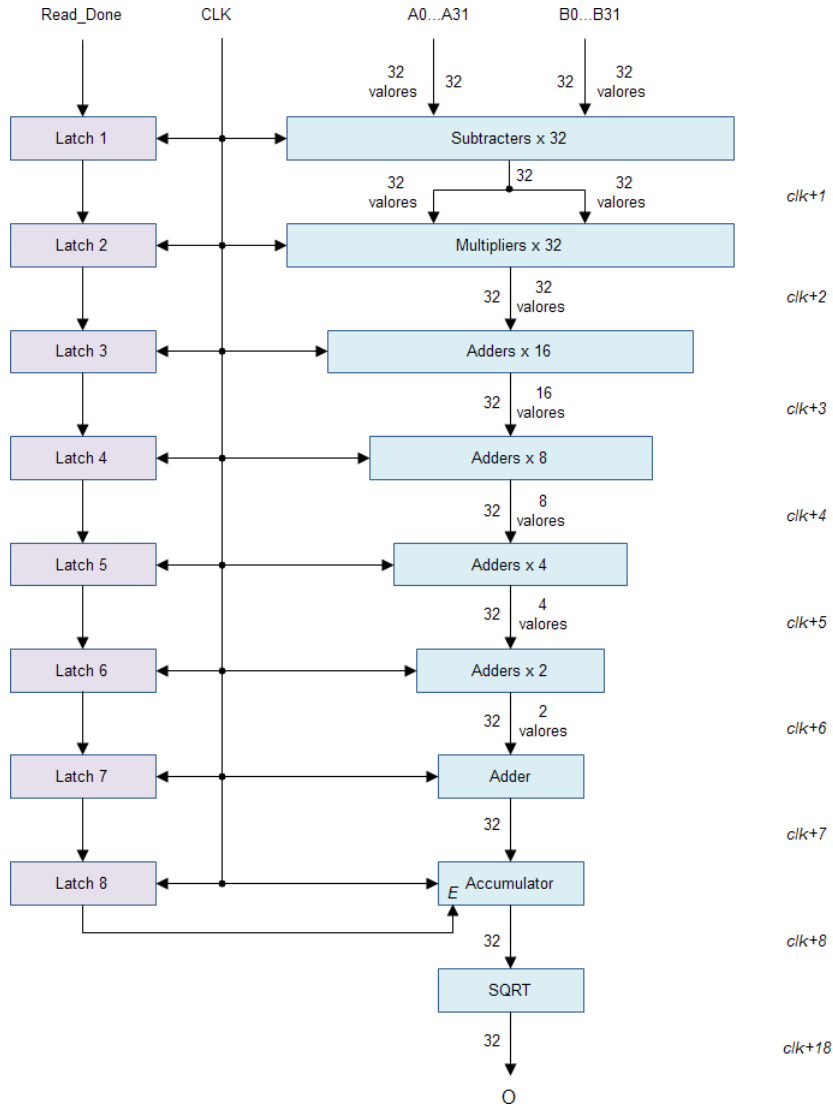


Figura A.5: Circuito paralelo optimizado para el cálculo de la distancia Euclídea multi-espectral.

$$\begin{aligned}
 P_MEDiC_opt_{cc} &= (SUBS_{cc} + MULS_{cc} + ADDS_{cc} + ACC_{cc}) + \left(\left\lceil \frac{n}{32} \right\rceil - 1 \right) + SQRT_{cc} = \\
 &= 8 + \left\lceil \frac{n}{32} \right\rceil - 1 + 10 = \left\lceil \frac{n}{32} \right\rceil + 17
 \end{aligned} \tag{A.1}$$

Donde $SUBS_{cc}$, $MULS_{cc}$, $ADDS_{cc}$, ACC_{cc} y $SQRT_{cc}$ son los ciclos de reloj requeridos por los submódulos restadores, multiplicadores, sumadores, acumulador y SQRT, respectivamente.

A.1.2. Segundo diseño de la arquitectura hardware.

A continuación se definen los módulos implementados para el desarrollo de la segunda arquitectura de comunicaciones hardware, el cual hace uso de la memoria DDR2 como dispositivo intermediario para la optimización de las comunicaciones internas entre los periféricos y el procesador PowerPC. Para ello, se dedican cuatro puertos del componente MPMC (a través de conexiones NPI con el modo Burst Read activo) para el periférico MEDiC, y uno para el procesador.

El módulo MPMC Control Logic.

El módulo MPMC Control Logic (Figura A.6) realiza la lectura de 16 valores de espectro de un determinado puerto de memoria con el modo Burst Read habilitado. Ya que la conexión NPI trabaja con 64 bits, y que en cada lectura se obtienen dos valores de espectro al mismo tiempo, será necesario realizar un total de 8 lecturas. Este módulo tiene el mismo diseño que el módulo MPMC Control Logic 1 (Figura A.1), pero con las siguientes modificaciones:

- Un contador hardware (Counter) de 3 bits que se incrementa en uno cada vez que una lectura ha sido realizada. Ya que trabaja con 3 bits, puede contar hasta 8.
- Una lógica central compuesta por puertas NAND, OR y AND que controlan las lecturas de memoria. Se habilita cuando el contador es menor que 9, cuando el módulo MPMC Control Logic se ejecuta por primera vez (la señal E estará activa por sólo un ciclo de reloj), y el módulo MPMC Controller ha terminado de leer (el módulo MPMC Control Logic no debería estar funcionando mientras este módulo está en ejecución, es decir, mientras se está realizando la lectura). Una vez que el contador llega a 7 (entrada 111 para la puerta NAND), esta lógica central evita que el módulo MPMC Control Logic continúe funcionando y la ejecución finaliza).
- Un decodificador (DEC) de 3 bits. Sólo una de sus señales de salida estará activa de acuerdo al contador.
- El número de registros ha sido incrementado a 16 (R0... R15), para almacenar los 16 valores de espectro leídos de memoria.

El módulo MPMC Controller.

El módulo MPMC Controller ha sido modificado respecto a su diseño original (Figura A.2(b)), ya que la lecturas de memoria con el modo Burst Read habilitado ha sido implementado como máquina de estados (Figura A.7). Para el caso de la escritura, la señales se gestionan del mismo modo que en el diseño original.

A.2. Simulación de un sistema SpaceCube para procesamiento a bordo.

Este apartado muestra de manera detallada los módulos que conforman las metodologías implementadas en cada una de las tarjetas FPGA, utilizadas para la emulación del sistema SpaceCube. La Sección A.2.1 presenta los módulos correspondientes a la metodología desarrollada en la primera tarjeta FPGA: el suavizado espectral Savitzky-Golay (S-C); mientras que la Sección A.2.2 presenta los módulos correspondientes a la metodología desarrollada en la segunda tarjeta FPGA: cálculos estadísticos e índices de vegetación.

A.2.1. Metodología desarrollada en la primera tarjeta FPGA.

A continuación se definen los módulos correspondientes a la metodología implementada en la primera tarjeta FPGA, el suavizado espectral Savitzky-Golay (S-C).

El módulo ToSingle256.

Este módulo se encarga de convertir los valores de espectro recibidos del exterior a formato *Single*, para así poder conseguir una mayor precisión en los cálculos necesarios al realizar el suavizado. Para

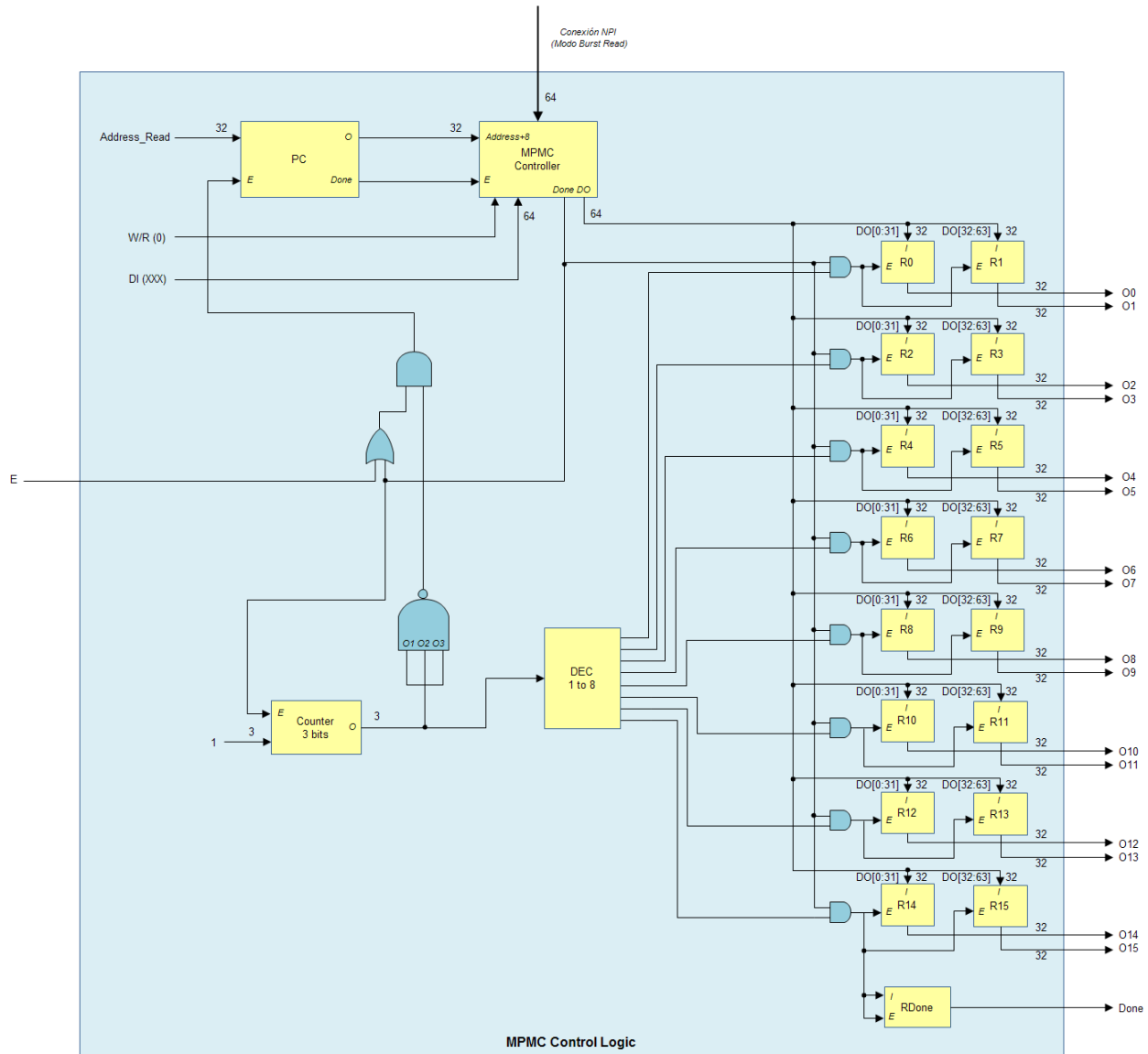


Figura A.6: Diagrama del módulo MPMC Control Logic, responsable de obtener de memoria los 16 valores de espectro a través de cada puerto MPMC y usando el modo Burst Read.

ello hará uso del componente Floating-Point Operator proporcionado por Xilinx (en su opción de conversión de binario natural a *Single*) [Xil12c] (Figura A.8). Como el sistema es capaz de procesar hasta 256 valores de espectro, se dispondrá de hasta 256 componentes que realizarán la conversión en paralelo.

El módulo Filter.

El módulo Filter realiza un filtrado general a los valores de espectro de entrada. Para ello hará uso de una ventana o *kernel* de tamaño 1×3 ($k = [k_1, k_2, k_3]$), de tal manera que cada valor de espectro A_i quedará determinado por $A_i = A_{i-1} \times k_1 + A_i \times k_2 + A_{i+1} \times k_3$. El tipo de filtrado a realizar quedará determinado por los valores de k , los cuales son introducidos al módulo mediante las señales *Coef1* (k_1), *Coef2* (k_2) y *Coef3* (k_3). De esta forma, si se desea realizar un filtrado de paso bajo, basta con asignar las señales como $Coef1 = 1/3$, $Coef2 = 1/3$ y $Coef3 = 1/3$; mientras que si se desea realizar un filtrado de paso alto, dichas señales serán asignadas como $Coef1 = -1/3$, $Coef2 = 2/3$ y $Coef3 = -1/3$. Por supuesto, este módulo es de propósito general, por lo que cambiando los valores de dichas señales correspondientes al *kernel* sería posible realizar filtrados de cualquier otro tipo.

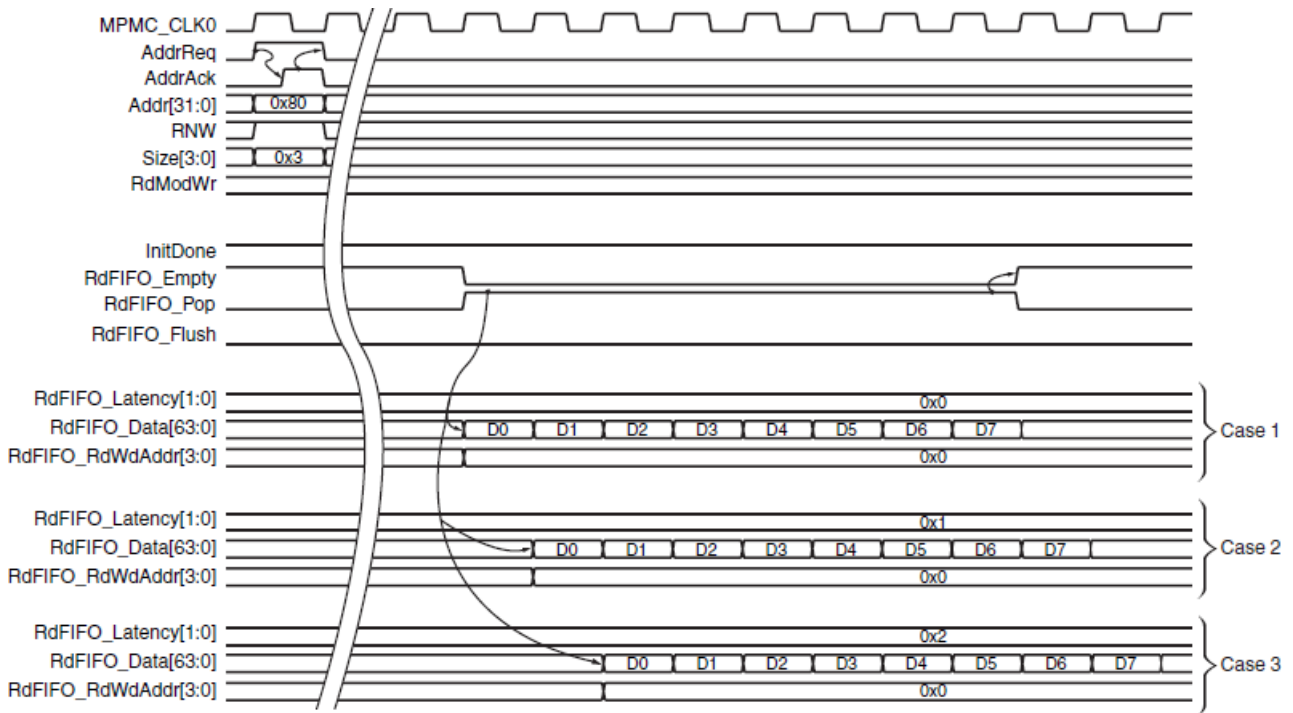


Figura A.7: Señales de control del componente MPMC para realizar la lectura de un bloque de palabras de 64 bits de memoria, usando el modo Burst Read.

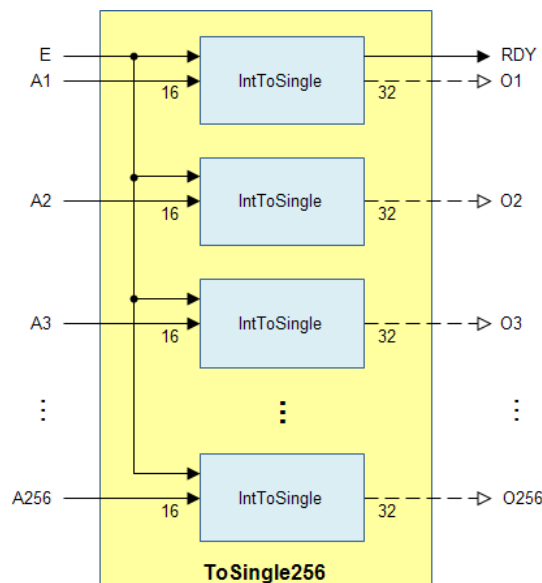


Figura A.8: Diagrama del módulo ToSingle256.

Para realizar correctamente la operación de filtrado, se hará uso de los siguientes módulos (Figura A.9), de tal manera que:

- El módulo Filter funciona mediante una serie de iteraciones, un total de $NElmts_1 - 2$, donde $NElmts_1$ es el número de elementos que componen la serie de datos a filtrar.
- En cada iteración i se determinan los valores $[A_{i-1}, A_i, A_{i+1}]$ a los cuales se les aplicará el *kernel*. La selección de estos valores se realizarán mediante los tres multiplexores en la parte superior del diagrama.
- Los valores $[A_{i-1}, A_i, A_{i+1}]$ correspondientes a la iteración i y proporcionados por los multiplexores son transferidos al módulo Kernel, el cual realiza la operación de filtrado

correspondiente utilizando las señales $Coef1$, $Coef2$ y $Coef3$.

- El resultado del filtrado proporcionado por Kernel es transferido al demultiplexor, el cual determina dicho resultado como salida O_i para la iteración i .
- Los contadores centrales permitirán llevar un recuento de las iteraciones realizadas, de tal manera que para la iteración i , la salida del contador de la izquierda será $i - 1$, para el central i y para el de la derecha $i + 1$. Dichas salidas establecerán los valores de espectro a considerar para la iteración i , comunicándose a los tres multiplexores y al demultiplexor.
- Los componentes restantes compuestos por los latches, las puertas lógicas y el comparador permiten la correcta sincronización del circuito, permitiendo que los contadores incrementen en 1 su salida cuando el filtrado de la iteración i haya sido realizado, y terminando la ejecución cuando no queden más valores de espectro a procesar. Esto se determinará cuando el contador de la derecha haya alcanzado el valor máximo de iteración ($i + 1 = NElmts_1$), enviando una señal RDY al exterior.
- Los espectros $A1$ y $A256$, al estar situados en los extremos de la serie de datos, no pueden ser considerados por el *kernel*, por lo que no serán procesados y salen del módulo con el mismo valor con el que entraron.

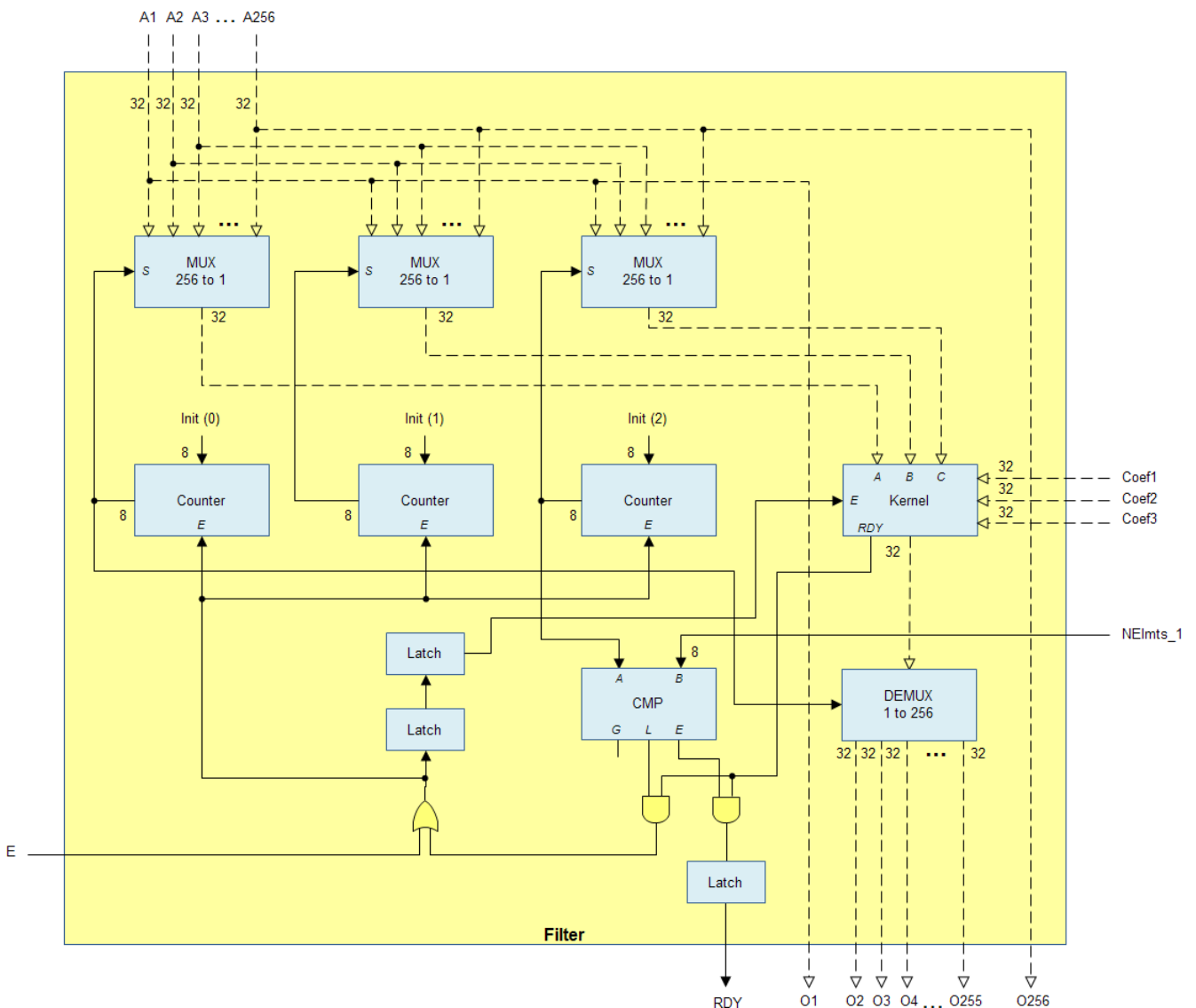


Figura A.9: Diagrama del módulo *Filter*.

El módulo Kernel.

El módulo Kernel aplica el un *kernel* correspondiente de un determinado filtrado a los valores de entrada. Es decir, a partir de los valores A , B y C , y de los coeficientes del *kernel* $k = [Coef1, Coef2, Coef3]$, realiza la operación $A \times Coef1 + B \times Coef2 + C \times Coef3$.

Para realizar correctamente esta operación, se hará uso de los siguientes módulos (Figura A.10), de tal manera que:

- Los tres multiplicadores superiores funcionan en paralelo y realizan las multiplicaciones $A \times Coef1$, $B \times Coef2$ y $C \times Coef3$.
- Los dos sumadores calculan el sumatorio de las multiplicaciones anteriores, proporcionando de esta forma la operación $A \times Coef1 + B \times Coef2 + C \times Coef3$. La salida del último sumador corresponderá al resultado de aplicar el *kernel*, notificándolo al exterior mediante la activación de la señal *RDY*.

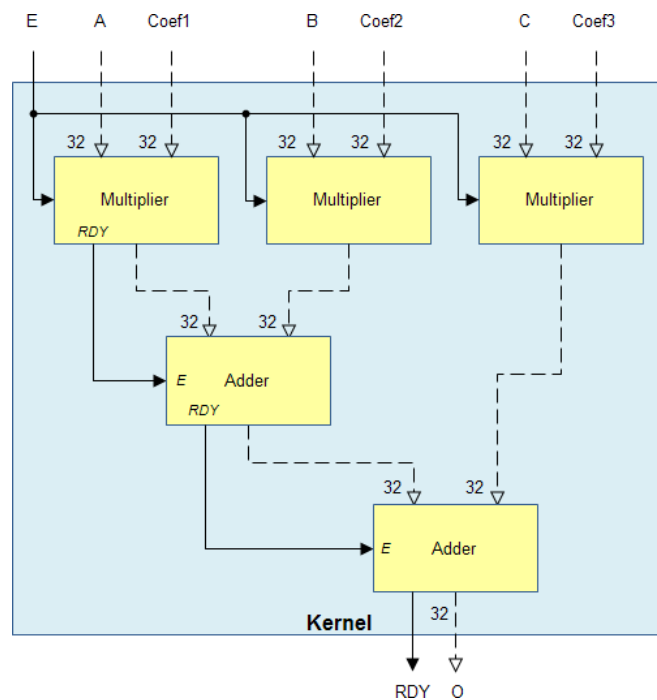


Figura A.10: Diagrama del módulo Kernel.

El módulo Mean.

El módulo Mean calcula la media de hasta un total de 256 valores de espectro de entrada, de acuerdo a la Ecuación 6.8.

Para realizar correctamente esta operación, se hará uso de los siguientes módulos (Figura A.11), de tal manera que:

- El módulo Mean funciona principalmente en dos fases diferentes. Una primera fase en la cual se realiza el cálculo del sumatorio de los valores de entrada ($\sum_{i=1}^{256} A_i$), y otra segunda fase en la cual se realiza la división del sumatorio entre el número de elementos (representado por la señal $NElmts_D$), obteniendo de esta manera la media de todos los valores ($(\sum_{i=1}^{256} A_i)/NElmts$).

- La primera fase se realiza de manera iterativa. En cada iteración i se determina el valor A_i que será considerado para el sumatorio. La selección de dicho valor se realizará mediante el multiplexor en la parte superior del diagrama y será determinado por el contador, el cual lleva un recuento del número de iteraciones. A medida que todos los valores van siendo proporcionados por el multiplexor, el sumador va calculando las sumas acumuladas, las cuales se van almacenando y acumulando en el registro R . Los componentes compuestos por los latches, las puertas lógicas y el comparador permiten la correcta sincronización de esta primera fase, permitiendo que el contador incremente en 1 su salida cuando el sumatorio de la iteración i haya sido realizado, y terminando la ejecución cuando no queden más valores de espectro a procesar. Esto se determinará cuando se haya alcanzado el valor máximo de iteración ($i = NElmts_1$), comenzando la segunda fase.
- La segunda fase comenzará cuando las iteraciones correspondientes al cálculo del sumatorio de los valores haya finalizado. Al ocurrir esto, el comparador detectará que se ha alcanzado el número de iteraciones máximo y activará el divisor. Éste recibirá el resultado del sumatorio y el número de elementos de la serie de datos (representado por la señal $NElmts_D$), y realizará la división. Al terminar, el resultado representará la media de los valores recibidos, y se indicará al exterior mediante la señal RDY .

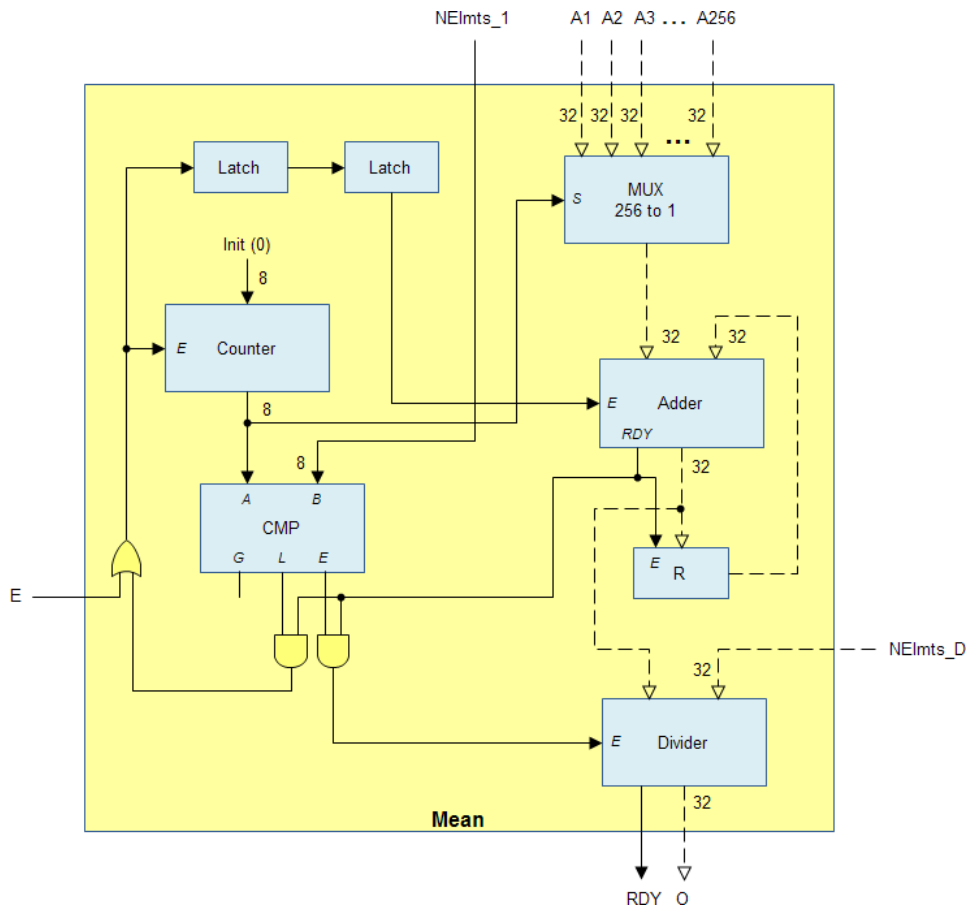


Figura A.11: Diagrama del módulo Mean.

El módulo Mean, aunque permite realizar la media de hasta 256 elementos, en la primera tarjeta FPGA será utilizado para calcular la media del resultado del módulo HPF Filter, compuesto por los valores impares de espectro tras aplicarles un filtrado de paso alto. Por tanto, en este caso, se calculará la media de un máximo de 128 elementos.

El módulo StdDev.

El módulo StdDev calcula la desviación estándar de hasta un total de 256 valores de espectro de entrada, de acuerdo a la Ecuación 6.9.

Para realizar correctamente esta operación, se hará uso de los siguientes módulos (Figura A.12), de tal manera que:

- El módulo StdDev funciona principalmente en tres fases diferentes. Una primera fase en la cual se realiza el sumatorio de los cuadrados de las diferencias de los valores respecto a su media, siendo ésta representada por la señal *Mean* ($\sum_{i=1}^{256} (A_i - \bar{A})^2$); otra segunda fase en la cual se realiza la división de este cálculo entre el número de elementos menos 1, siendo este último valor representado por la señal *NElmts_D_1* ($(\sum_{i=1}^{256} (A_i - \bar{A})^2) / (NElmts - 1)$); y otra tercera en la cual se realiza la raíz cuadrada de todo el cálculo obteniendo, de esta forma, la desviación estándar de todos los valores ($\text{sqrt}((\sum_{i=1}^{256} (A_i - \bar{A})^2) / (NElmts - 1))$).
- La primera fase se realiza de manera iterativa. En cada iteración i se determina el valor A_i que será considerado para el sumatorio. La selección de dicho valor se realizará mediante el multiplexor en la parte superior del diagrama y será determinado por el contador, el cual lleva un recuento del número de iteraciones. A medida que todos los valores del espectro van siendo proporcionados por el multiplexor, se realiza la diferencia respecto a la media (restador), se eleva al cuadrado (multiplicador) y se calcula la suma acumulada. Esta suma acumulada se va almacenando en el registro R . Los componentes compuestos por los latches, las puertas lógicas y el comparador permiten la correcta sincronización de esta primera fase, permitiendo que el contador incremente en 1 su salida cuando el sumatorio de la iteración i haya sido realizado, y terminando la ejecución cuando no queden más valores de espectro a procesar. Esto se determinará cuando se haya alcanzado el valor máximo de iteración ($i = NElmts_1$), comenzando la segunda fase.
- La segunda fase comenzará cuando las iteraciones correspondientes al cálculo del sumatorio de los cuadrados de los valores respecto a la media haya finalizado. Al ocurrir esto, el comparador detectará que se ha alcanzado el número de iteraciones máximo y activará el divisor. Éste recibirá el resultado del sumatorio y el número de elementos de la serie de datos menos 1 (representado por la señal *NElmts_D_1*), y realizará la división.
- La tercera fase comenzará cuando la división anterior haya finalizado. El componente SQRT recibirá el resultado anterior y realizará la raíz cuadrada. Al terminar, el resultado representará la desviación estándar de los valores recibidos, y se indicará al exterior mediante la señal *RDY*.

El módulo StdDev, aunque permite realizar la desviación estándar de hasta 256 elementos, en la primera tarjeta FPGA será utilizado para calcular la desviación estándar del resultado del módulo HPF Filter, compuesto por los valores impares de espectro tras aplicarles un filtrado de paso alto. Por tanto, en este caso, se calculará la desviación estándar de un máximo de 128 elementos.

El módulo Threshold128.

El módulo Threshold realiza el umbralizado de una serie de datos. A partir de un valor t referencia, pone a 0 aquéllos valores de entrada que sean menores a $-t$ o bien superiores a t . Es decir, para un valor de entrada A_i , su salida será exactamente la misma si se cumple que $-t \leq A_i \leq t$, y 0 en caso contrario. En este módulo, dicha operación puede realizarse hasta a un total de 128 valores de manera paralela.

Para ello, se hará uso del módulo Threshold, que realiza el umbralizado de manera individual a cada componente (Figura A.13). Como es necesario aplicar esta operación a 128 valores diferentes, se necesitarán un total de 128 módulos Threshold que funcionarán en paralelo. Aparte cada valor de entrada, cada uno de estos módulos necesitarán los valores de umbralizado t (representado por

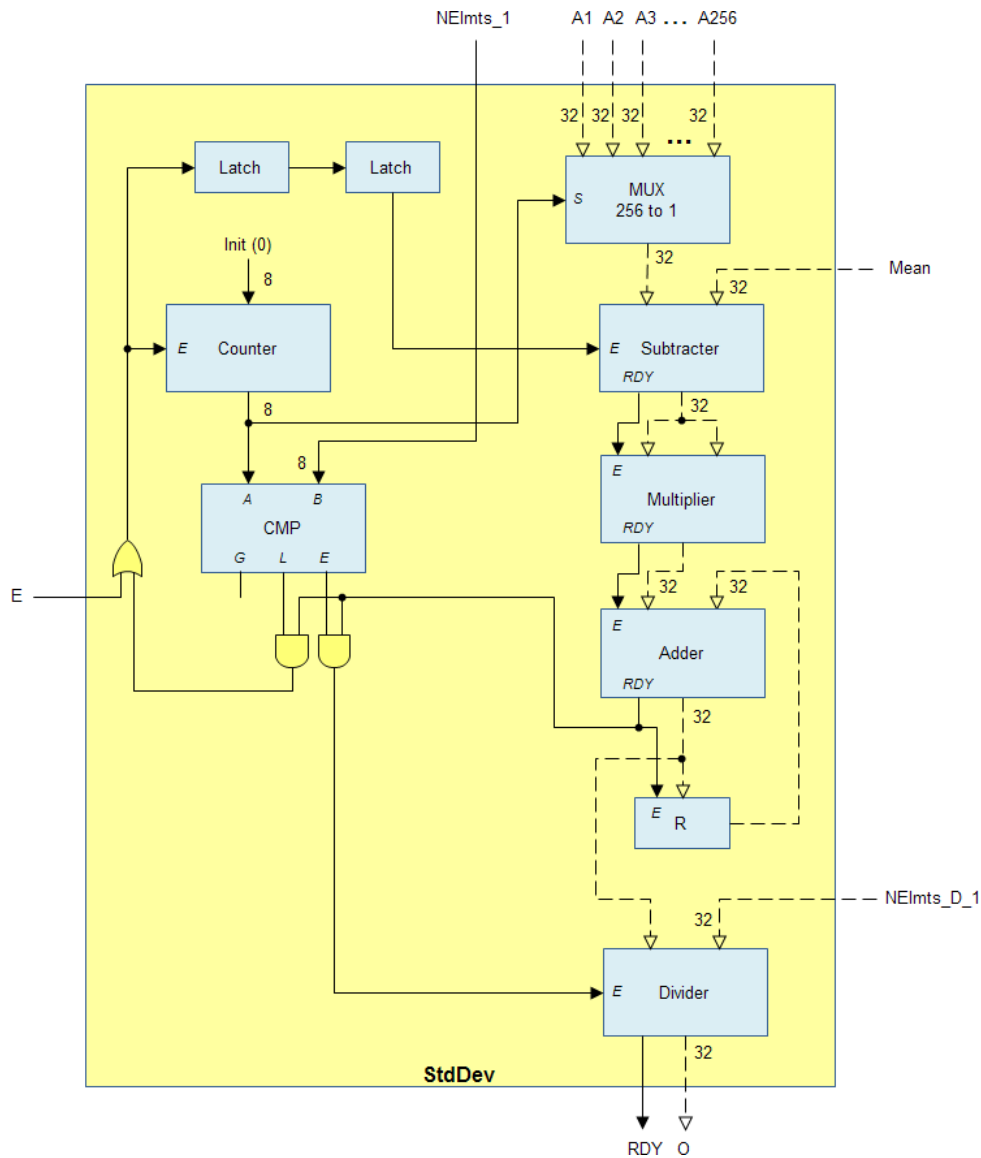


Figura A.12: Diagrama del módulo *StdDev*.

la señal *Threshold*) y $-t$ (representado por la señal $_Threshold$). Como todos los módulos *Threshold* están sincronizados y finalizan en el mismo instante, se usará la señal *RDY* del primero para indicar que la ejecución ha finalizado.

El módulo *Threshold*.

Este módulo realiza el umbralizado a un componente A de manera independiente, de tal manera que éste mantendrá su valor original si se cumple que $-t \leq A \leq t$ (siendo t el valor umbral), o le será asignado un 0 en caso contrario.

La manera de realizar dicha operación es mediante la utilización de dos comparadores (Figura A.14) que comprobarán si $-t \leq A$ (comparador derecho) y si $A \leq t$ (comparador izquierdo). Ambos valores t y $-t$ estarán representadas por las señales *Threshold* y $_Threshold$ respectivamente. En caso de que $A < -t$ o $A > t$, los comparadores proporcionarán al multiplexor central un valor de 1 (a través de la puerta OR), indicando una salida 0 al exterior, al no cumplirse la condición. En caso contrario (se cumple la condición), la selectora del multiplexor recibirá un 0, por lo que se proporcionará el valor original al exterior.

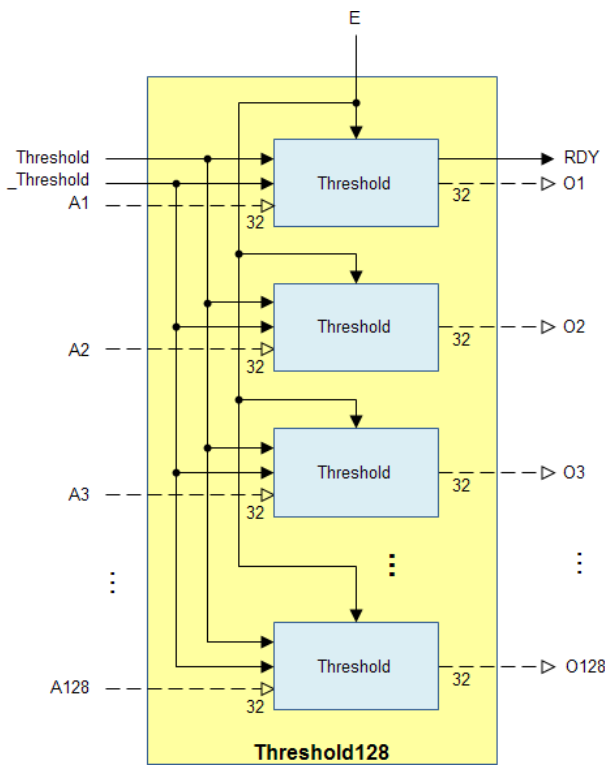


Figura A.13: Diagrama del módulo *Threshold128*.

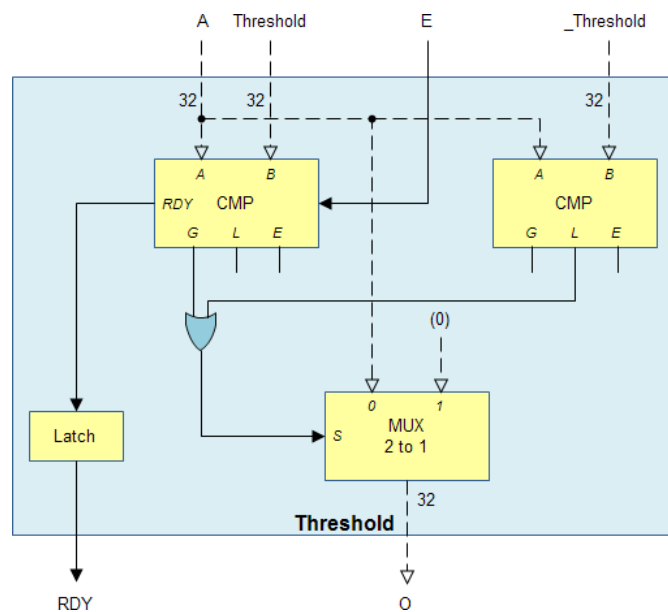


Figura A.14: Diagrama del módulo *Threshold*.

Ya que el multiplexor no dispone de señal *RDY*, el latch permitirá retrasar la señal *RDY* del comparador un ciclo de reloj (tiempo de ejecución del multiplexor), indicando de esta manera que la ejecución ha finalizado.

El módulo *FilterAdder*.

Este módulo permite realizar la suma de los valores de espectro procesados mediante un filtro paso bajo A_i , y aquéllos procesados mediante un filtro paso alto y posterior umbralizado UT B_i . La manera de realizar la suma es componente a componente, tal que $O_i = A_i + B_i$.

Para realizar correctamente esta operación, se hará uso de los siguientes módulos (Figura A.15), de tal manera que:

- El módulo FilterAdder funciona mediante una serie de iteraciones, un total de $NElmts$, donde $NElmts$ es el número de elementos que componen la serie de datos a filtrar.
- En cada iteración i se determinan los valores A_i y B_i a los cuales se les aplicará la suma. La selección de estos valores se realizarán mediante los dos multiplexores en la parte superior del diagrama.
- Los valores A_i y B_i correspondientes a la iteración i y proporcionados por los multiplexores son transferidos al sumador.
- El resultado de la suma de ambos valores es transferido al demultiplexor, el cual lo determina como salida O_i para la iteración i .
- El contador central permitirá llevar un recuento de las iteraciones realizadas, de tal manera que para la iteración i , su salida será i . Dicha salida establecerá los valores de entrada a considerar para la iteración i , comunicándose al multiplexor y al demultiplexor.
- Los componentes restantes compuestos por los latches, las puertas lógicas y el comparador permiten la correcta sincronización del circuito, permitiendo que el contador incremente en 1 su salida cuando la suma de la iteración i haya sido realizado, y terminando la ejecución cuando no queden más valores de entrada a procesar. Esto se determinará cuando el contador haya alcanzado el valor máximo de iteración ($i = NElmts_1$), enviando una señal RDY al exterior.

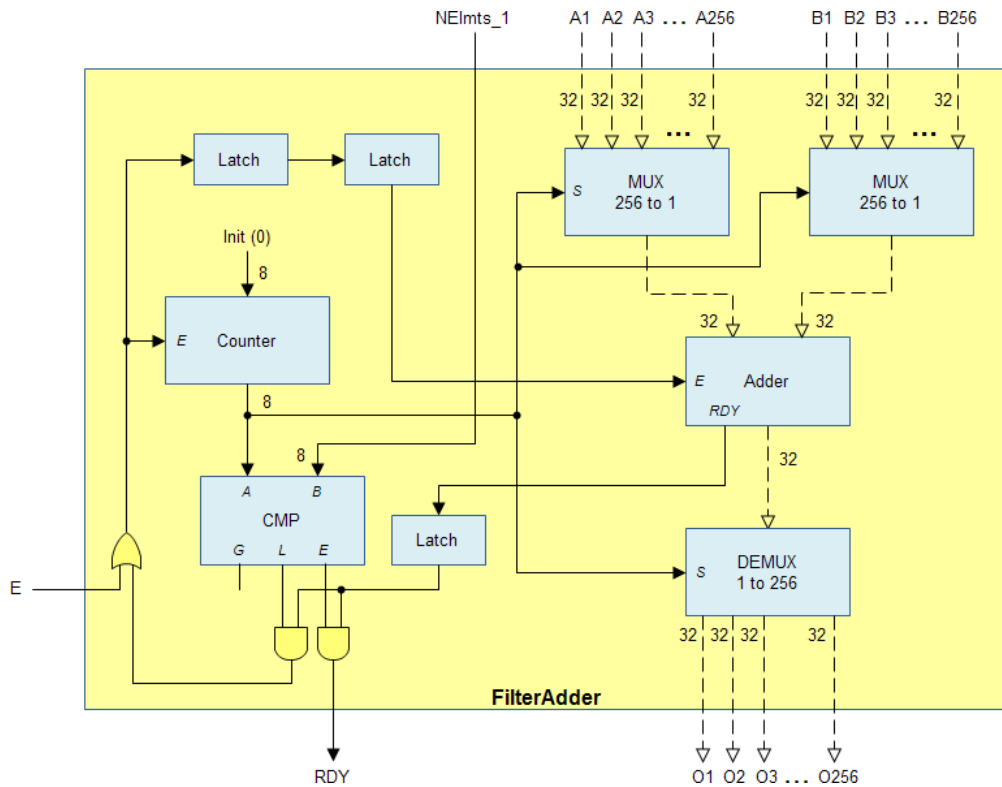


Figura A.15: Diagrama del módulo FilterAdder.

Este módulo se encarga de convertir los valores de espectro suavizados recibidos a formato binario natural y enviarlos al exterior. Para ello hará uso del componente Floating-Point Operator proporcionado por Xilinx (en su opción de conversión de *Single* a binario natural (Figura A.16)). Como el sistema es capaz de procesar hasta 256 valores de espectro, se dispondrá de hasta 256 componentes que realizarán la conversión en paralelo.

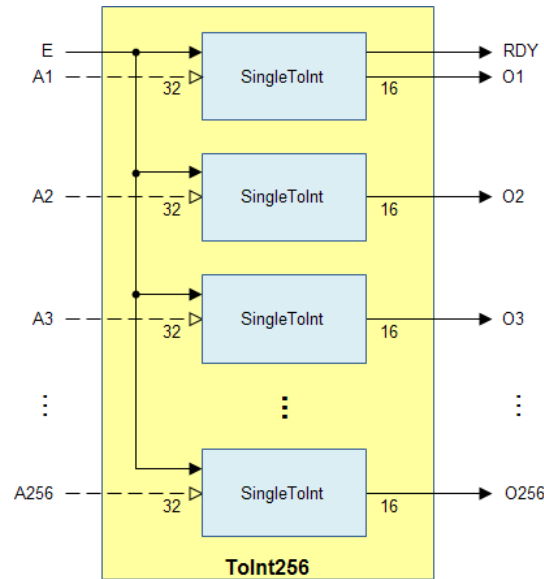


Figura A.16: Diagrama del módulo *ToInt256*.

A.2.2. Metodología desarrollada en la segunda tarjeta FPGA.

A continuación se definen los módulos correspondientes a la metodología implementada en la segunda tarjeta FPGA, cálculos estadísticos e índices de vegetación.

El módulo *ToSingle256*.

Este módulo se encarga de convertir los valores de espectro recibidos del exterior a formato *Single*, para así poder conseguir una mayor precisión en los cálculos necesarios al realizar el suavizado.

Su funcionamiento es exactamente el mismo que el de la primera tarjeta, por lo que su descripción ya está detallada en la Sección A.2.1.

El módulo *Maximum*.

El módulo *Maximum* obtiene el valor máximo de hasta un total de 256 valores de entrada.

Para realizar correctamente esta operación, se hará uso de los siguientes módulos (Figura A.17), de tal manera que:

- El módulo *Maximum* funciona mediante una serie de iteraciones, un total de $NElmts$, donde $NElmts$ es el número de elementos que componen la serie de datos a procesar.
- En cada iteración i se determina el valor A_i para comprobar si supera el máximo obtenido hasta ese instante. La selección de este valor se realizará mediante el multiplexor en la parte superior del diagrama.

- El contador central permitirá llevar un recuento de las iteraciones realizadas, de tal manera que para la iteración i , la salida del mismo será i . Dicha salida establecerá el valor de entrada a considerar para la iteración i , comunicándose al multiplexor.
- El valor A_i correspondiente a la iteración i y proporcionado por el multiplexor es transferido al comparador, el cual comprueba si dicho valor es mayor al acumulado en el registro R . En caso de que este valor sea mayor, será almacenado en dicho registro para comparaciones en futuras iteraciones.
- Los componentes restantes compuestos por los latches, las puertas lógicas y el otro comparador permiten la correcta sincronización del circuito, permitiendo que el contador incremente en 1 su salida cuando la comparación con el máximo haya sido realizada, y terminando la ejecución cuando no queden más valores de espectro a procesar. Esto se determinará cuando el contador haya alcanzado el valor máximo de iteración ($i = NElmts_1$), enviando una señal RDY al exterior. En este mismo instante, el valor almacenado en el registro R corresponderá al máximo de la serie de datos de entrada, por lo que también es transferido al exterior.

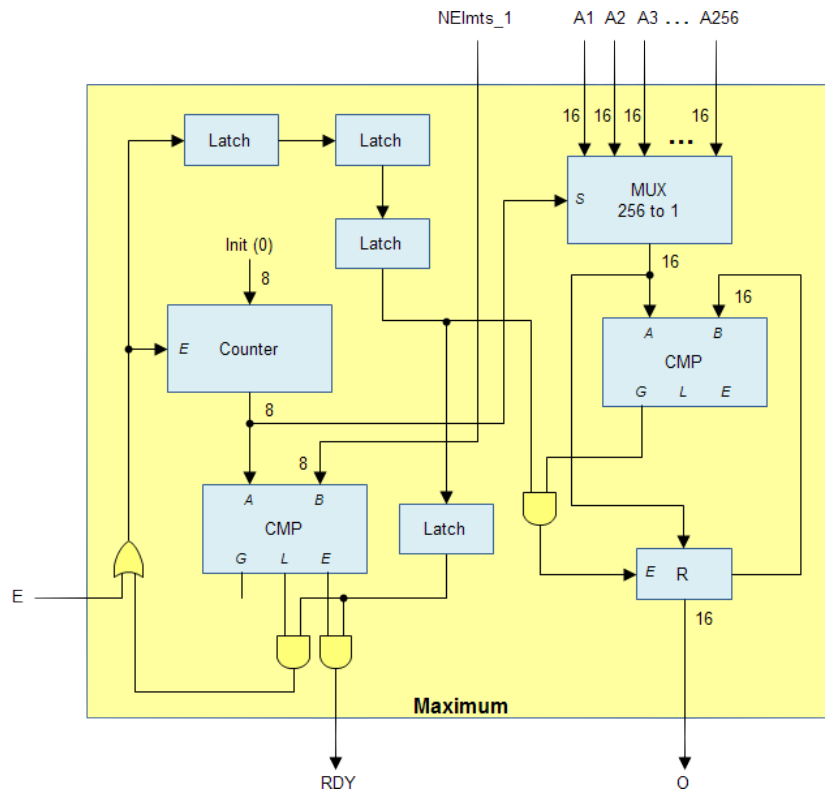


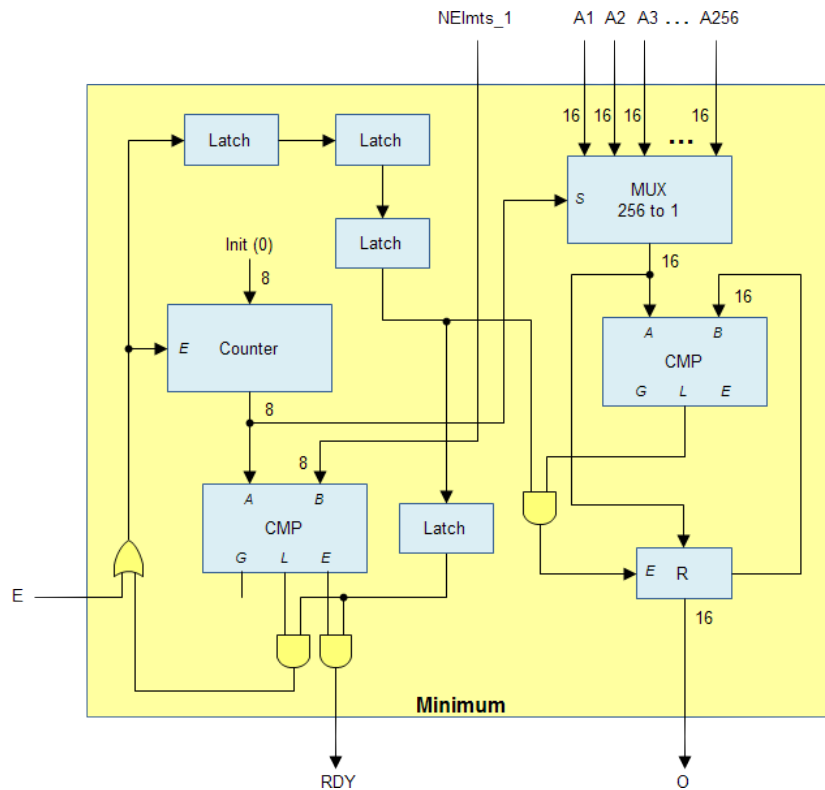
Figura A.17: Diagrama del módulo Maximum.

El módulo Minimum.

El módulo Minimum obtiene el valor mínimo de hasta un total de 256 valores de entrada.

El funcionamiento de este módulo es exactamente igual al del módulo Maximum, y está compuesto por los mismos componentes (Figura A.18). El único cambio reside en el comparador, el cual acumulará el mínimo de los valores de entrada en el registro R en vez del máximo.

El módulo Mean.

Figura A.18: Diagrama del módulo *Minimum*.

El módulo *Mean* realiza la media de hasta un total de 256 valores de espectro de entrada, de acuerdo a la Ecuación 6.8.

Su funcionamiento es exactamente el mismo que el de la primera tarjeta, por lo que su descripción ya está detallada en la Sección A.2.1.

El módulo *StdDev*.

El módulo *StdDev* calcula la desviación estándar de hasta un total de 256 valores de espectro de entrada, de acuerdo a la Ecuación 6.9.

Su funcionamiento es exactamente el mismo que el de la primera tarjeta, por lo que su descripción ya está detallada en la Sección A.2.1.

El módulo *IR_Ratio*.

Este módulo calcula el ratio *IR Ratio* a partir de los valores de espectro con frecuencias infrarrojo cercano y rojo, de acuerdo a la Ecuación 6.11.

Como el cálculo de dicho ratio solo implica una división, este módulo estará compuesto por un único divisor (Figura A.19) que realizará la división del valor correspondiente a las frecuencias del infrarrojo cercano (representado por la señal *IR*) entre el correspondiente a las frecuencias del rojo (representado por la señal *R*). Cuando el divisor haya terminado de realizar la operación, la señal *RDY* se activa para indicar al exterior que el resultado ya está disponible.

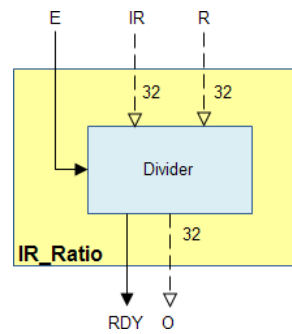


Figura A.19: Diagrama del módulo *IR_Ratio*.

El módulo NDVI.

El módulo calcula el índice NDVI a partir de los valores de espectro con frecuencias infrarrojo cercano y rojo, de acuerdo a la Ecuación 6.11.

Para realizar correctamente esta operación, se hará uso de los siguientes módulos (Figura A.20), de tal manera que:

- El restador y el sumador permiten realizar las operaciones $IR - R$ y $IR + R$, respectivamente, donde el valor correspondiente a las frecuencias de infrarrojo cercano estará representado por la señal IR , y el correspondiente a las frecuencias de rojo por la señal R . Ambos módulos funcionan en paralelo y emplean el mismo tiempo de ejecución, por lo que se utilizará únicamente la señal RDY del restador para indicar que las operaciones ya han terminado al divisor.
- El divisor recibe los resultados de las operaciones $IR - R$ y $IR + R$ y empezará a funcionar en cuanto la señal RDY del restador esté activa. Proporciona la división de ambos elementos calculando de esta manera el índice NDVI $((IR - R)/(I + R))$. Una vez el resultado está disponible, envía una señal RDY al exterior.

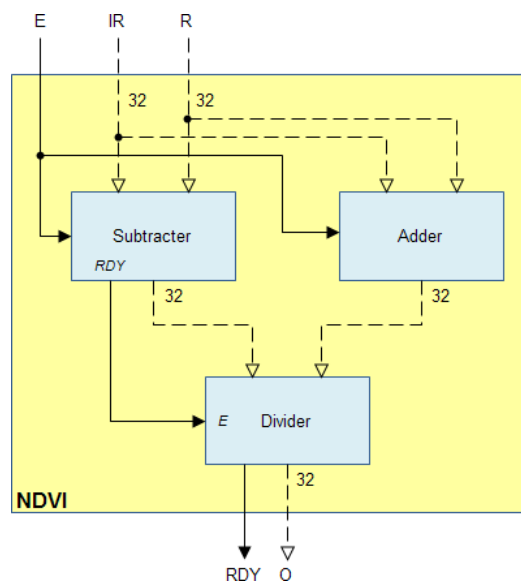


Figura A.20: Diagrama del módulo *NDVI*.

Créditos de las imágenes

- Figura 3.1(a): ACME Worldwide Enterprises Inc. / Analytical Imaging and Geophysics LLC.
- Figura 3.1(c): Satellite Imaging Corporation (SIC).
- Figura 3.1(a): National Aeronautics and Space Administration (NASA).
- Figura 3.1(e): National Aeronautics and Space Administration (NASA).
- Figura 3.2: Edwin Díaz Caballero.
- Figura 3.3: Scientific Electronic Library Online (SciELO).
- Figura 3.6: Commonwealth Scientific and Industrial Research Organisation (CSIRO).
- Figura 3.7(a): Asociación GDCCT.
- Figura 3.7(b): White Mountain Research Center (WMRC).
- Figura 3.8: Penn State University.
- Figura 3.9: National Aeronautics and Space Administration (NASA).
- Figura 3.14: Emilio Chuvieco.
- Figura 3.18: National Aeronautics and Space Administration (NASA).
- Figura 3.19: National Aeronautics and Space Administration (NASA).
- Figura 3.20: The CPU Shack Museum.
- Figura 3.21(a): The CPU Shack Museum.
- Figura 3.21(b): CPU Grave Yard.
- Figura 3.22: Atmel Corporation.
- Figura 3.23: Aeroflex Gaisler.
- Figura 3.24: University of Windsor.
- Figura 3.25: University of Windsor.
- Figura 3.26: Muhammad Yasir.
- Figura 3.27: Xilinx.
- Figura 3.28: Peter Warnes Consulting.
- Figura 3.29: National Aeronautics and Space Administration (NASA).
- Figura 3.30: National Aeronautics and Space Administration (NASA).

- Figura 3.31: National Aeronautics and Space Administration (NASA).
- Figura 3.32: National Aeronautics and Space Administration (NASA).
- Figura 3.33: National Aeronautics and Space Administration (NASA).
- Figura 3.34: National Aeronautics and Space Administration (NASA).
- Figura 3.35: Xilinx.
- Figura 3.36: Xilinx.
- Figura 3.35: Xilinx.
- Figura 5.79: European Space Agency (ESA).
- Figura 5.80: European Space Agency (ESA).
- Figura 6.28: Albert Alonso.
- Figura 8.4: University of Maryland.
- Figura A.2: Xilinx.
- Figura A.7: Xilinx.

El resto de figuras han sido diseñados por el autor y por tanto le corresponden todos los derechos.

Bibliografía

- [AHSwww] Página web oficial del instrumento AHS: <http://www.uv.es/leo/sen2flex/ahs.htm> (Último acceso: 28 Julio 2013).
- [AVIwww] Página web oficial del instrumento AVIRIS: <http://aviris.jpl.nasa.gov/> (Último acceso: 28 Julio 2013).
- [Abr77] MJ Abrams, RP Ashley, LC Rowan, AFH Goetz, AB Kahle (1977). "Mapping of hydrothermal alteration in the Cuprite mining district, Nevada, using aircraft scanner images for the spectral region 0.46 to 2.36 micro-m". *Geology* 5(12): 713-718.
- [Aer10] Aeroflex Gaisler AB (2010). "SPARC V8 32-bit processor LEON3 / LEON3-FT CompanionCore data sheet".
- [Aer12] Aeroflex (2012). "UT6325 RadTol Eclipse FPGA".
- [Aer13] Aeroflex Gaisler AB (2013). "LEON3-FT SPARC V8 processor LEON3FT-RTAX data sheet and user's manual".
- [Agg11] R Aggarwal, JK Singh, VK Gupta, S Rathore, M Tiwari, A Khare (2011). "Noise reduction of speech signal using wavelet transform with modified universal threshold". *International Journal of Computer Applications* 20(5): 14-19.
- [Agu07] M Aguirre, B Berruti, JL Bezy, M Drinkwater, F Heliere, U Klein, C Mavrocordatos, P Silvestrin, B Greco, J Benveniste (2007). "Sentinel-3 - The ocean and medium-resolution land mission for GMES operational services". *ESA Bulletin* 131: 24-29.
- [Air09] Air Command, Staff College Space Research Electives Seminars (2009). "AU-18 space primer". *Ed. Air University Press*.
- [Ale02] MD Alexandrov, AA Lacis, BE Carlson (2002). "Remote sensing of atmospheric aerosols and trace gases by means of multifilter rotating shadowband radiometer. Part I: Retrieval algorithm". *Journal of the Atmospheric Sciences* 59: 524-543.
- [Ale07] C Alecu, N Chrysoulakis, S Oancea, G Stancaliea (2007). "The georeferencing errors of satellite data in remote sensing applications". *Image and Signal Processing for Remote Sensing XIII* 6748(10): 1-8.
- [Alt00] Altera Corporation (2000). "Nios soft core embedded processor".
- [Alt08] Altera Corporation (2008). "Introduction to the Altera Nios II soft processor".
- [And93] R Andraka (1993). "A survey of CORDIC algorithms for FPGA based computers". *International Symposium on Field Programmable Gate Arrays*, Monterey, USA.
- [Atm01a] Atmel (2001). "8-bit AVR microcontroller with 8K bytes in-system programmable flash".
- [Atm01b] Atmel (2001). "Atmel AT94K FPSLIC starter kit".

- [Axe99] J Axelson (1999). "Designing RS-485 circuits". *Circuit Cellar Ink* 107: 20-24.
- [Bab10] SS Baboo, MR Devi (2010). "An analysis of different resampling methods in Coimbatore, District". *Global Journal of Computer Science and Technology* 10(15): 61-66.
- [Bai98] C Baillard, O Dissard, O Jamet, H Maître (1998). "Extraction and textural characterization of above-ground areas from aerial stereo pairs: a quality assessment". *ISPRS Journal of Photogrammetry and Remote Sensing* 53: 130-141.
- [Bai99] C Baillard, H Maître (1999). "3-D reconstruction of urban scenes from aerial stereo imagery: a focusing strategy". *Computer Vision and Image Understanding* 76(3): 244-258.
- [Bal09] AM Baldridge, SJ Hook, CI Grove, G Rivera (2009). "The ASTER spectral library version 2.0". *Remote Sensing of Environment* 113(4): 711-715.
- [Ban95] A Bannari, D Morin, GB Bénie, FJ Bonn (1995). "A theoretical review of different mathematical models of geometric corrections applied to remote sensing images". *Remote Sensing Reviews* 13(1-2): 27-47.
- [Bar03] BR Brakstrom, M Folk (2003). "Attributes of file formats for long-term preservation of scientific and engineering data in digital libraries". *Proceedings of 3rd ACM/IEEE-CS Joint Conference on Digital Libraries*, Houston, USA.
- [Bar04] MJ Barnsley, JJ Settle, MA Cutter, DR Lobb, F Teston (2004). "The PROBA/CHRIS mission: a low-cost smallsat for hyperspectral multiangle observations of the Earth surface and atmosphere". *IEEE Transactions on Geoscience and Remote Sensing* 42(7): 1512-1520.
- [Bar99] JL Barker, SK Dolan, PA Sabelhaus, DL Williams, JR Irons, BL Markham, JT Bolek, SS Scott, RJ Thompson, JJ Rapp, TJ Arvidson, JR Kane, JC Storey (1999). "Landsat-7 mission and early results". *Sensors, Systems, and Next-Generation Satellites III*, Firenze, Italia.
- [Bec92] RJ Becherer, C Werner (1992). "Lidar for remote sensing". *Ed. Society of Photo Optical*.
- [Bek11] JK Bekkeng (2011). "Radiation effects on space electronics". *Sanntids Og Embedded Datasystemer*.
- [Ben75] JL Bentley (1975). "Multidimensional binary search trees used for associative searching". *Communications of the ACM* 18(9): 509-517.
- [Ber01] M Berger, M Rast, P Wursteisen, E Attema, J Moreno, A Müller, U Beisl, R Richter, M Schaepman, G Strub, MP Stoll, F Nerry, M Leroy (2001). "The DAISEX campaigns in support of a future land-surface-processes mission". *ESA Bulletin*: 101-111.
- [Ber04] P Bergsman (2004). "Xilinx FPGA blasted into orbit". *Tech. Rep. xCell Journal* 46.
- [Ber07] M Berg, C Poivey, D Petrick, K Label, M Friendlich, S Strawberry, H Kim (2007). "Risk reduction for use of complex devices in space projects". *IEEE Transactions on Nuclear Science* 54(6): 2137-2140.
- [Ber12] B Berruti (2012). "The Sentinel-3 Mission". *S3 Cal/Val Planning ESA/ESRIN*, Frascati, Italy.
- [Bev92] GA Bever (1992). "The development of an airborne information management system for flight test". *NASA Technical Memorandum 104251*.

- [Bez12] Q Bezuidenhout (2012). “Satellite communications. Strategy selection for optimal LEO satellite communication”.
- [Bir94] RJ Birk (1994). “Airborne hyperspectral sensor systems”. *IEEE Aerospace and Electronic Systems Magazine* 9(10): 26-33.
- [Bla12] I Blanes, J Serra-Sagrista, MW Marcellin, J Bartrina-Rapesta (2012). “Divide-and-conquer strategies for hyperspectral image processing: a review of their benefits and advantages”. *IEEE Signal Processing Magazine* 29(3): 71-81.
- [Boa10] JW Boardman (2010). “A novel approach to report HypsIRI location and observation information in a compact format for data distribution and utilization”. *HypsIRI Workshop 3*, Pasadena, USA.
- [Boa95] JW Boardman, FA Kruse, RO Green (1995). “Mapping target signatures via partial unmixing of AVIRIS data”. *JPL TRS 1992+*.
- [Boa99] JW Boardman (1999). “Precision geocoding of low altitude AVIRIS data: lessons learned in 1998”. *Eighth JPL Airborne Earth Science Workshop*, Pasadena, USA.
- [Boj03] S Bojinski, M Schaepman, D Schläpfer, K Itten (2003). “SPECCHIO: A spectrum database for remote sensing applications”. *Computers & Geosciences* 29(1): 27-38.
- [Bol96] G Bolotin (1996). “Space-Cube: A flexible computer architecture based on stacked modules”. *Multi-Chip Module Conference*, Santa Cruz, USA.
- [Bre00] M Breuer, J Albertz (2000). “Geometric correction of airborne whiskbroom scanner imagery using hybrid auxiliar data”. *International Archives of Photogrammetry and Remote Sensing XXXIII(B3)*: 93-100.
- [Bre10] FM Bréon, P Ciais (2010). “Spaceborne remote sensing of greenhouse gas concentrations”. *Comptes Rendus Geoscience* 342(4-5): 412-424.
- [Bru00] L Bruzzone, D Fernández (2000). “Automatic analysis of the difference image for unsupervised change detection”. *IEEE Transactions on Geoscience and Remote Sensing* 38(3): 1171-1182.
- [Bru02] L Bruzzone, D Fernández (2002). “An adaptative semiparametric and context-based approach to unsupervised change detection in multitemporal remote-sensing images”. *IEEE Transactions on Image Processing* 11(4): 452-466.
- [Buc08] R Buckley (2008). “JPEG 2000 - A practical digital preservation standard?”.
- [Buj09] EJA Bujalance (2009). “Microprocesadores ESA. Una visión de los microprocesadores en aplicaciones espaciales”.
- [Bur11] J Burger, A Gowen (2011). “Data handling in hyperspectral image analysis”. *Chemometrics and Intelligent Laboratory Systems* 108(1): 13-22.
- [CDFwww] Página web oficial del formato CDF: <http://cdf.gsfc.nasa.gov> (Último acceso: 28 Julio 2013).
- [CEOwww] Página web oficial de la organización CEOS: <http://www.ceos.org/> (Último acceso: 28 Julio 2013).
- [Cad02] A Cady (2002). “RAD750 system flight computer”.
- [Cam02] JB Campbell (2002). “Introduction to remote sensing”. *Ed. Taylor & Francis*.
- [Car97] TN Carlson, DA Ripley (1997). “On the relation between NDVI, fractional vegetation cover, and leaf area index”. *Remote Sensing of Environment* 62(3): 241-252.

- [Car99] C Carmichael, E Fuller, P Blain, M Caffrey (1999). "SEU mitigation techniques for Virtex FPGAs in space applications". *Military and Aerospace Programmable Logic Devices International Conference*, Laurel, USA.
- [Cer07] Certification Authorities Software Team (2007). "Simple electronic hardware and RTCA document DO-254 and EUROCAE document ED-80, design assurance guidance for airborne electronic hardware".
- [Cet04] H Cetin (2004). "Comparison of spaceborne and airborne hyperspectral imaging systems for environmental mapping". *XXth ISPRS Congress*, İstanbul, Türkiye.
- [Cha01] C Chang, H Ren, S Chiang (2001). "Real-time processing algorithms for target detection and classification in hyperspectral imagery". *IEEE Transactions on Geoscience and Remote Sensing* 39(4): 760-768.
- [Cha07a] C Chang (2007). "Hyperspectral data exploitation. Theory and applications". *Ed. Wiley*.
- [Cha07b] JM Chaves, MA Vega, P Martínez, JM Sánchez, JA Gómez (2007). "Diffuse Matrix: An optimized data structure for the storage and processing of hyperspectral images". *International Conference on Signal Processing and Multimedia Applications*, Barcelona, Spain.
- [Cha13] C Chang (2013). "Hyperspectral data processing: algorithm design and analysis". *Ed. Wiley*.
- [Cha93] SH Chang, MJ Westfield, F Lehmann, D Oertel, R Richter (1993). "79-channel airborne imaging spectrometer". *Imaging Spectrometry of the Terrestrial Environment*, Orlando, USA.
- [Chi12] S Chien, J Doubleday, K Ortega, D Tran, J Bellardo, A Williams, J Piug-Suari, G Crum, T Flatley (2012). "Onboard autonomy and ground operations automation for the Intelligent Payload Experiment (IPEX) Cubesat mission". *International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Torino, Italia.
- [Chrwww] Página web oficial del instrumento CHRIS: <https://earth.esa.int/web/guest/missions/esa-operational-eo-missions/proba/instruments/chris> (Último acceso: 28 Julio 2013).
- [Chu07] E Chuvieco (2007). "Teledetección ambiental: La observación de la Tierra desde el espacio". *Ed. Ariel*.
- [Cla07] RN Clark, GA Swayze, R Wise, E Livo, T Hoefen, R Kokaly, SJ Sutley (2007). "USGS digital spectral library splib06a: U.S. Geological Survey, Digital Data Series 231."
- [Cla89] RN Clark, TL Roush (1984). "Reflectance spectroscopy: Quantitative analysis techniques for remote sensing applications". *Journal of Geophysical Research: Solid Earth* 89(B7): 6329-6340.
- [Cla96] RN Clark, GA Swayze (1996). "Evolution in imaging spectroscopy analysis and sensor signal-to-noise: An examination of how far we have come". *Sixth Annual JPL Airborne Earth Science Workshop*, Pasadena, USA.
- [Com94] Commission of the European Communities (1994). "Proposals for council decisions concerning the specific programmes implementing the Fourth European Community Framework Programme for Research, Technological Development and Demonstration Activities". *COM 94(68)*.
- [Cra00] M Cramer, D Stallmann, N Haala (2000). "Direct georeferencing using GPS/inertial exterior orientations for photogrammetric applications". *XIXth ISPRS Congress*, Amsterdam, Nederland.

- [Cra01] M Cramer, D Stallmann (2001). “On the use of GPS/inertial exterior orientation parameters in airborne photogrammetry”. *ISPRS Workshop - High Resolution Mapping from Space*, Hannover, Deutschland.
- [Cri11] A Cristo, P Martínez, D Valencia, RM Pérez, LM Hernández (2011). “Adaptation of the Diffused Matrix Image Format to store simulated data from FLEX / Sentinel-3 future ESA missions”. *3rd International Conference on Image Processing & Communications*, Bydgoszcz, Poland.
- [Cri12a] A Cristo, D Valencia, RM Pérez, P Martínez, LM Hernández (2012). “Adaptation of the Diffused Matrix Format (DMF) for remote sensing image processing”. *IEEE Gold Remote Sensing*, Roma, Italy.
- [Cri12b] A Cristo, RM Pérez, P Martínez, LM Hernández, E Infante (2012). “Simulation of Sentinel-2 data acquisition and fusion by using the Diffused Matrix Format”. *ESA Sentinel-2 Preparatory Symposium*, Frascati, Italy.
- [Cri12c] A Cristo, D Valencia, P Martínez, RM Pérez (2012). “DMF data format usage for change detection”. *Journal of Electrical and Computer Engineering* 2012: 1-9.
- [Cve93] SR Cvetkovic, GJ Robertson (1993). “Spacecraft design considerations for small satellite remote sensing”. *IEEE Transactions on Aerospace and Electronic Systems* 29(2): 391-403.
- [Cza05] DR Czajkowski, MP Pagey, PK Samudrala, M Goksel, MJ Viehman (2005). “Low power, high-speed radiation hardened computer & flight experiment”. *IEEE Aerospace Conference*, Big Sky, USA.
- [DAIwww] Página web oficial del instrumento DAIS 7915: <http://www.op.dlr.de/DAIS/> (Último acceso: 28 Julio 2013).
- [DATwww] Web oficial del proyecto DATASAT: <http://www.vision.uji.es/datasat/> (Último acceso: 28 Julio 2013).
- [DAm08] J D’Ambrosia, D Law, M Nowell (2008). “40 Gigabit Ethernet and 100 Gigabit Ethernet. Technology overview”.
- [DYN99] DYNEX Semiconductor (1999). “MA31751 memory management & block protection unit”.
- [Dan03] SP Dandamudi (2003). “Fundamentals of computer organization and design”. *Ed. Springer*.
- [Dav00] M Davidson, T LeToan, F Mattia, G Satalino, T Manninen, M Borgeaud (2000). “On the characterization of agricultural soil roughness for radar remote sensing studies”. *IEEE Transactions on Geoscience and Remote Sensing* 38(2): 630-640.
- [DeL11] D de Luna, Z Zalewski (2011). “FPGA level in-hardware verification for DO-254 compliance”. *IEEE/AIAA 30th Digital Avionics Systems Conference*, Seattle, USA.
- [DeM05] RF DeMara, K Zhang (2005). “Autonomous FPGA fault handling through competitive runtime reconfiguration”. *NASA/DoD Conference on Evolvable Hardware*, Washington D.C., USA.
- [Del04] Dell Computer Corporation, Intel Corporation, Maxtor Corporation, Seagate Technology, Vitesse Semiconductor Corporation (2004). “Serial ATA II: Electrical specification”.
- [Del11] A del Amo (2001). “Power PC”. *Arquitectura de Computadores*.

- [Den12] A Deniz (2012). “History of remote sensing”. *Remote Sensing*.
- [Dhr03] N Dhruv, TM Kwon, SA Patwardhan, E Kwon (2003). “CDF archival of large-scaled ITS data for efficient archival, retrieval, and portability”. *Transportation Research Board (TRB) 2003 Annual Meeting*, Washington D.C., USA.
- [Dom08] C Domokos, Z Kato, J Francos (2008). “Parametric estimation of affine deformations of binary images”. *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Las Vegas, USA.
- [Don08] C Donlon (2008). “Sentinel-3 OLCI and SLSTR”. *Medspiration / GlobColour Symposium*, Frascati, Italy.
- [Don95] DL Donoho, LM Johnstone (1995). “Adapting to unknown smoothness via wavelet shrinkage”. *Biometrika* 81(3): 425-455.
- [Dou06] R Douglas (2006). “Quick history of remote sensing”. *Applied Remote Sensing*.
- [Dox13] G Doxani, M Papadopoulou, P Lafazani, M Tsakiri-Strati, E Mavridou (2013). “Sun glint correction of very high spatial resolution images”. *Thales, in Honor of Prof. Emeritus Michael E. Contidakis*: 329-340.
- [Doz04] J Dozier, TH Painter (2004). “Multispectral and hyperspectral remote sensing of Alpine snow properties”. *Annual Review of Earth and Planetary Sciences* 32: 465-494.
- [Du03] Q Du, H Ren (2003). “Real time constrained linear discriminant analysis to target detection and classification in hyperspectral imagery”. *Pattern Recognition* 36: 1-12.
- [Dub01] PA Dubock, F Spoto, J Simpson, D Spencer, E Schutte, H Sontag (2001). “The Envisat satellite and its integration”. *ESA Bulletin* 106: 26-45.
- [EIA69] Electronic Industries Association, Engineering Dept. (1969). “Interface between data terminal equipment and data communication equipment employing serial binary data interchange”. *Ed. Electronic Industries Association, Engineering Dept.*
- [ETMwww] Página web oficial del instrumento ETM+: <http://landsat.gsfc.nasa.gov/about/etm+.html> (Último acceso: 28 Julio 2013).
- [EU12] European Commission (2012). “Annex 3. Current definition of primary products”.
- [Eas00a] ML Eastwood, RO Green, CM Sarture, BJ Chippindale, CJ Chovit, JA Faust, DL Johnson, SP Monacos, JJ Raney (2000). “Recent improvements to the AVIRIS sensor: Flight season 2000”. *Airborne Visible / Infrared Imaging Spectrometer (AVIRIS) Earth Science Workshop 2000*, Pasadena, USA.
- [Eas00b] ML Eastwood, CM Sarture, BJ Chippindale, CJ Chovit, JA Faust, RO Green, DL Johnson, SP Monacos, JJ Raney (2000). “Recent improvements to the AVIRIS sensor”. *NASA Technical Report*.
- [Eic12] J Eickhoff (2012). “Onboard computers, onboard software and satellite operations. An introduction”. *Ed. Springer*.
- [Ekl01] L Eklundh, L Harrie, A Kuusk (2001). “Investigating relationships between Landsat ETM+ sensor data and leaf area index in a boreal conifer forest”. *Remote Sensing of Environment* 78(3): 239-251.
- [Esp09] JK Espenshade (2009). “Scalable framework for heterogeneous clustering of commodity FPGAs”. *Master’s Thesis, Rochester Institute of Technology, Kate Gleason College of Engineering, New York*.

- [Esp11] J Esper, T Flatley, B Bull (2011). “Modular reconfigurable spacecraft Small Rocket/Spacecraft Technology platform SMART”. *Small Satellite Conference*, Logan, USA.
- [FITwww] Página web oficial del formato FITS: <http://fits.gsfc.nasa.gov> (Último acceso: 28 Julio 2013).
- [Fal02] E Falkner, D Morgan (2002). “Aerial mapping. Methods and applications”. *Ed. Lewis Publishers*.
- [Fan04] W Fang, MY Jin (2004). “On board processor development for NASA’s spaceborne imaging radar with VLSI system-on-chip technology”. *2004 IEEE International Symposium on Circuits and Systems*, Vancouver, Canada.
- [Fer05] A Fernández-Renau, JA Gómez, E Miguel (2005). “The INTA AHS system”. *Sensors, Systems and Next-Generation Satellites IX*, Brugge, Belgique.
- [Fie10] B Fienberg (2010). “Xilinx launches first high-density, rad-hard reconfigurable FPGA for space applications”.
- [Fla10] T Flatley (2010). “SpaceCube on-board science data processing”. *American Geophysical Union Fall Meeting*, San Francisco, USA.
- [Fla11] T Flatley (2011). “Advanced hybrid on-board science data processor - SpaceCube 2.0”. *Earth Science Technology Forum*, Pasadena, USA.
- [Fla12] T Flatley (2012). “SpaceCube: A family of reconfigurable hybrid on-board science data processors”. *NASA/ESA Conference on Adaptive Hardware and Systems*, Nürnberg/Erlang, Deutschland.
- [Fog03] RJ Fogler (2003). “Multi- and hyper- spectral sensing for autonomous ground vehicle navigation”.
- [Fol10] M Folk, E Pourmal (2010). “Balancing performance and preservation. Lessons learned with HDF5”. *U.S. Workshop on Roadmap for Digital Preservation Interoperability Framework*, Gaithersburg, USA.
- [For92] BV Fortes, VP Lukin (1992). “Modeling of the image observed through a turbulent atmosphere”. *Atmospheric Propagation and Remote Sensing*, Orlando, USA.
- [Fra00] PD Franzon (2000). “Introduction to ASIC design”. *ASIC and FPGA Design and Verification*.
- [Gai05] J Gaisler (2005). “LEON3-FT-RTAX SEU test results”. *Gaisler Research* 1: 1-8.
- [Gai07] J Gaisler (2007). “LEON3 SPARC processor, the past present and future”. *RAMP Winter Retreat*, Berkely, USA.
- [Gai94] J Gaisler (1994). “Concurrent error-detection and modular fault-tolerance in a 32-bit processing core for embedded space flight applications”. *Twenty-Fourth International Symposium on Fault-Tolerant Computing*, Austin, USA.
- [Gao09] BC Gao, MJ Montes, CO Davis, AFH Goetz (2009). “Atmospheric correction algorithms for hyperspectral remote sensing data of land and ocena”. *Remote Sensing of Environment* 113(1): S17-S24.
- [Gel04] P Geladi, J Burger, T Lestander (2004). “Hyperspectral imaging: calibration problems and solutions”. *Chemometrics and Intelligent Laboratory Systems* 72(2): 209-217.

- [Gen08] R Gens (2008). "Remote sensing: Data formats and software packaging". *Remote Sensing and GIS Seminar*.
- [Geowww] Página web oficial del formato GeoTIFF: <http://trac.osgeo.org/geotiff/> (Último acceso: 28 Julio 2013).
- [Gil05] E Gill (2005). "Comparison of the performance of microprocessors for space-based navigation applications".
- [Gla09] T Glavich, RO Green, SJ Hook, B Middleton, F Rogez, S Ungar (2009). "HypSIRI decadal survey mission development status". *Decadal Survey Symposium*, Washington D.C., USA.
- [God06] J Godfrey (2006). "SpaceCube IRAD development effort".
- [Goe85] AFH Goetz, G Vane, JR Solomon, BN Rock (1985). "Imaging spectrometry for Earth Remote Sensing". *Science* 228(4704): 1147-1153.
- [Gom07] JA Gómez, E de Miguel, OG de la Cámara, A Fernández-Renau (2007). "Status of the INTA AHS sensor". *5th EARSeL Workshop on Imaging Spectroscopy*, Bruges, Belgium.
- [Gon12] LF González, MPG Castro, F Tamagnone (2012). "Multidisciplinary design and flight testing of a remote gas/particle airborne sensor system". *28th International Congress of the Aeronautical Sciences*, South Bank, Australia.
- [Goo13] KR Goodwin (2013). "N-Modular redundancy". *Fault Tolerant Computing*.
- [Gos05] AA Goshtasby (2005). "2-D and 3-D image registration for medical, remote sensing and industrial applications". *Ed. Wiley Press*.
- [Gra03] PS Graham, MP Caffrey, Los Alamos National Laboratory, MJ Wirthlin, DE Johnson, NH Rollins (2003). "Reconfigurable computing in space: From current technology to reconfigurable systems-on-a-chip". *IEEE Aerospace Conference*, Big Sky, USA.
- [Gre11] DJ Greaves (2). "System on chip. Design and modelling".
- [Gre98] RO Green, ML Eastwood, CM Sarture, TG Chrien, M Aronsson, BJ Chippendale, JA Faust, BE Pavri, CJ Chovit, M Solis, MR Olah, O Williams (1998). "Imaging spectroscopy and the Airborne Visible / Infrared Imaging Spectrometer (AVIRIS)". *Remote Sensing of Environment* 65(3): 227-248.
- [Gro13] B Grofic (2013). "Earth science space missions in the 21st century". *IEEE Aerospace Conference*, Big Sky, USA.
- [Gua02] JA Gualtieri, JC Tilton (2002). "Hierarchical segmentation of hyperspectral data". *2002 AVIRIS Earth Science and Applications Workshop*, Pasadena, USA.
- [Gua05] L Guanter, L Alonso, J Moreno (2005). "First results from the PROBA/CHRIS hyperspectral / multiangular satellite system over land and water targets". *IEEE Geoscience and Remote Sensing Letters* 2(3): 250-254.
- [Gua10] L Guanter, L Alonso, L Gómez-Chova, M Meroni, R Preusker, J Fischer, J Moreno (2010). "Developments for vegetation fluorescence retrieval from spaceborne high-resolution spectrometry in the O2-A and O2-B absorption bands". *Journal of Geophysical Research: Atmospheres* 115(D19): 1984-2012.
- [Gup97] R Gupta, RI Hartley (1997). "Linear pushbroom cameras". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19(9): 963-975.

- [HDFwww] Página web oficial del formato HDF: <http://www.hdfgroup.org> (Último acceso: 28 Julio 2013).
- [Hal11] AC Hall, GJ Schumann, JL Bamber, PD Bates (2011). "Tracking water level changes of the Amazon Basin with space-borne remote sensing and integration with large scale hydrodynamic modelling: A review". *Physics and Chemistry of the Earth, Parts A/B/C* 36(7-8): 223-231.
- [Han05] L Hanson (2005). "NetCDF file data access".
- [Har87] RM Haralick, SR Sternberg, X Zhuang (1987). "Image analysis using mathematical morphology". *IEEE Transactions on Pattern Analysis and Machine Intelligence* PAMI-9(4): 532-550.
- [Har92] PJ Hardin, CN Thomson (1992). "Fast nearest neighbor classification methods for multispectral imagery". *The Professional Geographer* 44(2): 191-202.
- [Hen98] FM Henderson, AJ Lewis (1998). "Principles and applications of imaging radar. Manual of remote sensing, Volume 2". Ed. Wiley.
- [Her11] LM Hernández, A Cristo, P Martínez, RM Pérez (2011). "Diffused Matrix Format (DMF): A data structure for airborne and spaceborne image processing". *ESA Advanced Training Course in Land Remote Sensing*, Kraków, Poland.
- [Hol11] S Holzwarth, M Bachmann, M Freer, M Hoffmann (2011). "Standards for airborne hyperspectral image data". *EARSeL 7th SIG-Imaging Spectroscopy Workshop*, Edinburgh, United Kingdom.
- [Hu08] X Hu, Y Deng, X Wang (2008). "Collection of techniques on dealing with image files of FITS format and applications to the solar observation". *Astronomical Research and Technology* 5(1): 55-65.
- [Hua10] X Huang, L Zhang (2010). "A comparative study of spatial approaches for urban mapping using hyperspectral ROSIS images over Pavia City, northern Italy". *International Journal of Remote Sensing* 30(12): 3205-3221.
- [Hub12] PA Hubanks, A Chu, B Ridgway, K Strabala, S Platnick, S Mattoo, E Moody, MD King, R Hucek (2012). "MODIS Atmosphere QA Plan for collection 005 and 051".
- [Hus10] W Hussain, L Lehtonen, T Patyk (2010). "FPGA and ASIC". *Digital Design for FPGA*.
- [IEEE85] ANSI/IEEE (1985). "IEEE standard for binary floating point arithmetic". *ANSI/IEEE Standard 754-1985*.
- [JPEwww] Página web oficial del formato JPEG2000: <http://www.jpeg.org/jpeg2000/> (Último acceso: 28 Julio 2013).
- [Jac91] RD Jackson, AR Huete (1991). "Interpreting vegetation indices". *Preventive Veterinary Medicine* 11: 185-200.
- [Jen08] PP Jenkins, RJ Walters, MJ Krasowski, JJ Chapman, PG Ballard, JA Vasquez, DR Mahony, SN LaCava, WR Braun, R Skalitzky, NF Prokop, JM Flatco, LC Greer, KB Gibson, WH Kinard, HG Pippin (2008). "MISSE7: Building a permanent environmental testbed for the International Space Station". *Protection of Materials and Structures from Space Environment*, Toronto, Canada.
- [Jos05] G Joseph (2005). "Fundamentals of remote sensing". Ed. Universities Press.

- [Jus02] CO Justice, JRG Townshend, EF Vermonte, E Masuoka, RE Wolfe, N Saleous, DP Roy, JT Morissette (2002). "An overview of MODIS Land data processing and product status". *Remote Sensing of Environment* 83(1-2): 3-15.
- [Kam08] R Kamal (2008). "Direct memory access". *Device Drivers and Interrupts Service Mechanism*.
- [Kau12] H Kauhanen, P Rönholm (2012). "Image acquisition constraints for panoramic frame camera imaging". *The XXII Congress of the International Society for Photogrammetry and Remote Sensing*, Melbourne, Australia.
- [Ken99] O Kenta, R Shuichi, M Tsuneo, N Yoshiki (1999). "Evaluation of thickness of desert varnish on the rock surface using thermal infrared multispectral remote sensing data". *Journal of Remote Sensing Society of Japan* 19(1): 45-58.
- [Kes02] N Keshava, JF Mustard (2002). "Spectral unmixing". *IEEE Signal Processing Magazine* 19(1): 44-57.
- [Key04] W Keydel (2004). "Present and future airborne and space-borne systems". *Radar Polarimetry and Interferometry*.
- [Kha12] GN Khan (2012). "FPGA based embedded SoPCs: System on Programmable Chips". *Embedded Computer Systems*.
- [Kla94] JW Klatt (1994). "Error characterization of spectral products using a factorial designed experiment".
- [Kle07] LA Klerk, A Broersen, IW Fletcher, R Van Liere, RMA Heeren (2007). "Extended data analysis strategies for high resolution imaging MS: New methods to deal with extremely large image hyperspectral datasets". *International Journal of Mass Spectrometry* 260(2-3): 222-236.
- [Kno07] K Knödel, G Lange, H Voigt (2007). "Environmental Geology". *Ed. Springer*.
- [Kom05] GJ Komar, J Wang, T Kimura (2005). "Enabling sensor and platform technologies for spaceborne remote sensing". *Ed. Society of Photo Optical*.
- [Koz10] Q Koziol, E Pourmal (2010). "HDF5: State of the Union". *SC10*, New Orleans, USA.
- [Kra10] S Kraft, B Carnicero, M Drusch, JL Bézy, R Meynart (2010). "FIMAS - Feasibility study of a fluorescence imaging spectrometer to be flown on a small platform in tandem with Sentinel 3". *International Conference on Space Optics*, Rhodes, Greece.
- [Kra95] WB Krabill, RH Thomas, CF Martin, RN Swift, EB Frederick (1995). "Accuracy of airborne laser altimetry over the Greenland ice sheet". *International Journal of Remote Sensing* 16(7): 1211-1222.
- [Kum95] SR Kumar, S Ameenulla, A Prabhu (1995). "MONTBLEX tower observations: Instrumentation, data acquisition and data quality". *Proceedings of the Indian Academy of Sciences - Earth and Planetary Sciences* 104(2): 221-248.
- [Kun91] B Kunkel, F Blechinger, D Viehmann, H Van Der Piepen, R Doerffer (1991). "RODIS imaging spectrometer and its potential for ocean parameter measurements (airborne and spaceborne)". *International Journal of Remote Sensing* 12(4): 753-761.
- [Kuo07] I Kuon, R Tessier, J Rose (2007). "FPGA architecture: Survey and challenges". *Foundations and Trends in Electronic Design Automation* 2(2): 135-253.

- [Kur10] F Kurz, D Rosenbaum, U Thomas, J Leitloff, G Palubinskas, K Zeller, P Reinartz (2010). "Near real time airborne monitoring system for disaster and traffic applications". *Near Real Time Airborne Monitoring System for Disaster and Traffic Applications*, Paris, France.
- [Lan02] D Landgrebe (2002). "Hyperspectral image data analysis as a high dimensional signal processing problem". *IEEE Signal Processing Magazine* 19(1): 17-28.
- [Lan03] DA Landgrebe (2003). "Signal theory methods in multispectral remote sensing". *Ed. Wiley-Interscience*.
- [Lat11] Lattice Semiconductor Corporation (2008). "LatticeMico8 processor reference manual".
- [Lat12] Lattice Semiconductor Corporation (2012). "LatticeMico32 processor reference manual".
- [Le04] C Le, S Chan, F Cheng, W Fang, M Fischman, S Hensley, R Johnson, M Jourdan, M Marina, B Parham, F Rogez, P Rosen, B Shah, S Taft (2004). "Onboard FPGA-based SAR processing for future spaceborne systems". *IEEE Radar Conference*, Pennsylvania, USA.
- [LeB10] J LeBoeuf (2000). "Practical applications of remote sensing technology - An industry perspective". *HortTechnology* 10(3): 475-480.
- [LeM11] J Le Moigne, NS Netanyahu, RD Eastman (2011). "Image registration for remote sensing". *Ed. Cambridge*.
- [LeM12] J Le Moigne (2012). "Image registration for remote sensing". *IS&T Colloquia Lecture Series*.
- [Lee80] JS Lee (2008). "Digital image enhancement and noise filtering by use of local statistics". *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2: 165-168.
- [Lef02] MA Lefsky, WB Cohen, GG Parker, DJ Harding (2002). "Lidar remote sensing for ecosystem studies". *BioScience* 52(1): 19-30.
- [Len03] M Lennon, V Mariette, A Coat, V Verbeque, P Mouge (2003). "3rd EARSeL Workshop on Image Spectroscopy". *Detection and mapping of the November 2002 PRESTIGE Tanker oil spill in Galicia, Spain, with the airborne multispectral CASI sensor*, Oberpfaffenhofen, Deutschland.
- [Lew09] S Lewis (2009). "Remote sensing for natural disasters: Facts and figures".
- [Lie09] W Lie, W Feng-Yan (2009). "Dynamic partial reconfiguration in FPGAs". *Third International Symposium on Intelligent Information Technology Application*, Nanchang, China.
- [Liu12] T Liu, H Wang (2012). "The design of infrared spaceborne remote sensing signal processing circuit for multi-spectral and multi-focal plane". *Infrared, Millimeter-Wave, and Terahertz Technologies II*, Beijing, China.
- [Liu13] C Liu (2013). "Image rectification and restoration". *Analysis and applications of remote sensing imagery*.
- [Los12] L Losik (2012). "Results from the prognostic analysis completed on the NASA extreme Ultra Violet Explorer satellite". *IEEE Aerospace Conference*, Big Sky, USA.

- [Lun99] RS Lunetta, CD Elvidge (1999). "Remote sensing change detection: environmental monitoring methods and applications". *Ed. Taylor & Francis Ltd.*
- [MI03] Maxim Integrated (2003). "Determining clock accuracy requirements for UART communications".
- [MODwww] Página web oficial del instrumento MODIS: <http://modis.gsfc.nasa.gov/> (Último acceso: 28 Julio 2013).
- [Man10] JF Manera, L Rodríguez, C Delrieux, R Coppo (2010). "Aerial image acquisition and processing for remote sensing". *Journal of Computer Science & Technology* 10(2): 97-103.
- [Mar04] JR Marshall, RW Berger, JC Rodger (2012). "Reconfigurable processing subsystems in spaceborne applications". *IEEE Aerospace Conference*, Big Sky, USA.
- [Mar09] P Marriot, AD Stone (2009). "Understanding DO-254 compliance for the verification of airborne digital hardware".
- [Mar10] P Martínez, A Cristo, M Koch, RM Pérez, T Schmid, LM Hernández (2010). "Diffused Matrix Format: A new storage and processing format for airborne hyperspectral sensor images". *Sensors* 10: 4996-5013.
- [Mas10] DC Mason, R Speck, B Devereux, GJP Schumann, JC Neal, PD Bates (2010). "Flood detection in urban areas using TerraSAR-X". *IEEE Transactions on Geoscience and Remote Sensing* 48(2): 882-894.
- [Mat04] PM Mather (2004). "Computer processing of remotely-sensed images. An introduction". *Ed. Wiley.*
- [McD08] J McDonald (2008). "Insider's guide: FPGAs, tools, and boards".
- [McL12] J McLeroy (2012). "Highlights of DoD research on the ISS".
- [McW05] N McWilliam, R Teeuw, M Whiteside, P Zukowskyj (2005). "GIS GPS and remote sensing. Field techniques manual". *Royal Geographical Society with IBG.*
- [Mea01] P Mead (2001). "Systems on Programmable Chips - Will SOPC eclipse SoC?". *Designing Systems on Silicon.*
- [Meh05] PC Mehlitz, J Penix (2005). "Expecting the unexpected - Radiation hardened software". *American Institute of Aeronautics and Astronautics Infotech at Aerospace:* 1-10.
- [Mil80] EJ Milton (1980). "A portable multiband radiometer for ground data collection in remote sensing". *International Journal of Remote Sensing* 1(2): 153-165.
- [Mob99] CD Mobley (1999). "Estimation of the remote-sensing reflectance from above-surface measurements". *Applied Optics* 38(36): 7442-7455.
- [Mol04] BV Mol, K Ruddick (2004). "The Compact High Resolution Imaging Spectrometer (CHRIS): the future of hyperspectral satellite sensors. Imagery of Oostende coastal and inland waters". *Airborne Imaging Spectroscopy Workshop*, Bruges, Belgium.
- [Mor04] J Moreno, L Alonso, G Fernández, JC Fortea, S Gandía, L Guanter, JC García, JM Martí, J Melia, F Camacho (2004). "The SPECTRA Barrax Campaign (SPARC): Overview and first results from CHRIS data". *2nd CHRIS/PROBA Workshop*, Frascati, Italia.

- [Mor08a] B Morse (2008). "Introduction to digital signal and image processing. Low-pass filtering".
- [Mor08b] B Morse (2008). "Introduction to digital signal and image processing. High-pass and band-pass filtering".
- [Mor09] J Moreno, L Guanter, L Alonso, L Gómez-Chova, M Drusch, S Kraft, B Carnicero, J Bezy (2009). "Monitoring dynamical vegetation processes with solar-induced chlorophyll fluorescence measurements from space". *American Geophysical Union, Fall Meeting*, San Francisco, USA.
- [Mos00] MMR Mostafa, K Schwarz (2000). "A multi-sensor system for airborne image capture and georeferencing". *Photogrammetric Engineering & Remote Sensing* 66(12): 1417-1423.
- [Mos01] MMR Mostafa, KP Scharwz (2001). "Digital image georeferencing from a multiple camera system by GPS/INS". *ISPRS Journal of Photogrammetry and Remote Sensing* 56(1): 1-12.
- [Mur05] MVA Murzina, JP Farrell (2005). "Dynamic hyperspectral imaging". *Nondestructive Detection and Measurement for Homeland Security III* 135: 1-10.
- [NAS06a] National Aeronautics and Space Administration (NASA) (2006). "The state-of-technology at Goddard: Center director Ed Weiler speaks out". *Goddard Tech Trends* 2(4).
- [NAS06b] National Aeronautics and Space Administration (NASA) (2006). "Hubble Space Telescope Servicing Mission 4". *NASA Facts*.
- [NAS07] National Aeronautics and Space Administration (NASA) (2007). "International Space Station basics".
- [NAS09a] National Aeronautics and Space Administration (NASA) (2009). "Hubble Space Telescope Servicing Mission 4. Science Instrument Command and Data Handling Module". *NASA Facts*.
- [NAS09b] National Aeronautics and Space Administration (NASA) (2009). "What would you rather have: More data or perfect data?". *Goddard Tech Trends* 5(3).
- [NAS09c] National Aeronautics and Space Administration (NASA) (2009). "SpaceCube offers more processing power - and potential for commercial applications". *NASA Tech Transfer* 7(1).
- [NAS10] National Aeronautics and Space Administration (NASA) (2010). "Tour of the electromagnetic spectrum".
- [NAS11] National Aeronautics and Space Administration (NASA) (2011). "STS-134 mission summary". *NASA Mission Summary*.
- [NAS12] National Aeronautics and Space Administration (NASA) (2012). "SpaceCube Mini". *Goddard Tech Briefs*.
- [NAS13] National Aeronautics and Space Administration (NASA) (2013). "Flatley wins prestigious AAS Award". *Goddard Tech Trends* 9(3).
- [Naa09] BJ Naasz, RD Burns, SZ Queen, JV Eepoel, J Hannah, E Skelton (2009). "The HST SM4 Relative Navigation Sensor System: Overview and preliminary testing results from the Flight Robotics Lab". *The Journal of the Astronautical Sciences* 57(1-2): 457-483.
-

- [Net01] F Neto (2001). "Airborne and spaceborne imagery for mapping". *The Cal Poly Pomona Journal of Interdisciplinary Studies* 2001: 119-126.
- [Netwww] Página web oficial del formato NetCDF: <http://www.unidata.ucar.edu/software/netcdf/> (Último acceso: 28 Julio 2013).
- [Ngu08] Q Nguyen, W Yuknis, S Pursley, N Haghani, D Albaijes, O Haddad (2008). "A high performance command and data handling system for NASA's Lunar Reconnaissance Orbiter". *AIAA SPACE 2008 Conference & Exposition*, San Diego, USA.
- [Nor05] J Norhuzaimin, HH Maimun (2005). "The design of high speed UART". *Asia - Pacific Conference on Applied Electromagnetics*, Johor Bahru, Malaysia.
- [Nor09] CD Norton, TA Werne, PJ Pingree, S Geier (1999). "An evaluation of the Xilinx Virtex-4 FPGA for on-board processing in an advanced imaging system". *IEEE Aerospace Conference*, Big Sky, USA.
- [Noy07] G Noyel, J Angulo, D Jeulin (2007). "Morphological segmentation of hyperspectral images". *Image Analysis and Stereology* 26: 1-9.
- [Ots79] N Otsu (1979). "A threshold selection method from gray-level histograms". *IEEE Transactions on Systems, Man, and Cybernetics* SMC-9(1): 62-66.
- [PCI04] PCI Geomatics (2004). "Hyperspectral image analysis. Geomatica 9".
- [PIC99] PCI Industrial Computers Manufacturers Group (PICMG) (1999). "CompactPCI specification".
- [Pal97] M Palaniyandi, V Nagarathinam (1997). "Land use / land cover mapping and change detection using space borne data". *Journal of the Indian Society of Remote Sensing* 25(1): 27-33.
- [Pan98] PV Panetta, H Culver, J Gagosian, M Johnson, J Kellogg, D Mangus, T Michalek, V Sank, S Tompkins (1998). "NASA/GSFC nano-satellite technology development". *Science Closure and Enabling Technologies for Constellation Class Missions*: 115-122.
- [Pap01] R Papasin, Y Gawdiak, DA Maluf, C Leidich, PB Tran (2001). "Airport remote tower sensor systems". *NASA Tech Docs*.
- [Par03] CL Parkinson (2003). "Aqua: an Earth-Observing Satellite mission to examine water and other climate variables". *IEEE International Geoscience and Remote Sensing Symposium*, Toulouse, France.
- [Par99] SM Parkes (1999). "Space Wire: the standard". *European Space Agency Publications ESA SP 447*: 111-116.
- [Pas95] A Pasetti, W Pree (1995). "The component software challenge for real-time systems". *First International Workshop on Active and Real-Time Database Systems*, Skövde, Sverige.
- [Pet11] D Petrick, D Espinosa, T Flatley, A Geist, G Crum, M Lin, J Hosler, M Buenfil, K Blank (2009). "SpaceCube: Success on Hubble SM4, current missions and ongoing platform advances". *Information Science & Technology Colloquium Series*, Greenbelt, USA.
- [Pha13] D Phantavong, M Bansal, M Chitnis, DJ Wang (2013). "Optimization techniques for efficient implementation of DSP in FPGAs".
- [Phi12] S Phillips (2012). "Emission and absorption spectroscopy - Background". *General Chemistry I Laboratory*.

- [Phi13] W Philpot (2013). “Geometric correction”. *Digital Image Processing*.
- [Pia99] E Piazza (1999). “Real-time processing of multispectral satellite remote sensing images”. *Applications of Digital Image Processing XXII*, Denver, USA.
- [Pin08] PJ Pingree, LJ Scharenbroich, TA Werne (2008). “FPGA coprocessor for accelerated classification of images”. *NASA Technical Reports Server*.
- [Pla03] S Platnick, MD King, SA Ackerman, WP Menzel, BA Baum, JC Riédi, RA Frey (2003). “The MODIS cloud products: algorithms and examples from Terra”. *IEEE Transactions on Geoscience and Remote Sensing* 41(2): 459-473.
- [Pol00] A Polesel, G Ramponi, VJ Mathews (2000). “Image enhancement via adaptative unsharp masking”. *IEEE Transactions on Image Processing* 9(3): 505-5010.
- [Por03] C Portalés, JL Lerma (2003). “Sensores hiperespectrales aerotransportados. Eliminación del defecto de deriva”. *Mapping Interactivo* 87.
- [Pum03] HC Pumphrey (2003). “Image distortions and how to correct them”. *Remote Sensing and Image Processing*.
- [Pyr05] Pyramid Semiconductor Corporation (2005). “PACE1753 single chip, 40 MHz CMOS MMU/Combo”.
- [Pér10] RM Pérez, A Cristo, M Koch, P Martínez, T Schmid, A Gualtieri, LM Hernández (2010). “FLEX / Sentinel-3 data fusion by using the Diffused Matrix Format (DMF)”. *4th International Workshop on Remote Sensing of Vegetation*, Valencia, Spain.
- [Quiwww] Página web oficial del instrumento Quickbird: <http://www.digitalglobe.com/about-us/content-collection#satellites&quickbird> (Último acceso: 28 Julio 2013).
- [ROSwww] Página web oficial del instrumento ROSIS: http://www.opairs.aero/rosis_en.html (Último acceso: 28 Julio 2013).
- [RTCwww] Página web oficial de la organización RTCA Incorporated: <http://www.rtca.org/> (Último acceso: 28 Julio 2013).
- [Ram97] M Ramström, Bo Törnberg, R Svenningsson (1997). “32-bit microprocessor and computer system development. ERC32 system overview Rev. CBA”.
- [Ras08] U Rascher, B Gioli, F Miglietta (2008). “FLEX - Fluorescence Explorer: A remote sensing approach to quantify spatio-temporal variations of photosynthetic efficiency from Space”. *Photosynthesis. Energy from the Sun*: 1388-1390.
- [Ras11] B Rasaiah, TJ Malthus, SD Jones, C Bellman (2011). “The role of hyperspectral metadata in hyperspectral data exchange and warehousing”. *EARSeL 7th SIG-Imaging Spectroscopy Workshop*, Edinburgh, United Kingdom.
- [Rej04] JG Rejas, E Prado, M Jiménez, A Fernández-Renau, JA Gómez, E de Miguel (2004). “Caracterización del sensor hiperespectral AHS para la georreferenciación directa de imágenes a partir de un sistema inercial GPS/IMU”. *Semana Geomática*, Barcelona, España.
- [Rev10] P Reviriego, JA Maestro, B Sanghyeon (2010). “Optimizing scrubbing sequences for advanced computer memories”. *IEEE Transactions on Device and Materials Reliability* 10(2): 192-200.
- [Rew90] R Rew, G Davis (1990). “NetCDF: an interface for scientific data access”. *IEEE Computer Graphics and Applications* 10(4): 76-82.

- [Rey09] LM Reyneri, D del Corso, S Speretta (2009). "A redundant power bus for distributed power management for a modular satellite". *60th International Astronautical Congress*, Daejeon, South Korea.
- [Ric12] JA Richards (2012). "Remote sensing digital image analysis: An introduction". *Ed. Springer*.
- [Ric13a] JA Richards (2013). "Remote sensing digital image analysis. An introduction". *Ed. Springer*.
- [Ric13b] JA Richards (2013). "Correcting and registering images". *Remote Sensing Digital Image Analysis 2013*: 27-77.
- [Rit00] N Ritter, M Ruth, BB Grissom, G Galang, J Haller, G Stephenson, S Covington, T Nagy, J Moyers, J Stickley (2000). "GeoTIFF format specification". *SPOT Image Corp*.
- [Rit97] N Ritter, M Ruth (1997). "The GeoTIFF data interchange standard for raster geographic images". *International Journal of Remote Sensing* 18(7): 1637-1647.
- [Ror12] A Rorhbach (2012). "Airborne sensor technology". *TerraSolid - User Group Meeting*, Levi, Suomi.
- [Ros93] J Rose, A El Gamal, A Sangiovanni-Vincentelli (1993). "Architecture of Field-Programmable Gate Arrays". *Proceedings of the IEEE* 81(7): 1013-1029.
- [Sad11] HFW Sadrozinski, J Wu (2011). "Field Programmable Gate Arrays in scientific research". *Ed. Taylor & Francis*.
- [Sad97] K Sasidhar, L Alkalai, A Chatterjee (1997). "Test strategies for NASAs 3-D stack multichip module space flight computer". *6th International Conference on Multichip Modules*, Denver, USA.
- [Sal02] VV Salomonson, W Barnes, J Xiong, S Kempler, E Masuoka (2002). "An overview of the Earth Observing System MODIS instrument and associated data systems performance". *IEEE International Geoscience and Remote Sensing Symposium*, Toronto, Canada.
- [Sap12] Tomislav Sapic (2012). "Image georeferencing". *GIS in Management Planning*.
- [Sav65] A Savitsky, MJE Golay (1965). "Smoothing and differentiation of data by simplified least square procedures". *Analytical Chemistry* 36: 1627-1639.
- [Sch05] J Schiewe (2005). "Status and future perspective of the application potential digital airborne sensor systems". *International Journal of Applied Earth Observation and Geoinformation* 6(3-4): 215-228.
- [Sch07] RA Schowengerdt (2007). "Remote sensing. Models and methods for image processing". *Ed. Academic Press*.
- [Sch10] H Schmitz (2010). "Application examples: How to use FPGAs in satellite systems". *Actel Corporation*, San Jose, USA.
- [Sch93] KP Scharwz, MA Chapman, MW Cannon, P Gong (1993). "An integrated INS/GPS approach to the georeferencing of remotely sensed data". *Photogrammetry Engineering and Remote Sensing* 59(11): 1667-1674.
- [Sea08] G Seagrave (2008). "SpaceCube: A reconfigurable processing platform for space". *Military Aerospace Programmable Logic Devices*, Annapolis, USA.

- [Sea09] G Seagrave (2009). “Reconfigurable hardware SpaceCube & beyond”.
- [Ser84] J Serra (1984). “Image analysis and mathematical morphology”. *Ed. Academic Press*.
- [Sez04] M Sezgin, B Sankur (2004). “Survey over image thresholding techniques and quantitative performance evaluation”. *Journal of Electronic Imaging* 13(1): 146-165.
- [Sha02] G Shaw, D Manolakis (2002). “Signal processing for hyperspectral image exploitation”. *IEEE Signal Processing Magazine* 19(1): 12-16.
- [Shi09] FY Shih (2009). “Image processing and mathematical morphology: Fundamentals and applications”. *Ed. CRC Press*.
- [Sim07] K Simonson, S Drescher, F Tanner (2005). “A statistics based approach to binary image registration with uncertainty analysis”. *IEEE Pattern Analysis and Machine Intelligence* 29(1): 112-125.
- [Sim08] M Simmons (2008). “Ethernet Theory of Operation”.
- [Simwww] Web oficial del entorno Simulink: <http://www.mathworks.com/products/simulink/> (Último acceso: 28 Julio 2013).
- [Sin10] V Singh (2010). “CISC processor design”. *Processor Design*.
- [Ska04] J Skaloud, P Viret (2004). “GPS/INS integration”. *European Journal of Navigation* 2(4): 40-44.
- [Ska10] T Skauli, TV Haavardsholm, I Kasen, G Arisholm, A Kavara, TO Opsahl, A Skaugen (2010). “An airborne real-time hyperspectral target detection system”. *Algorithms and Technologies for Multispectral, Hyperspectral, and Ultraspectral Imagery*, Orlando, USA.
- [Sma00] A Smailbegovic, JV Taranik, F Kruse (2000). “Importance of spatial and radiometric resolution of AVIRIS data for recognition of mineral endmembers in the Geiger Grade Area, Nevada, USA”. *Ninth JPL Airborne Earth Science Workshop*, Pasadena, USA.
- [Smi99] PC Smits, A Annoni (1999). “Updating land-cover maps by using texture information from very high-resolution space-borne imagery”. *IEEE Transactions on Geoscience and Remote Sensing* 37(3): 1244-1254.
- [Sob03] JS Sobolewski (2003). “Cyclic redundancy check”. *Encyclopedia of Computer Science*.
- [Sob04] JA Sobrino, JC Jiménez-Muñoz, J El-Kharraz, M Gómez, M Romaguera, G Sòria (2004). “Single-channel and two-channel methods for land surface temperature retrieval from DAIS data and its application to the Barrax site”. *International Journal of Remote Sensing* 25(1): 215-230.
- [Sob08] JA Sobrino, JC Jiménez-Muñoz, G. Sòria, M Gómez, A Barella, M Romaguera, M Zaragoza, Y Julien, J Cuenca, M Atitar, V Hidalgo, B Franch, C Mattar, A Ruescas, L Morales, A Gillespie, L Balick, Z Su, F Nerry, L Peres, R Libonati (2008). “Thermal remote sensing in the framework of the SEN2FLEX project: field measurements, airborne data and applications”. *International Journal of Remote Sensing* 29(17-18): 4961-4991.
- [Sob09] JA Sobrino, JC Jiménez-Muñoz, PJ Zarco-Tejada, G Sepulcré-Cantó, E Miguel, G Sòria, M Romaguera, Y Julien, J Cuenca, V Hidalgo, B Franch, C Mattar, L Morales, A Gillespie, D Sabol, L Balick, Z Su, L Jia, A Gieske, W Timmermans, A Oliso, F Nerry, L Guanter, J Moreno, Q Shen (2009). “Thermal remote sensing from Airborne Hyperspectral Scanner data in the framework of the SPARC and SEN2FLEX projects: an overview”. *Hydrology and Earth System Sciences* 13: 2031-2037.

- [Spa07] Space Computer Corporation (2007). “An introduction to hyperspectral imaging technology”.
- [Spe07] S Speretta, LM Reyneri, C Sansoé, M Tranchero, C Passerone, D Del Corso (2007). “Modular architecture for satellites”. *58th International Astronautical Congress*, Hyderabad, India.
- [Sta01] PA Stadter, AA Chacos, RJ Heins, GT Moore, EA Olsen, MS Asher (2001). “Confluence of navigation, communication, and control in distributed spacecraft systems”. *IEEE Aerospace Conference*, Big Sky, USA.
- [Sta12] E Stanton (2012). “IPEX. Maximizing 1U payload potential”. *CubeSat Workshop*, San Luis Obispo, USA.
- [Ste01] T Sterling, DS Katz, L Bergman (2001). “High performance computing systems for autonomous spaceborne missions”. *International Journal of High Performance Computing Applications* 15(3): 282-296.
- [Str09] EV Stryland (2009). “Handbook of optics. Volume II. Design, fabrication and testing. Sources and detectors. Radiometry and Photometry”. *Ed. McGraw-Hill Professional*.
- [Syn12] Synopsis (2012). “FPGA design solutions for military and aerospace applications”.
- [TIA94] Telecommunications Industry Association (1994). “TIA/EIA-422-B”. *Telecommunications Industry Association*.
- [TIA97] Telecommunications Industry Association (1997). “TIA-232-F: Interface between data terminal equipment and data circuit-terminating equipment employing serial binary data interchange”. *Telecommunications Industry Association*.
- [TIB06] TIBCO (2006). “Common Data Format (CDF)”.
- [Tak96] K Takahashi, M Ono (1996). “Extraction of precise attitude by means of image navigation channel data of AVNIR”. *XVIIIth International Society for Photogrammetry and Remote Sensing Congress*, Wien, Austria.
- [Tei97] PM Teillet, K Staenz, DJ Williams (1997). “Effects of spectral, spatial and radiometric characteristics on remote sensing vegetation indices”. *Remote Sensing of Environment* 61(1): 139-149.
- [Terwww] Página web oficial del satélite TerraSAR-X: <http://terrasar-x.dlr.de> (Último acceso: 28 Julio 2013).
- [Tha00] Tharsys (2000). “SPARC SC. Space embedded single board computer for TEMIC ERC32 single chip”.
- [Tho10] DA Thomas (2010). “Causes of catastrophic failure in complex systems”. *An Optical Believe It or Not: Key Lessons Learned II*, San Diego, USA.
- [Til02] JC Tilton (2002). “Image information mining utilizing hierarchical segmentation”. *IEEE International Geoscience and Remote Sensing Symposium* 2(2): 1029-1031.
- [Til03] JC Tilton (2003). “Hierarchical image segmentation”. *Online Journal of Space Communication*: 1-6.
- [Tou02] T Toutin, P Cheng (2002). “Quickbird, a milestone for high-resolution mapping”. *Earth Observation Magazine* 11(4): 14-18.
- [Tou04] T Toutin (2004). “Review article: Geometric processing of remote sensing images: models, algorithms and methods”. *International Journal of Remote Sensing* 25(10): 1893-1924.

- [Tsa82] BM Tsai, CS Gardner (1982). "Remote sensing of sea state using laser altimeters". *Applied Optics* 21(21): 3932-3940.
- [Tuc79] CJ Tucker (1979). "Red and photographic infrared linear combinations for monitoring vegetation". *Remote Sensing of Environment* 8(2): 127-150.
- [Vane93] G Vane, RO Green, TG Chrien, HT Enmark, EG Hansen, WM Porter (1993). "The airborne visible/infrared imaging spectrometer (AVIRIS)". *Remote Sensing of Environment* 44(2-3): 127-143.
- [Ver01] D Verrier (2001). "A unified framework for spacecraft operations".
- [Ver08] J Verrelst, ME Schaepman, B Koetz, M Kneubühler (2008). "Angular sensitivity analysis of vegetation indices derived from CHRIS/PROBA data". *Remote Sensing of Environment* 112(5): 2341-2353.
- [Vin97] RK Vincent (1997). "Fundamentals of geological and environmental remote sensing". *Ed. Prentice Hall*.
- [Vin99] JS Vincent (1999). "Mapping shellfish distribution using hyperspectral remote sensing".
- [Vla04] T Vladimirova, M Sweeting (2004). "System-on-a-chip development for small satellite onboard data handling". *Journal of Aerospace Computing, Information, and Communication* 1.
- [Wan09] YQ Wang (2012). "Active and passive remote sensing". *Remote Sensing of the Environment*.
- [Wan09] YQ Wang (2009). "Geometric rectification of remote sensing images". *Remote Sensing in Natural Resources Mapping*.
- [Wel80] DC Wells, EW Greisen, RH Harten (1980). "FITS: A flexible image transport system". *Astronomy & Astrophysics Supplement Series* 44: 363-370.
- [Wer04] R Werninghaus (2004). "TerraSAR-X mission". *SAR Image Analysis, Modeling, and Techniques VI*, Barcelona, España.
- [Xia08] Z Xiang, Z Jianqi, J Wei, L Delian (2008). "Method for removing sun glint from hyperspectral image". *Acta Optica Sinica* 4.
- [Xil05] Xilinx (2005). "Xilinx University Program Virtex-II Pro Development System. Hardware reference manual".
- [Xil06] Xilinx (2006). "Xilkernel".
- [Xil09a] Xilinx (2009). "Virtex-4 configuration. User guide".
- [Xil09b] Xilinx (2009). "ML505/ML506/ML507 reference design".
- [Xil09c] Xilinx (2009). "Virtex-5 family overview".
- [Xil09d] Xilinx (2009). "XPS UART Lite (v1.01a)".
- [Xil10a] Xilinx (2010). "Virtex-4 family overview".
- [Xil10b] Xilinx (2010). "LogiCORE IP XPS Ethernet Lite Media Access Controller".
- [Xil11a] Xilinx (2011). "Practical use of FPGAs and IP in DO-254 compliant systems".
-

- [Xil11b] Xilinx (2011). “PicoBlaze 8-bit embedded microcontroller user guide for extended Spartan-3 and Virtex-5 FPGAs. Introducing PicoBlaze for Spartan-6, Virtex-6 and 7 series FPGAs”.
- [Xil11c] Xilinx (2011). “ML505/ML506/ML507 evaluation platform”.
- [Xil11d] Xilinx (2011). “ML510 embedded development platform”.
- [Xil11e] Xilinx (2011). “LogiCORE IP Tri-Mode Ethernet MAC v4.5”.
- [Xil11f] Xilinx (2011). “LogiCORE IP CORDIC v4.0”.
- [Xil11g] Xilinx (2011). “LogiCORE IP Multi-Port Memory Controller (MPMC) v6.03.a”.
- [Xil12a] Xilinx (2012). “MicroBlaze processor reference guide”.
- [Xil12b] Xilinx (2012). “Radiation-hardened, space-grade Virtex-5QV family overview”.
- [Xil12c] Xilinx (2012). “LogiCORE IP Floating-Point Operator v6.1. Product guide”.
- [Xio09] X Xiong, K Chiang, J Sun, WL Barnes, B Guenther, VV Salomonson (2009). “NASA EOS Terra and Aqua MODIS on-orbit performance”. *Advances in Space Research* 43(3): 413-422.
- [Xu08] W Xu, J Wang, R Shu, Z He, Y Ma (2008). “Airborne multidimensional integrated remote sensing system”. *XXIth International Society for Photogrammetry and Remote Sensing Congress*, Beijing, China.
- [Yam05] F Yamazaki, Y Yano, M Matsuoka (2005). “Visual damage interpretation of buildings in Bam city using Quickbird images following 2003 Bam, Iran, Earthquake”. *Earthquake Spectra* 21(S1): 329-336.
- [Yan05] M Yang, RE McGrath, M Folk (2010). “HDF5 - A high performance data format for Earth Science”. *21st International Conference on Interactive Information Processing Systems (IIPS) for Meteorology, Oceanography, and Hydrology*, San Diego, USA.
- [Yan06] W Yang, D Huang, B Tan, JC Stroeve, NV Shabanov, Y Knyazikhin, RR Nemani, RB Myneni (2006). “Analysis of leaf area index and fraction of PAR absorbed by vegetation products from the terra MODIS sensor: 2000-2005”. *IEEE Transactions on Geoscience and Remote Sensing* 44(7): 1829-1842.
- [Yan08] FJ Yanovsky (2008). “Recent researches in the field of weather radar at the National Aviation University”. *Microwaves, Radar and Remote Sensing*, Kiev, Ukraine.
- [Zab06] A Zabala, X Pons, R Díaz-Delgado, F García, F Auli-Llinas, J Serra-Sagrasta (2006). “Effects of JPEG and JPEG2000 lossy compression on remote sensing image classification for mapping crops and forest areas”. *IEEE International Geoscience and Remote Sensing Symposium*, Denver, USA.
- [Zar96] M Zargham (1996). “Computer architecture”. *Ed. Prentice Hall*.
- [Zei12] O Zeynali, D Masti, S Gandomkar (2012). “Shielding protection of electronic circuits against radiation effects of space high energy particles”. *Advances in Applied Science Research* 3(1): 446-451.
- [Zen11] F Zeng, GJ Collatz, J Pinzon, A Ivanoff (2011). “Analysis of the relationship between climate and NDVI variability at global scales”. *American Geophysical Union, Fall Meeting*, San Francisco, USA.

- [Zha10] Y Zhang, L Chang, G Yang, H Li (2010). “Reconfigurable-system-on-chip implementation of data processing units for space applications”. *Transactions of Tianjin University* 16(4): 270-274.
- [Zho13] Q Zhou (2013). “Space-borne Platforms and Sensors”. *Remote Sensing and Image Interpretation*.
- [Zit03] B Zitová, J Flusser (2003). “Image registration methods: A survey”. *Image Vision Computing* 21(11): 977-1000.
-

