



TESIS DOCTORAL

**POSICIONAMIENTO LOCAL DE PRECISIÓN
MEDIANTE RECONOCIMIENTO CON DISPOSITIVOS
MÓVILES DE BALIZAS DE LUZ VISIBLE CODIFICADA**

JUAN DIEGO GUTIÉRREZ GALLARDO

MODELIZACIÓN Y EXPERIMENTACIÓN EN CIENCIA Y TECNOLOGÍA

Directores: Fernando Javier Álvarez Franco / Teodor Aguilera Benítez

Esta tesis cuenta con la autorización del director/a y codirector/a de la misma y de la Comisión Académica del programa. Dichas autorizaciones constan en el Servicio de la Escuela Internacional de Doctorado de la Universidad de Extremadura.

2022

A mi familia y amigos, que tanto han tenido que prescindir de mí para que este trabajo saliese adelante.

Resumen

En este trabajo de tesis se propone un sistema de posicionamiento en interiores que utiliza luz visible codificada, lo que permite su identificación mediante dispositivos portables. Para alcanzar este objetivo se han diseñado un módulo emisor construido sobre una luminaria LED comercial doméstica para la transmisión de datos. La baliza resultante utiliza un microcontrolador para modular la señal, así como otros componentes electrónicos, montados en una PCB diseñada para tal fin. Dicho microcontrolador proporciona una interfaz disponible a través de Wi-Fi a través de la que se puede modificar el comportamiento de la baliza una vez desplegada. El sistema de codificación elegido, Manchester, ofrece una luminosidad relativa idéntica para todas las balizas que compartan el mismo espacio, lo que permite aprovechar la infraestructura de iluminación existente sin perturbaciones apreciables por el ojo humano.

El receptor aprovecha las capacidades de los dispositivos portables para realizar las tareas de decodificación y posicionamiento. Para ello se utiliza su cámara, que adquiere muestras en las que la baliza presente proporciona todo lo necesario para realizar la decodificación de los datos transmitidos por la baliza. En cada toma se capturan dos muestras que, combinadas, permiten optimizar el rendimiento del sistema de comunicación de datos, aumentando su alcance. De esas dos tomas, una se puede aprovechar para reconstrucción de la pose en la que se encontraba el receptor cuando realizó la adquisición de datos, logrando así estimar su posición.

El funcionamiento del sistema se ha probado en simulaciones y en entornos reales. El simulador se ha desarrollado como un add-on de código abierto y multiplataforma para Blender. En él es posible configurar las características de la baliza y de la cámara, tomar muestras en habitaciones con diferentes características, lo que permite analizarlas posteriormente y estudiar los resultados obtenidos. Para probar el funcionamiento del sistema se ha desarrollado una aplicación para Android que permite tomar muestras y analizarlas para decodificar los datos emitidos por las balizas y estimar la posición en la que se encontraba al tomar dichas muestras.

El sistema desarrollado logra posicionar con un error medio por debajo de los 9 cm, teniendo un peor rendimiento cuando las balizas no se capturan de forma completa. El sistema de comunicación, por el contrario, no se ve afectado por este problema.

Abstract

This thesis proposes an indoor positioning system using coded visible light and its identification using portable devices. To achieve this goal, an emitter module built on a commercial domestic LED luminaire has been designed for data transmission. The resulting beacon uses a microcontroller to modulate the signal, as well as other electronic components, mounted on a specially designed PCB. Said microcontroller provides an interface available via Wi-Fi through which the behaviour of the beacon, once deployed, can be modified. The chosen encoding system, Manchester, provides identical relative brightness for all beacons sharing the same space, allowing to take advantage of the existing lighting infrastructure without appreciable disturbances to the human eye.

The receiver is a portable device programmed to perform decoding and positioning tasks. For this purpose, its camera is used, acquiring samples in which the beacon is present and providing everything necessary to decode the data transmitted by the beacon. In each shot, two samples are captured, which, combined, allow the performance of the data communication system to be optimized, increasing its range. Of these two samples, one can be used to reconstruct the pose in which the receiver was at the time of data acquisition, thus estimating its position.

The system's operation has been tested in simulations and real environments. The simulator has been developed as an open-source, cross-platform add-on for Blender. It is possible to configure the characteristics of the beacon and the camera and take samples in rooms with different dimensions, which allows to analyze them later and study the results obtained. To test the operation of the system, an Android application has been developed that allows taking samples and analyzing them to decode the data emitted by the beacons and estimate the position in which it was when taking these samples.

The developed system achieves positioning with an average error below the 9 cm, having a worse performance when the beacons are not entirely captured. On the other hand, this problem does not affect the communication system.

Agradecimientos

Por razones que mejor contar en otro momento, en el año 2000 dejé pasar la ocasión de trabajar en la Universidad de Extremadura. Gracias a ese error comenzó un periplo de casi 20 años que me llevó por múltiples empresas privadas. En el año 2018 la oportunidad se volvió a presentar y, esa vez, no se me escapó. Se lo tengo que agradecer a Jorge Morera, que me puso en contacto con el grupo de investigación GISS y al director del grupo, Fernando Javier Álvarez Franco, que me ofreció de nuevo la posibilidad de trabajar para la UEx. “Gracias” se queda corto.

Fernando Javier Álvarez Franco y Teodoro Aguilera Benítez han codirigido esta tesis. No sólo se merecen mi agradecimiento sino un pedacito de cielo por el enorme trabajo que han hecho, revisando lo escrito por un novato de casi 50 años, enseñando nuevos trucos a un perro viejo y reviviendo la pasión por la investigación en un informático. Gracias también a todos los miembros del grupo GISS que, de una manera u otra, han contribuido en la realización de este trabajo.

Antonio Gordillo, Mario Figueira y Enrique Pino: muchísimas gracias por abrirme las puertas de Smart Open Lab. Menos mal que SOL no existía cuando estaba estudiando Informática, porque si hubiese tenido acceso a un laboratorio tan completo, con tanto *enrea* como yo junto, no hubiese acabado nunca mis estudios. Y gracias a todos los miembros de SOL que, con tanta generosidad, me han ayudado.

Cuando, por caprichos del destino, me encontré fuera de la UEx y con una tesis sin completar, Roberto Rodríguez Echeverría, Fernando Sánchez Figueroa y Juan Carlos Preciado me abrieron una vez más sus puertas y me ofrecieron trabajo, dándome la oportunidad de seguir adelante con esta (tardía) vocación de la academia. Fue, precisamente, este Fernando quien me ofreció aquella oportunidad, allá por el comienzo de siglo.

Pilar Bachiller Burgos no se esperaba aparecer en estos agradecimientos, pero aquí está. Porque me ha escuchado y aconsejado cuando lo he necesitado. Y lo mismo puedo decir de Alberto Gómez Mancha. Gracias a los dos.

También me gustaría dar las gracias a todos aquellos en cuyos hombros voy montado: científicos, desarrolladores, etc. Sin su generosidad al compartir sus conocimientos y su trabajo, mi trabajo sería muchísimo más complicado.

Estoy seguro de estar olvidando dar las gracias a muchas personas, pero que sepáis que os lo agradezco. De corazón. Ya seas un conserje que me ha dado unas pilas, un laboral que me ha dejado utilizar tus herramientas o una escalera, una limpiadora que no ha limpiado la sala de pruebas cuando estaba tomando muestras, el equipo directivo de la Escuela Politécnica, que me ha permitido acceder al centro en fines de semana y festivos, los guardas de seguridad, que literalmente me han abierto las puertas del centro, o cualquier otra persona que ha cometido el error de preguntarme por mi tesis y luego ha tenido la santa paciencia de escucharme sin buscar una excusa para largarse, sólo puedo deciros una cosa: gracias, gracias, gracias.

No hace tanto que comencé este viaje que está a punto de concluir. A veces se me olvida todo lo que hemos pasado juntos, Fernando. Gracias por confiar en mí.

Índice general

Resumen	v
Abstract	vii
Agradecimientos	ix
Índice de figuras	xix
Índice de tablas	xxi
Abreviaturas	xxv
1 Introducción	1
1.1 Por qué posicionamiento en interiores	2
1.2 Por qué utilizar paneles LED como emisores	2
1.3 Por qué utilizar dispositivos portables como receptores	4
1.4 Contextualización de la tesis	5
1.5 Estructura de la tesis	6
2 Situación actual y últimos avances	9
2.1 Introducción	9
2.2 Arquitectura	10
2.3 Comunicación con luz visible LED	10
2.3.1 Justificación del auge de esta tecnología	13
2.3.2 Tipos de LED	15
2.3.3 Esquemas de modulación	16
2.3.4 Compensación de la atenuación y el parpadeo	19
2.4 Posicionamiento con luz visible	21
2.4.1 Posicionamiento directo	21
2.4.2 Posicionamiento en dos pasos	22
2.4.3 Análisis de visión	43
2.4.4 Triangulación	47
2.4.5 Trilateración	51
2.4.6 Geometría de vista única	52

2.5	Objetivos	61
3	Comunicación de datos	65
3.1	Introducción	65
3.2	El efecto <i>rolling shutter</i>	65
3.3	Elección del sistema de codificación	68
3.4	Proceso de decodificación	68
3.5	Baliza	73
3.5.1	Prototipos	73
3.5.2	Diseño final	75
3.6	Pruebas realizadas	76
3.6.1	Alcance	78
3.6.2	Cobertura	81
3.6.3	Interferencia	85
3.7	Optimización	88
3.7.1	Elección del formato de imágenes	88
3.7.2	Rotación de bits	88
3.7.3	Toma en ráfagas	93
3.7.4	Doble toma	97
3.7.5	Doble toma + ráfagas	101
3.7.6	Consumo de batería	104
3.8	Conclusiones	106
4	Herramientas software	109
4.1	Introducción	109
4.2	Generador de códigos Manchester	109
4.3	Sistema de control de la baliza	110
4.4	Análisis offline	114
4.4.1	Otras herramientas	117
4.5	Análisis online	118
4.5.1	Decodificación	119
4.5.2	Procesamiento	123
4.5.3	Tiempo de procesamiento	124
4.5.4	Posicionamiento	125
4.6	GetSensorData	126
4.6.1	Cambios realizados	127
4.6.2	Descripción de resultados	135
4.7	Consumo de batería	136

4.8	Simulación	137
4.8.1	Add-on de Blender	142
4.8.2	Instalación	142
4.8.3	Utilización	143
4.8.4	Fidelidad de la simulación	147
4.8.5	Otras características	151
5	Posicionamiento	155
5.1	Introducción	155
5.2	Conceptos previos	155
5.2.1	Modelo de cámara estenopeica	155
5.2.2	Geometría de la formación de imágenes	156
5.2.3	Estimación de la pose	160
5.2.4	El algoritmo SQPnP	162
5.3	Estimación de la posición	168
5.3.1	Toma de muestras	169
5.3.2	Procesamiento de muestras	170
5.3.3	Búsqueda de elementos en la muestra	171
5.3.4	Obtención de la traslación y la rotación de la cámara	173
5.4	Cálculo del FOV de la cámara	177
5.5	Conclusiones	182
6	Resultados experimentales	183
6.1	Pruebas en la simulación	183
6.1.1	Descripción	184
6.1.2	Resultados	186
6.2	Banco de pruebas de tamaño reducido	192
6.2.1	Descripción	192
6.2.2	Resultados	194
6.3	Banco de pruebas extendido	197
6.3.1	Descripción	197
6.3.2	Resultados	202
6.4	Prueba de movimiento	209
6.4.1	Descripción	209
6.4.2	Resultados	209
7	Conclusiones y trabajos futuros	215
7.1	Conclusiones	215
7.2	Trabajos futuros	217
7.2.1	Mejoras en el sistema de comunicación	217

7.2.2	Mejoras en la baliza	218
7.2.3	Mejoras en el simulador	219
7.2.4	Mejoras en el sistema de posicionamiento	220
7.2.5	Mejoras en el receptor	220
7.3	Publicaciones y actividades relacionadas con la tesis	221
7.3.1	Publicaciones en revistas internacionales	221
7.3.2	Publicaciones en congresos internacionales	221
7.3.3	Publicaciones en congresos nacionales	222
	Bibliografía	223

Índice de figuras

1.1	Ancho de banda de cada tecnología diodo emisor de luz (<i>Light-Emitting Diode</i> , LED) [13].	3
1.2	Espectro electromagnético [12].	4
2.1	Arquitectura de los sistema de posicionamiento en interiores (<i>Indoor Positioning System</i> , IPS) según [10].	11
2.2	Diagrama de emisor y receptor del fotófono de Graham Bell [27].	12
2.3	Codificación mediante interrupción (<i>On-Off-Keying</i> , OOK).	17
2.4	Comparación de esquemas de modulación de pulsos [11].	17
2.5	Transmisión de datos en [54].	19
2.6	Modulación por polarización en [56].	19
2.7	Clasificación de las técnicas de posicionamiento mediante luz visible [25].	21
2.8	Posicionamiento directo y en dos pasos [58].	22
2.9	Sistema de posicionamiento con un emisor y varios receptores [63]. . .	24
2.10	Curvas función de distribución acumulativa (<i>cumulative distribution function</i> , CDF) de error de posicionamiento de [64].	25
2.11	Sistema de posicionamiento Epsilon [6].	26
2.12	Pasos en el posicionamiento mediante tiempo de llegada (<i>Time of Arrival</i> , TOA).	28
2.13	Sistema de posicionamiento mediante análisis de tiempo de llegada (<i>Time of Arrival</i> , TOA) [66].	28
2.14	Posicionamiento mediante TDOA [69].	30
2.15	Posicionamiento mediante TDOA [71].	31
2.16	Trilateración frente a multilateración.	31
2.17	Sistema de posicionamiento mediante ángulo de llegada [67].	32
2.18	Diagrama del sistema QADA de A. Arafa <i>et al.</i> [72].	33
2.19	Diagrama del sistema QADA de S. Cincotta <i>et al.</i> [74].	34
2.20	Diagrama del banco de pruebas simulado para el sistema QADA-PLUS [75].	34
2.21	Sistema propuesta por De-La-Llana-Calvo <i>et al.</i> en [76].	36
2.22	Prototipo [78].	38
2.23	Sistema propuesto por [80].	39
2.24	Sistema propuesto por [81].	40

2.25	Sistema propuesto por Vongkulbhisal <i>et al.</i> [84].	42
2.26	Desviación estándar del error de posicionamiento en [85].	44
2.27	Relación entre sistemas de coordenadas para análisis de visión [86]. . .	45
2.28	Triangulación [86].	47
2.29	Diagrama para la simulación de un sistema de posicionamiento basado en triangulación [92].	48
2.30	Diagrama del prototipo de un sistema de posicionamiento basado en triangulación [93].	49
2.31	Diagrama del prototipo de un sistema de posicionamiento basado en triangulación [94].	50
2.32	Esquema de funcionamiento de ALTAIR [95].	51
2.33	Trilateración [10].	53
2.34	Geometría de vista única [90].	54
2.35	Modelo de cámara estenopeica utilizado en Luxapose [3].	54
2.36	Prototipo de Luxapose para pruebas [3].	57
2.37	Banco de pruebas de [98].	58
2.38	Estructura de datos emitida por la baliza de [99].	59
2.39	Mejora en la obtención de los umbrales para la extracción del código emitido por la baliza de [99].	60
2.40	Elementos necesarios para la estimación de la posición del receptor en [99].	60
2.41	Errores medios de posicionamiento en función de la inclinación del receptor en [99].	61
3.1	El efecto <i>rolling shutter</i>	66
3.2	Resultado de ajustar tiempo de exposición y sensibilidad de la cámara. .	67
3.3	Valores del 0 al 3 codificados con dos bits usando Manchester, separador incluido. Se proporciona un ejemplo de cada secuencia [105].	69
3.4	Funcionamiento del receptor.	72
3.5	Huecos en la muestra.	72
3.6	Prototipos.	74
3.7	Prototipo 3.	77
3.8	Diseño de las pruebas de alcance.	78
3.9	Resultados de las pruebas de alcance.	80
3.10	Muestra con código repetido.	81
3.11	Comparación de los resultados de las pruebas de alcance utilizando la cámara trasera y la frontal.	82
3.12	Prueba de alcance sobre una superficie.	83
3.13	Correspondencia entre cobertura y campo visual (<i>Field of Vision</i> , FOV). .	85

3.14 Prueba de interferencia.	86
3.15 Resultados de las pruebas de interferencia.	87
3.16 Resultado del procesamiento de una muestra tomada a 1,2 m.	89
3.17 Resultados de las pruebas de alcance, sin y con rotación de bits.	91
3.18 Éxito en la detección de códigos a 1,5 m, sin y con la rotación de bits activada.	92
3.19 Cobertura sin y con rotación de bits.	93
3.20 Muestra y código de barras a 2,4 m.	94
3.21 Muestra y código de barras a 2,0 m.	94
3.22 Comparación del sistema de ráfaga con el de rotación de bits.	96
3.23 Cobertura del sistema de ráfaga.	97
3.24 Muestras con ceros a ambos lados.	98
3.25 Ejemplo de toma de dos muestras.	99
3.26 Rendimiento con la distancia del algoritmo de toma de dos muestras, comparado con el resto.	101
3.27 Tiempo medio de procesamiento del algoritmo de dos muestras, compa- rado con el resto.	102
3.28 Alcance del algoritmo de toma de dos muestras más ráfaga, comparado con el resto.	103
3.29 Tiempo medio de procesamiento del algoritmo de dos muestras más ráfaga, comparado con el resto.	103
3.30 Impacto de la distancia en el rendimiento del algoritmo de ráfaga con diferente número de muestras.	104
3.31 Rendimiento de la batería.	106
3.32 Estimación del alcance del sistema de comunicación en una habitación.	107
4.1 Ejemplo de utilización de la herramienta para la generación de códigos Manchester.	110
4.2 Valores del 0 al 3 codificados con dos bits usando Manchester, separador incluido. Se proporciona un ejemplo de cada secuencia [105].	111
4.3 Diagrama de flujo del algoritmo propuesto de codificación Manchester.	112
4.4 Selección de la red Wi-Fi.	113
4.5 Panel de control de la baliza.	115
4.6 Archivos generados tras analizar el contenido de una carpeta.	117
4.7 Archivos de depuración generados tras analizar el contenido de una muestra.	117
4.8 Aplicación para la toma de muestras.	120
4.9 Aplicación para la toma de muestras con decodificación nativa.	121
4.10 Efecto de utilizar los botones de la parte superior.	122

4.11 Banco de pruebas para el posicionamiento.	125
4.12 Calibración del banco de pruebas.	126
4.13 Aspecto de la aplicación GetSensorData [22].	128
4.14 Repositorios de GetSensorData.	129
4.15 Consejos de Android Studio.	131
4.16 Documentación interna.	132
4.17 Organización en Android Studio.	132
4.18 Clase DataSensor.	133
4.19 Clase AccelerometerDataSensor.	134
4.20 Aspecto de las diferentes versiones de la aplicación GetSensorData.	135
4.21 Nuevas preferencias de GetSensorData.	136
4.22 Detalle de la previsualización de cámara en GetSensorData.	137
4.23 Aplicación para tomar los valores de referencia de consumo de la batería.	138
4.24 Blender en funcionamiento.	139
4.25 Metadatos contenidos en cada muestra.	141
4.26 Add-on de Blender en PyCharm.	143
4.27 Operadores de Blender.	144
4.28 Paneles de Blender.	145
4.29 Mundo real y simulación.	149
4.30 Imagen de depuración.	151
4.31 Fidelidad de la simulación.	152
4.32 Add-on Control Panel.	153
5.1 Modelo de cámara estenopeica.	156
5.2 Sistemas de coordenadas presentes al tomar una muestra.	157
5.3 Proyección del punto P en el sensor de la cámara y sistemas de referencia [161].	159
5.4 Búsqueda de un mínimo local a partir de las soluciones del problema de la matriz ortogonal más próxima para el caso en el que $\mathbb{R} \in SO(2)$ [171].	165
5.5 Diagrama de funcionamiento del sistema de estimación de la posición.	168
5.6 Ejemplo de toma de dos muestras.	169
5.7 Procesamiento de muestras.	170
5.8 Detalle de las esquinas de las balizas utilizadas en la simulación y en el mundo real.	171
5.9 Comparación de la detección de esquinas.	173
5.10 Cámara en una esquina del FOV completo.	179
5.11 Cámara en una esquina del FOV ajustado a la baliza.	180
5.12 División del campo visual (<i>Field of Vision</i> , FOV) en celdas.	180
5.13 Los cuatro campo visual (<i>Field of Vision</i> , FOV) calculados.	182

6.1	Configuración de la simulación para las pruebas, con el panel de ajustes de la cámara abierto.	184
6.2	Paneles para la configuración de movimientos de cámara en el simulador.	186
6.3	Resultados de posicionamiento en todo el campo visual (<i>Field of Vision</i> , FOV) con diferentes rotaciones alrededor del eje X.	187
6.4	Muestras tomadas con rotación X 10° en las coordenadas Y -0,35 m, -0,40 m, -0,45 m y -0,50 m, cuando la baliza es parcialmente visible.	188
6.5	Resultados de posicionamiento en todo el campo visual (<i>Field of Vision</i> , FOV) con diferentes rotaciones alrededor del eje Z.	190
6.6	Captura parcial de la baliza al rotar alrededor del eje Z.	191
6.7	Banco de pruebas reducido.	193
6.8	Estimación de la posición.	195
6.9	Mapa de calor de la estimación de la posición.	196
6.10	Sala i3 antes de montar el banco de pruebas.	198
6.11	Simulación del banco de pruebas en la sala i3 antes del montaje.	199
6.12	Montaje final del banco de pruebas en la sala i3.	200
6.13	Proceso de toma de muestras.	202
6.14	Disponibilidad del sistema de comunicación de datos.	203
6.15	Muestras tomadas en las esquinas y el origen de coordenadas del banco de pruebas en la sala i3.	205
6.16	Cobertura del sistema de posicionamiento en la sala i3.	206
6.17	Estimación de las posiciones en la sala i3.	207
6.18	Toma de muestras en movimiento.	210
6.19	Estimación de las posiciones durante movimiento en espiral en la sala i3.	211
6.20	Estimación de las posiciones durante movimiento en zigzag en la sala i3.	212

Índice de tablas

2.1 Comparación de tecnologías diodo emisor de luz (<i>Light-Emitting Diode</i> , LED) [12].	16
2.2 Comparación de esquemas de modulación [12].	19
3.1 Parejas de bits diferentes para cuatro bits.	79
5.1 Primera parte del algoritmo SQPnP [171].	166
5.2 Segunda parte del algoritmo SQPnP [171].	167
6.1 Resultados de posicionamiento para diferentes rotaciones alrededor del eje X, en $10^{-3} m$	189
6.2 Resultados de posicionamiento para diferentes rotaciones alrededor del eje Z, en $10^{-3} m$	191
6.3 Resultados de posicionamiento en el banco de pruebas reducido, en $10^{-3} m$	197
6.4 Código emitido por cada baliza.	201
6.5 Porcentajes de disponibilidad en el banco de pruebas extendido para cada baliza y en total.	204
6.6 Resultados de posicionamiento en la sala i3.	208
6.7 Resultados de posicionamiento en las zonas centrales de en la sala i3, en metros.	208

Abreviaturas

- ALM** algoritmo Levenberg–Marquardt (*Levenberg–Marquardt Algorithm*)
- AOA** ángulo de llegada (*Angle of Arrival*)
- BFSK** modulación binaria por desplazamiento de frecuencia (*Binary Frequency-Shift Keying*)
- BLE** Bluetooth de baja energía (*Bluetooth Low Energy*)
- BMC** código bifásico de marca (*Biphase Mark Code*)
- CDF** función de distribución acumulativa (*cumulative distribution function*)
- CDK** clave de desplazamiento de color (*Color Shift Keying*)
- CIS** sensor de imagen de semiconductor complementario de óxido metálico (*Complementary Metal-Oxide-Semiconductor Image Sensor*)
- CMOS** semiconductor complementario de óxido metálico (*Complementary Metal-Oxide-Semiconductor*)
- CSIC** Consejo Superior de Investigaciones Científicas
- CSV** valores separados por comas (*comma-separated values*)
- DLT** transformación lineal directa (*Direct Linear Transformation*)
- Exif** formato de archivo de imagen intercambiable (*Exchangeable image file format*)
- FDM** acceso múltiple por división de frecuencia (*Frequency-Division Multiplexing*)
- FOV** campo visual (*Field of Vision*)
- FPGA** matriz de puertas lógicas programable por campo (*Field Programmable Gate Arrays*)
- GISS** Grupo de Investigación en Sistemas Sensoriales
- GNSS** Sistema Global de Navegación por Satélite (*Global Navigation Satellite System*)
- GPS** Sistema de Posicionamiento Global (*Global Positioning System*)

GSSK clave de desplazamiento de espacio generalizada (*Generalised Space Shift Keying*)

IDE entorno de desarrollo integrado (*Integrated Development Environment*)

IoT internet de las cosas (*Internet of Things*)

IPIN Congreso Internacional de Posicionamiento y Navegación en Interiores (*International Conference on Indoor Positioning and Navigation*)

IPS sistema de posicionamiento en interiores (*Indoor Positioning System*)

IR radiación infrarroja (*Infrared Radiation*)

ISI interferencia entre símbolos (*Intersymbol Interference*)

LBS servicio basado en la ubicación (*Location-Based Service*)

LED diodo emisor de luz (*Light-Emitting Diode*)

Li-Fi Light Fidelity

MPPM modulación por posición de múltiples pulsos (*Multipulse Pulse Position Modulation*)

NLQP programa cuadrático no lineal (*Non-Linear Quadratic Program*)

OFDM multiplexación por división de frecuencias ortogonales (*Orthogonal Frequency Division Multiplexing*)

OLED diodo orgánico emisor de luz (*Organic Light-Emitting Diode*)

OOK codificación mediante interrupción (*On-Off-Keying*)

OPPM modulación por posición de pulsos superpuestos (*Overlapping Pulse Position Modulation*)

PAPR relación pico a promedio (*Peak-to-Average Power Ratio*)

PC ordenador personal (*Personal Computer*)

pc-LED LED convertido de fósforo (*Phosphor Converted LED*)

PCB placa de circuito impreso (*Printed Circuit Board*)

PD fotodiodo (*Photodiode*)

PLED diodo de polímeros emisor de luz (*Polymer Light-Emitting Diode*)

PPM modulación por posición de pulsos (*Pulse Position Modulation*)

PSD detector sensible a la posición (*Position Sensitive Detector*)

PWM modulación por ancho de pulsos (*Pulse Width Modulation*)

QADA cuadrante de fotodiodos con apertura angular diversa (*Quadrant Photodiode Angular Diversity Aperture*)

RF radiofrecuencia (*Radio Frequency*)

RFID identificación por radiofrecuencia (*Radio Frequency Identification*)

RGB rojo, verde, azul (*Red, Green, Blue*)

RLL longitud limitada (*Run Length Limited*)

ROM memoria de solo lectura (*Read-Only Memory*)

RSS intensidad de la señal recibida (*Received Signal Strength*)

SNR relación señal-ruido (*Signal-to-Noise Ratio*)

SOL Smart Open Lab

SSID identificador de grupo de servicios (*Service Set Identifier*)

SSL iluminación de estado sólido (*Solid-State Lighting*)

TDOA diferencia de tiempo de llegada (*Time Difference of Arrival*)

TOA tiempo de llegada (*Time of Arrival*)

UEx Universidad de Extremadura

UI interfaz de usuario (*User Interface*)

VLC comunicación mediante luz visible (*Visual Light Communications*)

VLP posicionamiento mediante luz visible (*Visual Light Positioning*)

VPPM modulación por posición variable de pulsos (*Variable Pulse Position Modulation*)

YAML YAML no es un lenguaje de marcado (*YAML Ain't Markup Language*)

Introducción

Como ocurre con todas las tecnologías que terminan siendo ampliamente utilizadas, ahora damos por hecho poder utilizar el Sistema de Posicionamiento Global (*Global Positioning System*, **GPS**) de un dispositivo portable (teléfono inteligente o tableta) para llegar a destino sano y salvo [1] sin tener que bajar la ventanilla para preguntar a nadie.

Esta introducción no estaría completa sin una mención al principal problema que presenta **GPS** (algo que también afecta al Sistema Global de Navegación por Satélite (*Global Navigation Satellite System*, **GNSS**)): la imposibilidad de ser tan precisos en interiores como lo son exteriores [2]. Las causas se resumen en que la señal de los satélites no se recibe adecuadamente dentro de un edificio, debido a que la estructura del mismo interfiere en la señal, desviándola, atenuándola o, incluso, impidiendo su recepción. A esto se debe añadir la necesidad de tener tres satélites (o cuatro, si se precisa también la altitud) en el campo de visión.

Desarrollar un sistema para interiores con las mismas ventajas que **GNSS** ofrece en exteriores es, en el momento de escribir estas líneas, uno de los principales objetivos de multitud de investigadores, tanto en el campo académico como en el empresarial. De él se beneficiarían los usuarios del sistema, que serían capaces de orientarse sin dificultad dentro de edificios sin necesidad de conocerlos previamente. De igual forma, las organizaciones encargadas de desplegar dicho sistema podrían utilizarlo para poder hacer llegar a potenciales clientes a su destino, quizá facilitando así las ventas de sus productos. Incluso sería posible analizar los patrones de comportamiento de dichos clientes, y mejorar la disposición del espacio para conseguir las mejoras que se consideren necesarias.

Este trabajo de tesis pretende desarrollar un sistema complementario al **GPS**, capaz de posicionar en interiores con una precisión adecuada. Para lograrlo se apoya en dos tecnologías tan presentes en nuestro día a día que resulta sencillo pasarlas por alto: las luminarias diodo emisor de luz (*Light-Emitting Diode*, **LED**) y los dispositivos portables. Es imprescindible ser consciente de los condicionantes derivados de tomar esta decisión, que suponen una autoimpuesta limitación. Por una parte, el emisor tendrá unas características que dependerán de la tecnología de iluminación seleccionada. Por otra, no será posible diseñar un receptor particular para el caso

que ocupa a este trabajo de tesis, pudiendo aprovechar únicamente los sensores disponibles en la mayoría de dispositivos portables.

1.1. Por qué posicionamiento en interiores

Como se menciona en [3], logran un posicionamiento preciso en interiores permitiría la utilización de servicios basados en la ubicación (*Location-Based Services*, **LBSs**) en múltiples sectores. Por ejemplo, los pequeños comercios, los supermercados y los centros comerciales podrían mejorar el flujo de potenciales clientes por sus instalaciones, permitiéndoles completar compras que, de otra manera, no se realizarían por no poder localizar los productos que buscasen, aumentando así sus beneficios fruto de inversiones en publicidad [4]. De hecho, la implantación de **LBSs** es una de las razones por las que se esperaba que la demanda de sistemas de posicionamiento en interiores creciese hasta los 5 mil millones en 2018 [5]. Sin embargo, a pesar de la esta previsión, el posicionamiento en interiores sigue siendo un gran desafío y no existe una solución precisa que ofrezca posicionamiento y orientación en dispositivos portables sin modificaciones [6].

1.2. Por qué utilizar paneles LED como emisores

La iluminación artificial es uno de los pilares en los que se apoya la civilización actual. Ser capaces de realizar actividades con independencia de la presencia de la luz del sol resulta imprescindible, tanto en el ámbito personal como en el profesional. En 2011 existían 14 000 millones de bombillas en todo el planeta [7], sin importar el tipo de tecnología empleada.

La tecnología **LED** es una de las variantes de iluminación de estado sólido (*Solid-State Lighting*, **SSL**) [8] de las que también forman parte otras tecnologías similares como diodo orgánico emisor de luz (*Organic Light-Emitting Diode*, **OLED**) y diodo de polímeros emisor de luz (*Polymer Light-Emitting Diode*, **PLED**). En la Unión Europea, a partir del 1 de septiembre de 2018, las bombillas halógenas deben sustituirse por alternativas **LED**, más eficientes [9].

Este tipo de iluminación tiene múltiples ventajas cuando se compara con el resto de tecnologías a las que sustituye [10], [11]:

- **gran ancho de banda:** los fotoreceptores actuales pueden alcanzar decenas de MHz [12]. Según el profesor Harald Haas, resultaría posible llegar a los 100 Gbps [13], aunque para ello sería necesario realizar cambios en la tecnología de iluminación empleada. Como muestra la figura 1.1, cada tecnología tiene un ancho de banda teórico máximo.

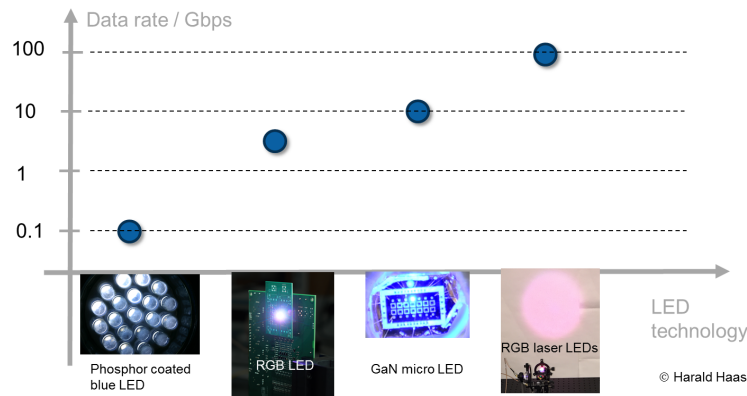


Figura 1.1. Ancho de banda de cada tecnología LED [13].

Existen esfuerzos para normalizar los sistemas de comunicación basados en luz visible. En concreto, IEEE 802.15.7 [14] (aún en desarrollo) promete hasta 96 Mbps.

- **ecológica:** la tecnología LED tiene muy bajo consumo si la comparamos con la de las tecnologías tradicionales. Según estudios de la Unión Europea [15], si se utilizase exclusivamente LED para la iluminación, el ahorro energético anual llegaría a 9,4 TWh. Esto equivale al consumo eléctrico de Portugal en 5 años. Además, genera menos calor residual.
- **sin interferencias:** por su propia naturaleza, la utilización de la tecnología LED como sistema de transmisión de datos no provoca interferencias electromagnéticas ni de radio.
- **cobertura:** dentro de la zona iluminada la cobertura es prácticamente idéntica debido a que este tipo de luz sigue un modelo lambertiano [16], lo que permite que el comportamiento del sistema sea más predecible.
- **larga duración:** una luminaria LED puede alcanzar una vida útil de hasta 50 000 horas de funcionamiento ininterrumpido [17]. Es necesario tener en cuenta que se trata de límites teóricos, ya que probarlo que llevaría 5,7 años de funcionamiento continuado, 24 horas al día, 7 días a la semana. Teniendo en cuenta la velocidad con la que esta tecnología evoluciona, para el momento

en el que la prueba concluyese ya existirían luminarias con mejores vidas útiles que las que se estuviesen probando.

- disponibilidad:** uno de los pilares del posicionamiento en interiores basado en luz visible es la posibilidad de transmitir datos utilizando el canal luminoso. El espectro de radiofrecuencias disponibles está disminuyendo a medida que la popularidad de las redes inalámbricas va en aumento [12]. En la figura 1.2 se pueden ver varias bandas de dicho espectro, desde las ondas de radio hasta los rayos gamma, pasando por los espectros de luz (visible e invisible para el ojo humano).

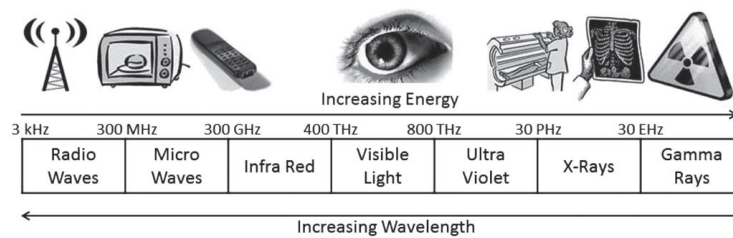


Figura 1.2. Espectro electromagnético [12].

Las ondas de radio ocupan las bandas de 3 kHz a 300 GHz de frecuencia, mientras que la luz visible está entre los 400 y los 800 THz. Con la tasa actual de crecimiento en la utilización del espectro de radiofrecuencia, el momento en el que no haya espacio disponible para la utilización de nuevos canales está cerca [7]. Por fortuna, el espacio disponible en el espectro de luz visible es suficiente para satisfacer las necesidades de comunicación previsible.

Existen muchas más ventajas, como la rentabilidad económica de emisores y receptores o su eficiencia, por sólo mencionar algunas. Sin embargo, a todas ellas, que por sí solas serían suficientes para justificar el cambio, es posible añadir una más, muy importante en el ámbito de esta tesis: resulta posible convertir una luminaria LED en una baliza con una modificación mínima. En este trabajo de tesis se describe de forma pormenorizada este proceso.

1.3. Por qué utilizar dispositivos portables como receptores

Existe un dicho en el mundo de la fotografía: “la mejor cámara es la que va contigo”. Podría atribuirse a Chase Jarvis, que incluso ha publicado un libro con el mismo

título [18] en el que explora las posibilidades de utilizar la cámara de un dispositivo portable para capturar la realidad social del mundo que nos rodea.

Aunque parezca no haber relación entre dicha afirmación y la razón por la que en esta tesis se ha decidido utilizar la cámara de un dispositivo portable como receptor, sí que la hay. Entre los objetivos de este trabajo está desarrollar un sistema de posicionamiento en interiores con luz visible que cualquiera con un dispositivo portable pueda aprovechar, sin necesidad de utilizarlo en conjunción con dispositivos diseñados específicamente para tal fin.

Se estima que, al terminar el año 2021, el parque de smartphones alcanzó los 5200 millones de unidades en funcionamiento en todo el mundo, con una tasa de penetración del 66 % [19]. Por otra parte, el parque de tabletas llegó a 1280 millones de unidades en funcionamiento en todo el mundo, con una tasa de penetración del 20 % [20]. Por lo tanto, un sistema de posicionamiento en interiores que se apoye en el uso de dispositivos portables contaría con una ventaja fundamental: la inmensa mayoría de los que podrían beneficiarse de él tendría un receptor en potencia a su alcance.

1.4. Contextualización de la tesis

El desarrollo de esta tesis está vinculado con el proyecto de transferencia tecnológica Limus (0246-LIMUS-4-E), financiado por el programa Interreg-POCTEP de la Unión Europea. El autor de esta tesis participa como técnico de apoyo a la investigación en dicho proyecto.

Según la web del Museo Arqueológico de Badajoz [21], el proyecto Limus, cuyo título es “Transferencia tecnológica para el libre acceso a museos y espacios arqueológicos en Extremadura y Alentejo basado en dispositivos móviles”, está financiado y promovido por el Programa Operativo Cooperación Transfronteriza España Portugal (POCTEP), Interreg 2014-2020, de la Unión Europea. Tiene como objetivo la transferencia tecnológica de material desarrollado en la Universidad de Extremadura (UEx) por el Grupo de Investigación en Sistemas Sensoriales (GISS) de la Escuela Superior de Ingenierías Industriales a museos, espacios arqueológicos y PYMES. Su finalidad es facilitar el acceso de los ciudadanos a toda la información que necesiten para acceder al conocimiento de los espacios culturales participantes utilizando sus propios dispositivos móviles y con diversos grados de profundidad según requiera cada público y usuario individual. Aparte de los objetivos de información, comunicación y transferencia tecnológica, subyace al proyecto la intención de ser útil en la

promoción del turismo cultural, transfronterizo e interfronterizo, en Extremadura y el Alentejo.

El proyecto se desarrolla entre 2017 y 2020. Los socios participantes en el proyecto son la Secretaría General de Cultura de la Junta de Extremadura, el Grupo de Investigación en Sistemas Sensoriales (GISS), la Cámara Municipal de Évora y el Campo Arqueológico de Mértola.

Aparte de las labores administrativas y de gestión del proyecto, la labor de los distintos socios será: por parte de la UEx, aportar el marco tecnológico y adaptarlo a las necesidades sobre el terreno; y los restantes aportarán por una parte la información sobre los lugares donde se aplicará tal tecnología, mapeando todos los recursos museísticos y arqueológicos de la zona Euroace, aplicando la tecnología en los museos y espacios arqueológicos que administran, definiendo y desarrollando los contenidos en distintos niveles de la información que se impartirá a los usuarios finales.

La tecnología y la aplicación se experimentarán en diversos puntos: Museo Arqueológico Provincial de Badajoz, Campo Arqueológico de Mértola y los diferentes monumentos y museos de la Cámara Municipal de Évora.

1.5. Estructura de la tesis

Cada uno de los capítulos que conforman esta tesis cubre una fase diferente del desarrollo del mismo. El **capítulo 1**, que sirve como introducción, sienta las bases del trabajo.

Tras él, en el **capítulo 2** se analiza la situación actual en el campo que afecta a este trabajo de tesis, atendiendo a sus dos aspectos fundamentales: la comunicación y el posicionamiento. Aunque se presentan técnicas de posicionamiento genéricas, se hace especial énfasis en su aplicación en sistemas basados en luz visible.

El camino recorrido para llegar al sistema de comunicación mediante luz visible utilizado en esta tesis se presenta en el **capítulo 3**, donde se describen tanto el emisor como el receptor. En concreto, el emisor está construido utilizando luminarias **LED** comerciales, modificadas para poder realizar la transmisión de datos. Por su parte, el receptor es un dispositivo móvil con un software específicamente diseñado para decodificar las señales emitidas por el panel **LED**. Una de las aportaciones de esta tesis se centra en la utilización de técnicas tanto en el emisor como en el receptor para mejorar el alcance del sistema, así como su ancho de banda efectivo.

El **capítulo 4** describe las herramientas software desarrolladas en el ámbito de esta tesis para la comunicación de datos, la simulación del sistema de posicionamiento y el posicionamiento en sí mismo. Además, se incluyen las modificaciones realizadas en la aplicación Android GetSensorData [22], parte de GetSensorDataSuite [23]. Dicho módulo permite tomar de muestras de los diferentes sensores del dispositivo portable que, posteriormente, serán procesadas.

Aprovechando el conocimiento adquirido tras el estudio de la situación actual y disponiendo de emisores y receptores comunicación mediante luz visible (*Visual Light Communications, VLC*), se está en disposición de desarrollar un sistema de posicionamiento. El **capítulo 5** describe pormenorizadamente el esquema de posicionamiento desarrollado en este trabajo de tesis, así como la aportación del mismo en este campo. Una vez se dispone de un emisor y un receptor suficientemente desarrollados llega el momento de comprobar que se dispone de un sistema de posicionamiento en interiores basado en luz visible. También se describe el banco de pruebas diseñadas para el emisor, el receptor y la combinación de ambos, así como a analizar los resultados obtenidos.

El **capítulo 6** se dedica a estudiar las fuerzas y debilidades del sistema de posicionamiento desarrollado, así como a extraer las principales conclusiones del trabajo. Además, establece las nuevas líneas de investigación derivadas de dichos resultados.

Situación actual y últimos avances

2.1. Introducción

El objetivo principal de este trabajo de tesis es el desarrollo de un sistema de posicionamiento local para dispositivos portables mediante codificación de luz visible. Resulta imprescindible enfrentarse a esa tarea conociendo el estado en el que se encuentra la investigación en lo que a sistemas de posicionamiento con luz visible se refiere: así podremos caminar a hombros de gigantes en pos de nuestra meta.

Sin embargo, como veremos a medida que desgranemos los diferentes avances conseguidos en este campo, dependiendo de los objetivos que el sistema de posicionamiento pretenda conseguir, éste dependerá en gran medida de su capacidad para transmitir una serie de datos a través del canal de luz visible como, por ejemplo, el identificador de cada luminaria, su geometría, su posición dentro de la sala, etcétera. La utilización de esquemas de comunicación con luz visible permite conseguir varios objetivos de forma simultánea: iluminar una estancia, transmitir datos, y posicionar. Este capítulo está dedicado, precisamente, a describir la situación actual y los últimos avances del posicionamiento en interiores utilizando luz visible con transmisores de tipo **LED**.

Por lo tanto, este capítulo cuenta con dos partes bien diferenciadas. En la primera (ver la sección 2.3) veremos qué técnicas se están utilizando para la transmisión de datos mediante luz visible. En la segunda (ver la sección 2.4) nos centraremos en el posicionamiento en interiores mediante luz visible. A lo largo de ambas partes recorreremos aquellos trabajos y resultados más relevantes en lo que concierne a nuestros objetivos, dado que serán los que utilicemos como principales referencias. Es necesario destacar que, aunque ambos campos de conocimiento son importantes, el posicionamiento constituye el objetivo primordial de este trabajo.

Para terminar, este capítulo recoge una declaración de objetivos (ver la sección 2.5) para el resto del trabajo de tesis. El lector interesado en ampliar conocimiento sobre

las técnicas de posicionamiento en interiores puede hacerlo en el segundo capítulo de [24].

2.2. Arquitectura

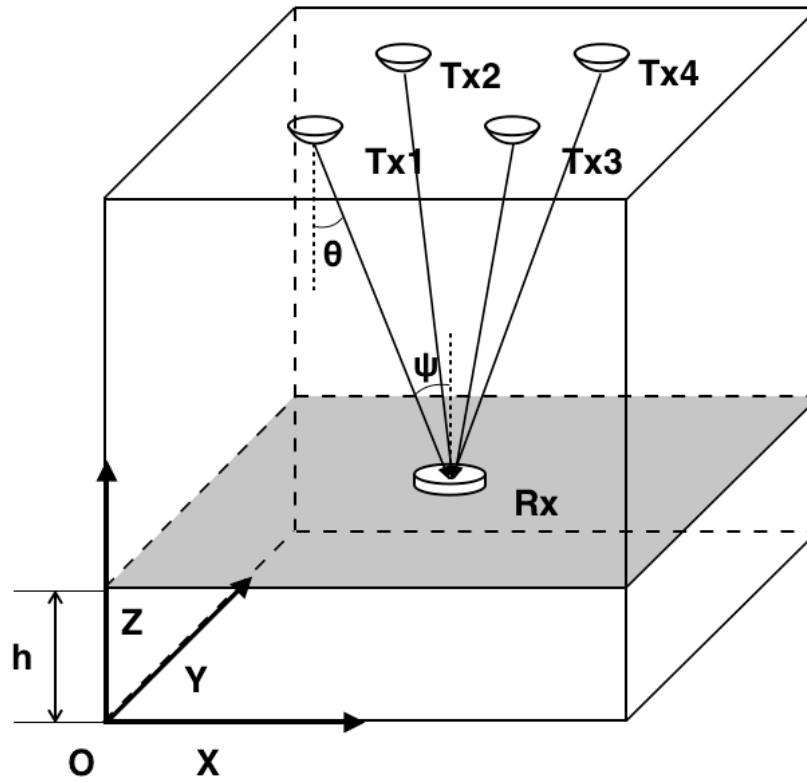
La mayoría de los sistemas que veremos utilizan fotodiodo (*Photodiode*, **PD**) como receptor o, al menos, algún elemento hardware que se incorpora a un dispositivo portable. De hecho, dada la evolución observada en el desarrollo de este tipo de sistema de posicionamiento en interiores (*Indoor Positioning System*, **IPS**), [10] propone una clasificación mucho más concisa que [25], dividiéndolos en dos grandes grupos: los que usan **PD** como receptor y los que usan cámaras. Así, la arquitectura de los **IPS** puede describirse desde un punto de vista físico o lógico, como muestra la figura 2.1.

La arquitectura física de los **IPS** (ver la figura 2.1a) muestra cada uno de los elementos que forman el sistema. Podemos pensar que el espacio en el que sistema está desplegado tiene forma de cubo. En el techo de la estancia se encuentran una serie de emisores que cumplen una doble función. Por una parte, iluminan la estancia; por otra, son las balizas en las que el **IPS** se apoya. Bajo ellas, un receptor situado a cierta altura. El objetivo del sistema es deducir la posición del receptor dentro de la estancia a partir de los datos obtenidos de los emisores.

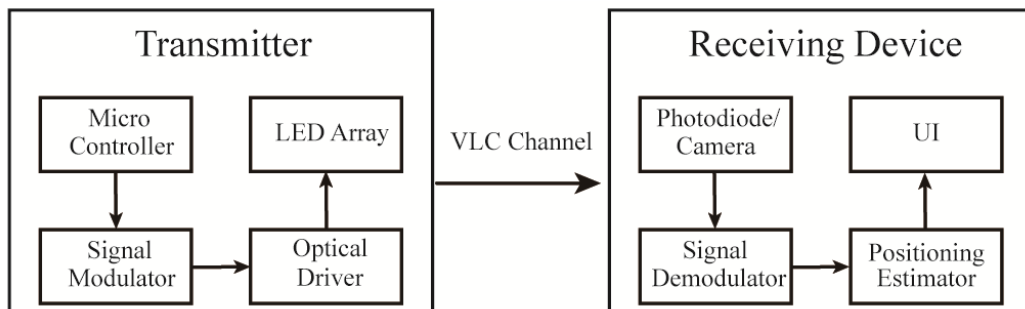
La arquitectura lógica de los **IPS** (ver la figura 2.1b) muestra el diseño genérico de emisor y receptor. En el emisor, un microcontrolador programado adecuadamente modula la señal y, actuando sobre un *driver* óptico, es capaz de emitir ciertos datos desde una o más luminarias **LED** a través del canal de **VLC**. Por su parte, el receptor, equipado con **PD** o cámara, demodula la señal recibida y, a partir de ella, es capaz de estimar su posición dentro de la estancia, mostrándola al usuario a través de la interfaz de usuario (*User Interface*, **UI**) correspondiente.

2.3. Comunicación con luz visible LED

En marzo de 2011 tuvo lugar un ciclo de conferencias TEDGlobal en Escocia, Reino Unido. Fue entonces cuando el profesor de la universidad de Edimburgo Harald Haas introdujo el término Light Fidelity (**Li-Fi**) [7].



(a) Arquitectura física.



(b) Arquitectura lógica.

Figura 2.1. Arquitectura de los IPS según [10].

La similitud de éste con Wi-Fi¹ no es casual, ya que ambos términos se refieren a un conjunto de tecnologías cuyo objetivo es la transmisión de datos de forma inalámbrica. Mientras que un sistema Li-Fi utiliza la luz visible para la transmisión de datos a alta velocidad, uno Wi-Fi utiliza radiofrecuencia.

Aunque aprovechar la luz visible para la transmisión de datos puede parecer una idea novedosa, el primer intento de realizar algo así del que tenemos constancia ocurrió a finales del siglo XIX. El fotófono, cuyo diagrama puede verse en la figura 2.2, fruto de una colaboración entre Alexander Graham Bell y Charles Sumner Tainter en 1880, era capaz de transmitir la voz mediante la propagación de la luz. Con un alcance de unos 200 m, Graham Bell lo consideraba su mayor logro.

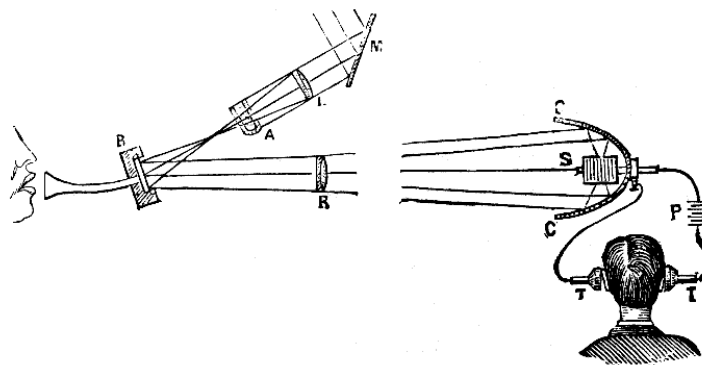


Figura 2.2. Diagrama de emisor y receptor del fotófono de Graham Bell [27].

Los esfuerzos para utilizar la luz visible como método de transmisión no cesaron, a pesar de la imposibilidad de cambiar rápidamente la intensidad del emisor para transmitir datos. Sirva como ejemplo el proyecto Fiat Lux [28], de 1998, que logró transmitir datos a 22 bps utilizando tubos fluorescentes como transceptores sin que el ojo humano percibiese cambio alguno. La llegada de la tecnología LED, capaz de llevar a cabo cambios de intensidad a alta frecuencia (hasta 1 MHz según [29], entre 1 y 300 MHz dependiendo del tipo según [12]), es la pieza que nos ha permitido completar el puzzle de la comunicación mediante luz visible que empezó hace más de un siglo.

El objetivo de este trabajo de tesis es el desarrollo de un sistema de posicionamiento basado en luz visible con transmisores de tipo LED, pero una de las columnas que lo sustentan es la capacidad de transmitir datos a través del medio. Para empezar, las balizas utilizadas podrían identificarse, con lo que el receptor las podría diferenciar. Pero podríamos ir más allá e incluir datos que ayuden en el posicionamiento como,

¹Curiosamente, el término Wi-Fi no tiene su origen en Wireless Fidelity sino en su similitud con Hi-Fi (sonido de alta calidad) [26].

por ejemplo, la sala en la que se encuentran, de qué edificio se trata, coordenadas, altura o geometría, por sólo mencionar algunos.

En un entorno de interior (como, por ejemplo, una de las salas de una exposición), la función principal de las luminarias **LED** es proporcionar iluminación de calidad y con eficiencia energética. Por lo tanto, cualquier esquema de comunicación de datos que se apoye en el sistema de iluminación debe pasar desapercibido para sus potenciales usuarios.

Ahora bien, cabe preguntarse cuál es la situación actual en este campo, cuáles son los últimos avances logrados y, sobre todo y dadas las premisas de nuestro trabajo, a qué límites debemos enfrentarnos.

2.3.1. Justificación del auge de esta tecnología

Las principales razones por las que el interés en la comunicación mediante luz visible está en aumento son las siguientes [12]:

- **Disponibilidad del espectro RF:** la disponibilidad del espectro de radiofrecuencia está disminuyendo como consecuencia del constante aumento de la demanda. **VLC** ofrece una alternativa con un espectro 10 000 veces mayor (mientras que las ondas de radio tienen unafrecuencia comprendida entre 3 kHz y 300 GHz, la de la luz visible está entre 400 y 780 THz, aproximadamente [30]). Aunque al empezar esta década ya existían 1,4 millones de torretas de telefonía móvil y 5000 millones de teléfonos móviles, también tenemos 14 000 millones de bombillas [7], emisores de datos en potencia.
- **Crisis de capacidad:** la previsión de que el consumo de datos móviles entre 2013 y 2018 se multiplicaría por 6,3 [31] en Europa y Estados Unidos parece haber alcanzado techo, con una tendencia a la baja durante el periodo 2019 - 2024 [32]. Sin embargo, las operadoras móviles están llegando al límite de su capacidad [33]. El uso de **VLC** supondría un alivio considerable de esa carga, con un ancho de banda considerablemente mayor, como se comenta en el punto anterior.
- **Interferencias:** **VLC** es un sistema de comunicación seguro por definición y no provoca ninguna interferencia con los sistemas de radiofrecuencia (*Radio Frequency*, **RF**) existentes. Esto lo hace ideal para su uso en aplicaciones sanitarias, industriales y aeroespaciales.

- **Seguridad:** mientras que las ondas **RF** pueden atravesar las paredes y resulta posible interceptarlas, la luz no puede abandonar un espacio confinado por paredes opacas, lo que permite definir zonas seguras utilizando **VLC**.
- **Reutilización del espacio:** resulta posible la coexistencia de diferentes enlaces sin interferencias gracias a la gran direccionalidad y confinamiento de la luz visible [34], lo que permite una mayor densidad de datos.
- **Salud:** en lo que a condiciones de iluminación se refiere, la luz visible no supone riesgo alguno para la salud. Al contrario que la radiación infrarroja (*Infrared Radiation*, **IR**), la luz visible cumple con las regulaciones de seguridad relacionadas con la vista y la piel [35]. Además, los tubos fluorescentes contienen mercurio, que puede contaminar el entorno, por lo que tienen que ser tratados de forma especial para su reciclaje, algo que no ocurre con las luminarias **LED**.
- **Eficiencia energética:** la tecnología **LED**, por sus características, se considera *verde* [36]. Si todas las fuentes luminosas se convirtiesen a esta tecnología, el consumo eléctrico global descendería un 50%, las emisiones de CO_2 se reducirían en 10 gigatoneladas y el consumo de crudo en 962 millones de barriles [37].
- **Red eléctrica inteligente:** existen ventajas inherentes en la combinación de utilizar la línea eléctrica como canal de datos y, a la vez, utilizar la iluminación **LED** como punto de enlace. Esta combinación es equivalente a la pareja Wi-Fi/Ethernet que se utiliza hoy en día [38].
- **Fácil incorporación en la infraestructura existente:** resulta sencillo adaptar la infraestructura de iluminación existente para que pueda utilizarse para la transmisión de datos, sólo con añadir unos componentes relativamente simples y asequibles [39].
- **Bajo coste:** otra de las ventajas de los dispositivos **VLC** es su bajo coste, comparado con el resto. Existen módulos para la transmisión de datos utilizando **RF** con un radio de 10 m para la transmisión de hasta 1 Mbps por unos 5 euros. Por contra, los enlaces **VLC** pueden transmitir hasta a 4 Mbps en cortas distancias por 1 euro por módulo [40].

2.3.2. Tipos de LED

Es posible clasificar los LED de diferentes maneras: por el tipo de color que emiten, por su aplicación o por su coste, entre otras. Aunque todos estos aspectos son relevantes dentro de este trabajo de tesis, resulta más interesante conocer el potencial que cada tecnología tiene para la transmisión de datos. Así, podríamos diferenciar los siguientes tipos de LED dependiendo de la tecnología utilizada para su fabricación:

- **pc-LED:** para poder emitir luz blanca, la tecnología LED debe utilizar diferentes métodos. Como se menciona en [12], un LED convertido de fósforo (*Phosphor Converted LED*, **pc-LED**) utiliza un único chip azul de InGaN para excitar el recubrimiento de fósforo. La mezcla de colores produce blanco, en lugar del azul original [41]. Este tipo de LED resulta energéticamente muy eficiente, proporcionando 130 lm/W y un ancho de banda entre 3 y 5 MHz.
- **Multichip LED:** a diferencia de **pc-LED**, esta tecnología combina tres LEDs de diferentes colores, rojo, verde, azul (*Red, Green, Blue, RGB*) que, combinados, emiten luz blanca. Esta tecnología permite controlar el color emitido, si así se desea, cambiando la intensidad de cada uno de los chips. Tanto si se emite luz blanca como si se busca algún color, la tecnología multichip es más cara, más compleja y con un ancho de banda menor que la **pc-LED**.
- **OLED:** la tecnología de OLED utiliza una capa orgánica, capaz de emitir luz, situada entre dos conductores, uno negativo y otro positivo [42]. Cuando la corriente circula entre las capas externas, la interna emite luz en respuesta a la corriente. En la actualidad, OLED se emplea principalmente para pantallas debido a su capacidad de apagar aquellos píxeles que se correspondan con el color negro puro, algo no posible con otras tecnologías que combinan cristal líquido y retroiluminación LED. Por contra, su vida útil es más limitada que las tecnologías inorgánicas.
- **microLED:** por sus características (cada píxel puede estar en el rango de 14 a $84\mu\text{m}$), el microLED es capaz de ofrecer comunicación en paralelo de alta densidad. No en vano es la tecnología con mayor ancho de banda de las presentadas (superando los 300 MHz). Un estudio conjunto de la Universidad de Strathclyde, junto con investigadores de las universidades de Oxford, Edinburgh (Harald Haas entre ellos), Cambridge y St. Andrews busca conseguir densidades de comunicación ultraparalela con LEDs basados en GaN [43].

En la tabla 2.1 se encuentran resumidas las características más importantes de las cuatro tecnologías LED que se acaban de comentar.

Tabla 2.1. Comparación de tecnologías LED [12].

Parámetro	pc-LED	Multichip LED	OLED	microLED
Ancho de banda	3-5 MHz	10-20 MHz	≤ 1 MHz	≥ 300 MHz
Eficiencia	130 lm/W	65 lm/W	45 lm/W	-
Coste	Bajo	Alto	Muy bajo	Alto
Complejidad	Baja	Moderada	Alta	Muy alta
Aplicación	Iluminación	Iluminación	Pantallas	Biosensores

La tecnología de iluminación LED está en camino de convertirse en la dominante en el mundo entero, lo que fomenta la investigación de nuevas tecnologías de producción y funcionamiento que disminuyan los costes de fabricación y, a la vez, aumenten su vida útil y sus prestaciones. El futuro de la iluminación puede llegar a través de tecnologías tan prometedoras como la *bioled*, basada en proteínas luminiscentes [44].

En lo que respecta a este trabajo de tesis, es más que probable que la tecnología pc-LED sea más que suficiente para los objetivos que se pretenden conseguir, gracias a su bajo coste y simplicidad, su eficiencia y su ancho de banda.

2.3.3. Esquemas de modulación

Los esquemas de modulación más utilizados en comunicación mediante luz visible son:

- **OOK:** por su baja complejidad, la codificación mediante interrupción (*On-Off-Keying*, OOK) es, sin duda, el más común de todos los esquemas de modulación utilizados en VLC. En él, el LED se encuentra apagado para transmitir un 0, y encendido para transmitir un 1, como muestra la figura 2.3. Si nos centramos en la tasa de transmisión, existen trabajos que se han acercado a los 500 Mbps con un único LED [45], o superado los 600 Mbps utilizando ecualización en emisor y transmisor con una técnica *doubinaria* [46]. Pero la mayor tasa de transmisión se consigue combinando varios LED que, individualmente, permiten transmisiones a 375 Mbps y de forma agregada pueden llegar a 1,5 Gbps [47].
- **Modulación de pulsos:** en los esquemas de modulación por pulsos, la portadora sin modular suele consistir en una serie de pulsos emitidos de forma regular

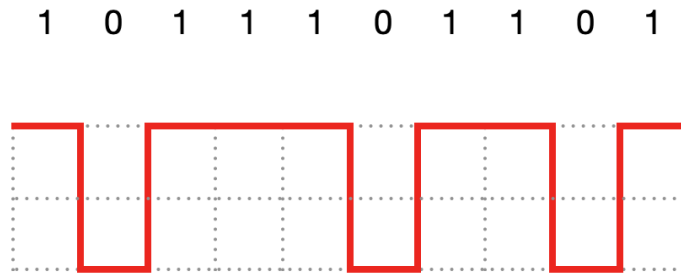


Figura 2.3. Codificación mediante interrupción (*On-Off-Keying*, OOK).

y continuada. Los datos a transmitir se incorporan a la portadora cambiando alguna de sus características como la amplitud, el ancho, la duración, el instante o la forma del pulso [48]. En la figura 2.4 se puede ver una comparación entre diferentes esquemas de modulación por pulsos.

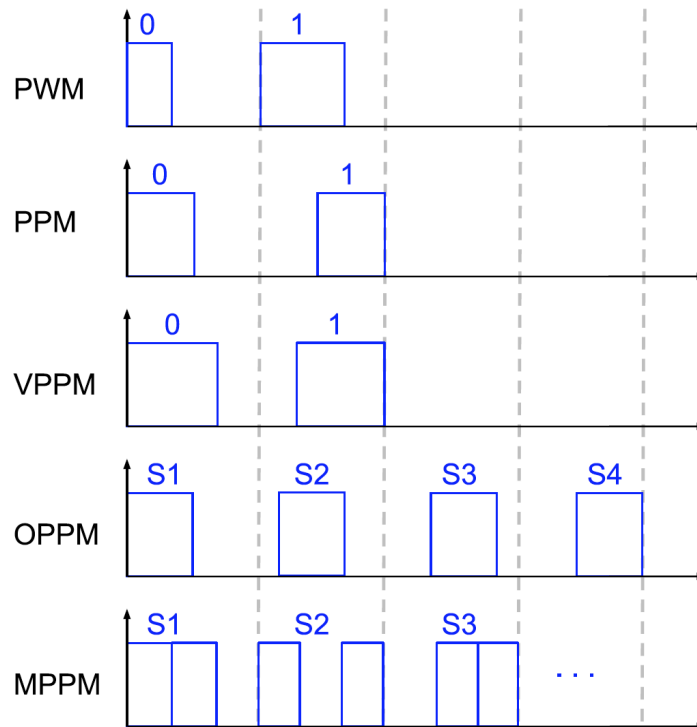


Figura 2.4. Comparación de esquemas de modulación de pulsos [11].

En modulación por ancho de pulsos (*Pulse Width Modulation*, PWM), es el ancho de la señal emitida el que determina su valor, no la presencia o ausencia de señal. Así, el menor ancho se corresponde con un 0 y el mayor ancho con un 1. Este esquema no consigue la misma eficacia en la utilización del ancho de banda que OOK, por ejemplo. En [49], la tasa de transferencia lograda

es de 4,8 Kbps. Por su parte, en el esquema de modulación por posición de pulsos (*Pulse Position Modulation*, **PPM**) es la posición dentro de cada pulso la que determina si un cierto valor es 0 ó 1. En [50] se consigue mejorar el rendimiento del esquema de modulación en situaciones adversas haciendo que la tasa de transferencia sea adaptativa. Es posible definir la modulación por posición variable de pulsos (*Variable Pulse Position Modulation*, **VPPM**) como la combinación de **PWM** y **PPM**, donde tanto el ancho del pulso como su posición aportan información sobre el valor emitido. Otra variante de **PPM** es la modulación por posición de pulsos superpuestos (*Overlapping Pulse Position Modulation*, **OPPM**), que permite la transmisión de más de un símbolo durante cada pulso. Por último, la modulación por posición de múltiples pulsos (*Multipulse Pulse Position Modulation*, **MPPM**) también permite la emisión de más de un símbolo durante cada pulso, con la diferencia de que estos no tienen por qué ser contiguos. Como se demuestra en [51], **MPPM** es más eficiente que **OPPM**.

- **GSSK**: tal y como se explica en [52], la modulación por clave de desplazamiento de espacio generalizada (*Generalised Space Shift Keying*, **GSSK**) es un subconjunto de modulación espacial con mayor eficacia espectral que **OOK** y **PPM** [53], capaz de alcanzar los 4 bits/s/Hz. Está pensada para la emisión *multiled*, con varios transmisores activos de manera simultánea. Aunque su complejidad es baja, es necesario que exista cierta sincronización entre los emisores y el receptor, lo que complica el diseño con relación a **OOK**.
- **OFDM**: el esquema de modulación basado en multiplexación por división de frecuencias ortogonales (*Orthogonal Frequency Division Multiplexing*, **OFDM**) se utiliza para transmitir a alta velocidad flujos de datos en serie. Para ello, el flujo de datos en serie se divide en varios subflujos de datos que se emiten a menor velocidad [54]. Gracias a la transmisión en paralelo se mitigan varios problemas derivados de la transmisión en serie como, por ejemplo, la interferencia entre símbolos (*Intersymbol Interference*, **ISI**) o la necesidad de ecualizadores complejos. El ancho de banda del canal se divide entre varios subcanales de frecuencia ortogonales que se utilizarán para la emisión de los subflujos de datos. En la figura 2.5 se puede ver el esquema del prototipo utilizado para la transmisión de datos en [54]. Aunque **OFDM** es mucho más complejo que el resto de esquemas comentados hasta ahora, es mucho más eficaz que ellos.

En la tabla 2.2 se encuentran resumidas las características más importantes de los cuatro esquemas de modulación que se acaban de comentar.

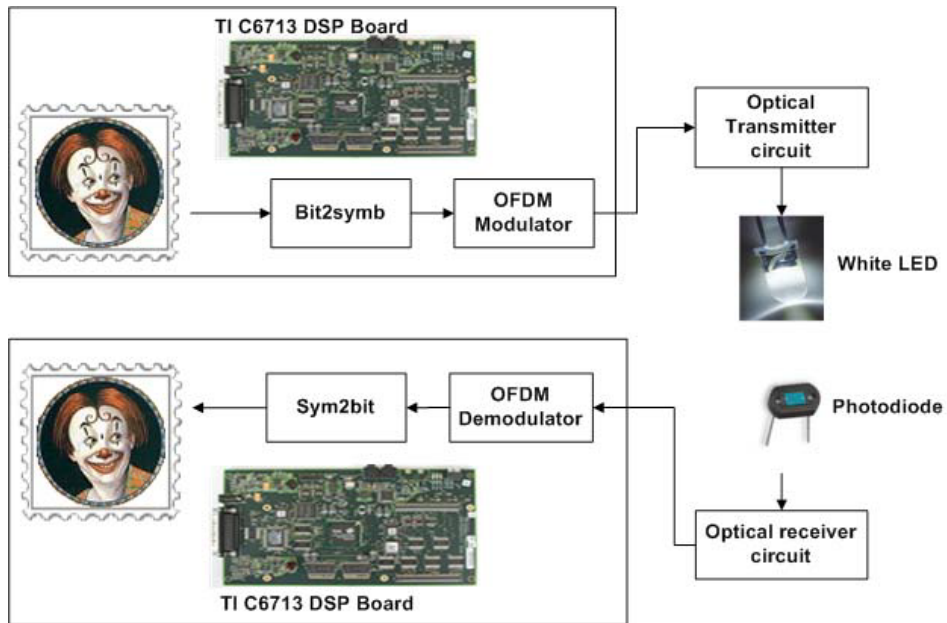
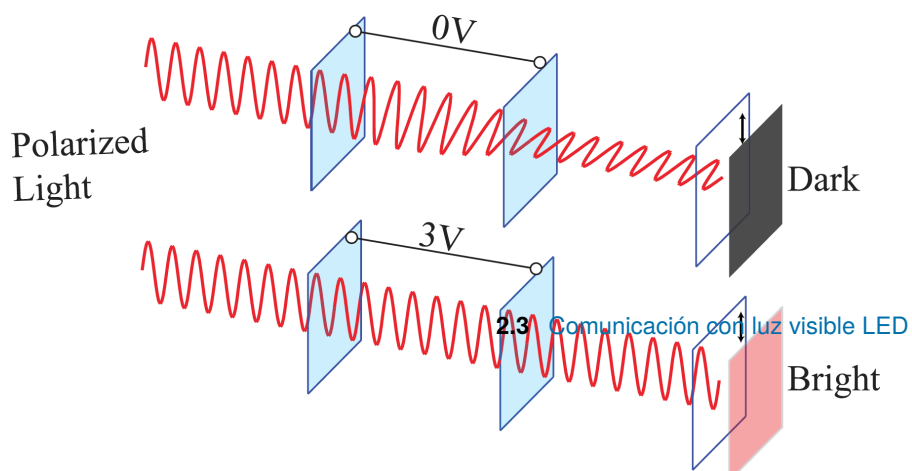


Figura 2.5. Transmisión de datos en [54].

Tabla 2.2. Comparación de esquemas de modulación [12].

Esquema	Eficiencia espectral	Eficiencia energética	Complejidad
OOK	Alta	Baja	Baja
PPM	Baja	Alta	Moderada
GSSK	Alta	Baja	Baja
OFDM	Alta	Moderada	Alta

Aunque sólo hemos visto cuatro esquemas de modulación, existen muchos otros que utilizan diferentes aspectos observables de la señal. Por ejemplo, en [55] se utiliza clave de desplazamiento de color (*Color Shift Keying, CDK*) para emitir diferentes códigos que dependen de la combinación de colores RGB utilizada en el emisor. El receptor filtra los diferentes componentes cromáticos de la señal recibida para poder demodularla. Otra interesante aproximación al problema de la transmisión de datos utilizando luz visible se presenta en [56], donde la polarización de la luz se aprovecha para codificar bits, como se muestra en la figura 2.6. Gracias al cristal líquido y dependiendo del voltaje aplicado, el ángulo de polarización de la luz puede modificarse, lo que permite la transmisión de datos.



de comunicación de datos a la infraestructura de iluminación existente cuando ésta es utilizada por personas. ¿Provoca algún parpadeo perceptible? ¿Es la luminosidad relativa de las balizas distinta dependiendo de los datos transmitidos? El parpadeo (*flicker* en inglés) se puede producir si la luminosidad de la fuente luminosa cambia entre apagado y encendido a una frecuencia que el ojo humano puede percibir. Por otra parte, la diferencia relativa de luminosidad entre balizas (*dimming* en inglés) puede darse si la diferencia entre el número de transiciones entre los estados apagado/encendido es significativamente diferente en entre las balizas.

El problema de la atenuación es fácil de visualizar. Simplificando mucho el sistema, digamos que se está utilizando una cabecera de 5 bits con valor 1, y utilizando un esquema de codificación **OOK** con 4 bits. Emitir el código 0 significaría mantener la luminaria encendida durante 5 ciclos de reloj, y apagada durante 4:

```
11111 0000
```

Ahora bien, si el código emitido fuera el 15, la luminaria pasaría por los siguientes estados:

```
11111 1111
```

En el primer caso, la luminaria estaría encendida el 55,55 % del tiempo, mientras que en el segundo caso lo estaría el 100 % del tiempo. La luminosidad de la primera baliza con respecto a la segunda sería más baja, y dicha diferencia sería apreciable a simple vista.

Una posible solución a este problema pasa por realizar algún tipo de conversión en el código binario que mantenga el número de ceros y unos similar. El método más popular en los sistemas de comunicación con luz visible es la codificación Manchester [57], un tipo de codificación de longitud limitada (*Run Length Limited*, **RLL**). En esta codificación, el mismo estado no se mantiene durante más de dos ciclos, de manera que el código 0, en lugar de una secuencia continua de tantos ceros como bits se utilicen, sería 11001100 si utilizásemos 4 bits en el código binario original, que pasarían a ser 8 en la versión Manchester del mismo. El 15 (1111) se codificaría como 10101010. Tanto el 0 como el 15 tienen el mismo número de ceros y unos en codificación Manchester. Añadiendo una cabecera, la aportación luminosa de dos balizas emitiendo ambos códigos sería la misma.

Resolver estos problemas es tan importante que la norma IEEE 802.15.7, que trata de regular la comunicación con luz visible, tiene una sección dedicada a las posibles soluciones que se están estudiando [14].

2.4. Posicionamiento con luz visible

Según [25], la estimación de la posición del receptor en un IPS, esté basado en RF o en VLC, se realiza intercambiando señales entre nodos con posiciones conocidas (las marcas de referencia) y nodos cuya ubicación debe estimarse (los nodos objetivo). En el caso particular de un IPS con luz visible, las marcas de referencia serían las luminarias y los nodos objetivo, los receptores.

Los IPS basados en luz visible se pueden clasificar de múltiples formas. En [25], por ejemplo, se dividen en aquellos capaces de realizar el posicionamiento directamente, como consecuencia de su observación del entorno, y aquellos que necesitan dos pasos para conseguir el mismo resultado: la observación del entorno y una serie de operaciones posteriores que tendrán como resultado el posicionamiento del receptor. Este último grupo, a su vez, puede clasificarse en otros dos grupos en función de su funcionamiento: los estimadores de parámetros y los estimadores de posición. Esta clasificación se puede ver en la figura 2.7.

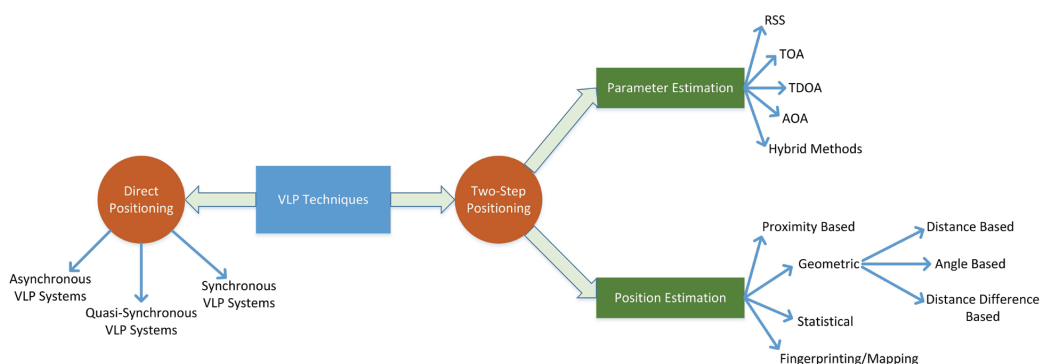


Figura 2.7. Clasificación de las técnicas de posicionamiento mediante luz visible [25].

En apartados posteriores seguiremos esta clasificación para explorar las principales técnicas de posicionamiento con luz visible. Conocerlas nos servirá para entender el estado actual de estos sistemas. Además, estas técnicas no son exclusivas del posicionamiento con luz visible, sino que también se pueden aplicar al resto de IPS.

2.4.1. Posicionamiento directo

En el posicionamiento directo, la estimación de la ubicación del receptor se realiza en un único paso, utilizando para ello todas las señales recibidas, sin pasos intermedios para extraer parámetros dependientes de la ubicación, como muestra la figura 2.8.

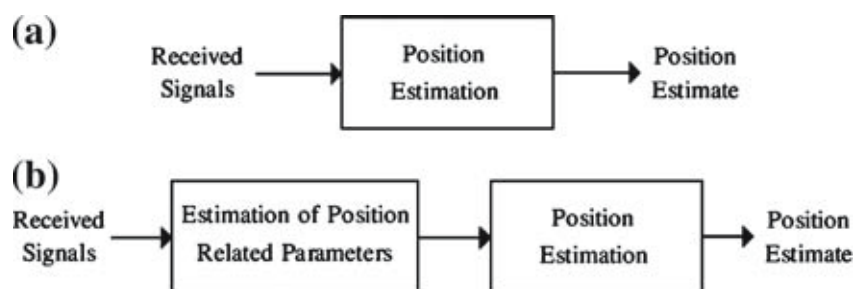


Figura 2.8. Posicionamiento directo y en dos pasos [58].

Sin embargo, esta técnica aún no está plenamente difundida para el posicionamiento en interiores mediante luz visible. En [59] sólo se muestran resultados numéricos fruto de una investigación teórica. Por otra parte, en [60] sí se muestran resultados experimentales. Las dimensiones de la habitación de pruebas eran $2,0 \times 0,8 \times 1,6 \text{ m}^3$. El emisor consistía en cinco luminarias LED, mientras que el receptor era un PD. Tanto el emisor como el receptor estaban controlados por su propia matriz de puertas lógicas programable por campo (*Field Programmable Gate Arrays*, FPGA). En estas pruebas, el error medio de posicionamiento en dos dimensiones fue de 3,18 cm, con el 80 % de los errores por debajo de los 5 cm. En este último trabajo se destaca que las principales ventajas del posicionamiento directo propuesto son sus mejores relación señal-ruido (*Signal-to-Noise Ratio*, SNR) y relación pico a promedio (*Peak-to-Average Power Ratio*, PAPR).

2.4.2. Posicionamiento en dos pasos

El posicionamiento en dos pasos puede clasificarse en dos grandes categorías: aquellos que realizan una estimación de parámetros y los que realizan una estimación de la posición. Los estimadores de parámetros extraen valores a partir de las señales recibidas, utilizándolos para estimar la posición del receptor. Por su parte, los estimadores de posición intentan determinar la posición de los emisores y, a partir de dichos valores, extrapolan la posición del receptor. Dentro del primer grupo se encuentran cinco técnicas ampliamente utilizadas tanto en los IPSs basados en luz como en los IPSs en general:

- **RSS:** utiliza la intensidad de la señal recibida como medida para estimar la posición del receptor.
- **TOA:** el observable, en este caso, es el tiempo que la señal tarda en llegar desde el emisor al receptor.

- **TDOA:** el tiempo también se utiliza para estimar la posición del receptor, aunque en este caso se trata de una diferencia temporal y no del tiempo absoluto.
- **AOA:** este sistema utiliza el ángulo de llegada de la señal percibido por el receptor para estimar su posición.
- **Métodos híbridos:** en lugar de utilizar una única característica de la señal para estimar la ubicación del receptor, estos sistemas utilizan varias características combinadas con el objetivo de mejorar la precisión del sistema.

Veamos en qué consiste cada una de ellas y qué resultados se están consiguiendo mediante su utilización en los **IPSs** basados en luz.

RSS

En el ámbito de las telecomunicaciones, la intensidad de la señal recibida (*Received Signal Strength*, **RSS**) mide la potencia presente al recibir una señal de radio [61]. Si buscamos el equivalente en el posicionamiento utilizando luz visible, la intensidad luminosa es la medida de potencia: a mayor distancia entre emisor y receptor, menor intensidad. Obtener esta medida utilizando únicamente un fotorreceptor es relativamente sencillo [10].

Las luminarias **LED** son fuentes luminosas que no irradian energía directamente, sino que lo hacen por difusión. Como se explica en [10], sus características hacen que este tipo de fuentes luminosas se consideren lambertianas [62]. Por lo tanto, la intensidad luminosa percibida por el receptor es directamente proporcional al coseno del ángulo entre la dirección de incidencia de la luz y la normal de la superficie. Esto podría describirse como muestra la ecuación (2.1):

$$P_i(\theta, \psi) = \frac{(m_i + 1) A_R P_{T_i}}{2\pi} \cdot \frac{\cos^{m_i}(\theta) \cos^M(\psi) T_s(\psi) g(\psi)}{D_i^2} \quad (2.1)$$

donde D_i es la distancia entre el led i y el receptor, A_R es el área efectiva del **PD** del receptor, P_{T_i} es la potencia óptica de la fuente luminosa, θ es el ángulo de radiación del emisor, ψ es el ángulo de incidencia en el receptor, $T_s(\psi)$ es la ganancia del filtro óptico y $g(\psi)$ es la ganancia del concentrador óptico situado frente al detector. M y m representan el orden lambertiano del receptor y el emisor, respectivamente en la ecuación (2.2):

$$m = -\frac{\ln 2}{\ln \left(\cos \theta_{\frac{1}{2}} \right)}$$

$$M = -\frac{\ln 2}{\ln \left(\cos \psi_{\frac{1}{2}} \right)}$$
(2.2)

siendo $\theta_{\frac{1}{2}}$ y $\psi_{\frac{1}{2}}$ los semiángulos a media potencia de emisor y receptor. Si el receptor no utiliza lente alguna se cumple que $T_s(\psi) = g(\psi) = 1$.

Este modelo describe con fiabilidad las características teóricas de propagación de las señales ópticas. Sin embargo, la realidad es que dichas señales pueden llegar al receptor directamente y a través de reflejos, por lo que no existe un único método para determinar la distancia o dirección del emisor a partir de las señales recibidas.

La forma de enfrentarse al problema en [63] es radicalmente distinta a la convencional, como podrá comprobarse a lo largo de este capítulo. En lugar de utilizar varios emisores y un único receptor, este esquema se invierte, teniendo un único emisor y tres receptores (PDs), como puede verse en la figura 2.9.

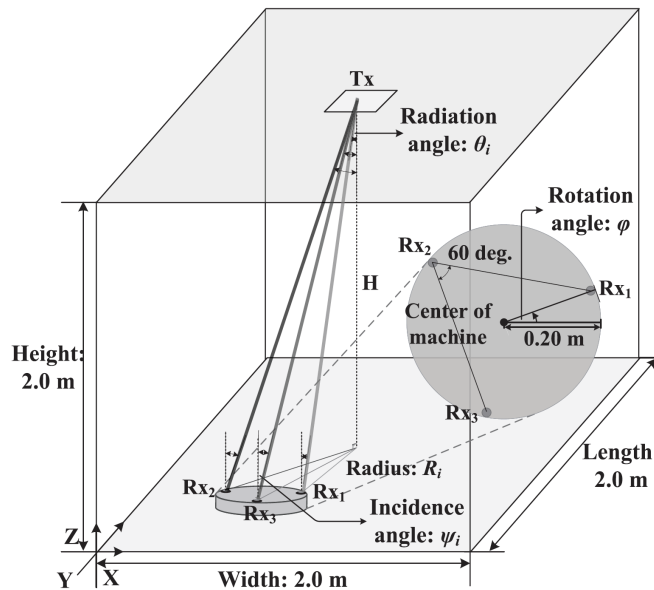


Figura 2.9. Sistema de posicionamiento con un emisor y varios receptores [63].

Los receptores ópticos se utilizan para determinar la intensidad de la señal recibida y, a partir de ese valor, aprovechar las relaciones geométricas de los diferentes elementos del sistema para determinar la posición del receptor con un margen de error de 1,5 cm, siempre y cuando el receptor se encuentre dentro de una zona circular simétrica dentro de la sala de pruebas.

Por otra parte, en [64] se presenta un sistema que, en simulaciones, permite posicionar con un margen de confianza del 95 % y una precisión de 5,9 cm. Para ello propone el uso de un único receptor y tres emisores que transmiten utilizando el protocolo ALOHA con OOK para poder compartir el canal sin interferencias. Se tiene en cuenta el parpadeo producido por dicha modulación, por lo que sólo se realiza el 12,5 % del tiempo, para minimizar su influencia en la iluminación de la sala. En la figura 2.10 se pueden ver las curvas de función de distribución acumulativa (*cumulative distribution function*, CDF) del error de posicionamiento con iluminación solar directa e indirecta.

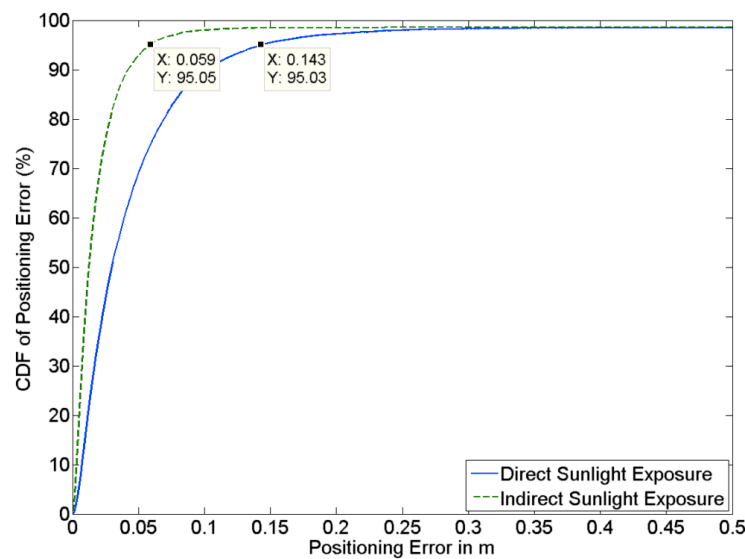
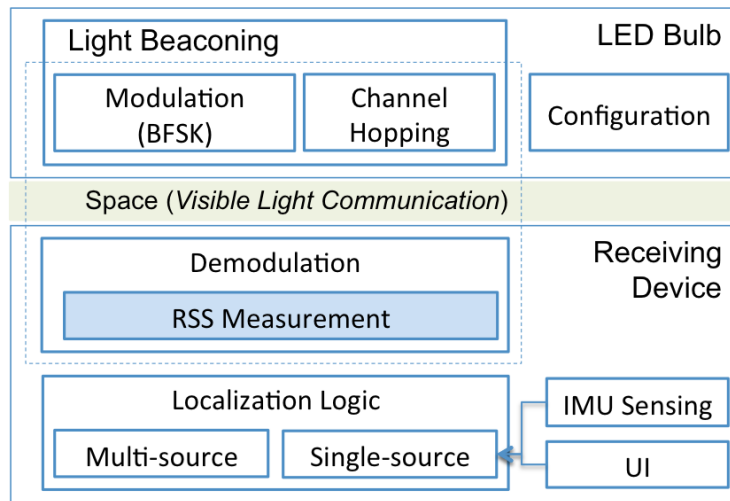
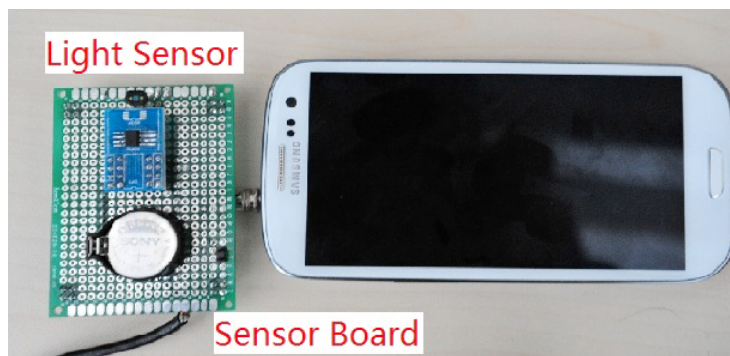


Figura 2.10. Curvas CDF de error de posicionamiento de [64].

Sin embargo, posiblemente el más completo de los sistemas desarrollados utilizando RSS sea Epsilon [6]. En él se aprovecha la trilateración (ver la sección 2.4.5) para determinar la posición del receptor. La arquitectura de Epsilon combina múltiples técnicas en un único sistema, como puede verse en la figura 2.11a. Se utiliza modulación binaria por desplazamiento de frecuencia (*Binary Frequency-Shift Keying*, BFSK) con salto de canal para compartir el medio sin que sea necesaria coordinación entre los diferentes emisores. El receptor es un smartphone modificado con un módulo que contiene lo necesario para determinar el valor de la RSS de cada una de las luminarias. Dicho módulo, que puede verse en la figura 2.11b, contiene un PD que es el encargado de convertir la señal para que pueda procesarse, extrayendo de ella tanto la intensidad recibida (para poder estimar la distancia y poder así trilaterar) y decodificando los datos enviados por cada luminaria que se encuentre en su campo de detección y poder así diferenciarlas.



(a) Arquitectura.



(b) Receptor.

Figura 2.11. Sistema de posicionamiento Epsilon [6].

Para evaluar el rendimiento de Epsilon se prepararon tres diferentes bancos de pruebas, correspondientes a tres entornos comunes: una sala de conferencias, una habitación y un pasillo. En cada uno de ellos se dispusieron cinco LEDs. Las áreas de cada entorno eran $5 \times 8 \text{ m}^2$, $2 \times 12 \text{ m}^2$ y $3,5 \times 6,5 \text{ m}^2$, respectivamente. En cada zona se establecieron 60 posiciones, desde la que se realizaron múltiples pruebas. Además, en el pasillo se realizó una prueba más, esta vez con único LED. En las diferentes pruebas experimentales realizadas su error medio fue de 0,4 m, 0,7 m y 0,8 m en cada una de las tres salas de pruebas, logrando un error medio de 1,1 m cuando sólo se utilizó una luminaria.

TOA

Según [65], para poder utilizar la técnica de posicionamiento basada en tiempo de llegada (*Time of Arrival*, TOA) mediante luz visible es necesario conocer tres datos fundamentales:

1. el momento exacto en el que la señal fue emitida.
2. el momento exacto en el que la señal fue recibida.
3. la velocidad con la que la señal viajó desde el emisor al receptor.

Conocidos esos tres valores, es posible determinar la distancia entre emisor y receptor mediante la ecuación (2.3), donde c es la velocidad de la luz. Se trata, precisamente, de la técnica utilizada por el GPS para determinar la posición del receptor.

$$d = c * (t_{recepción} - t_{emisión}) \quad (2.3)$$

En dos dimensiones, conocida la distancia, es posible obtener un conjunto de posibles posiciones en las que el emisor se encuentra gracias a la ecuación (2.4). El radio de la circunferencia es conocido (d en la ecuación (2.3)), así como la posición del emisor $((x, y)_{emisor})$.

$$d = \sqrt{(x - x_{emisor})^2 + (y - y_{emisor})^2} \quad (2.4)$$

Disponer de un segundo emisor permitiría encontrar la intersección entre dos circunferencias. La solución a este problema tendría dos soluciones (o una en caso de que ambos emisores se encontrasen a la misma distancia del receptor). Para determinar sin ambigüedades la posición en dos dimensiones del receptor sería necesario que hubiese tres emisores en el rango de cobertura del receptor. En la

figura 2.12 se puede ver la detección de un emisor, de dos y, por último, del tercer emisor que permite conocer la posición del receptor.

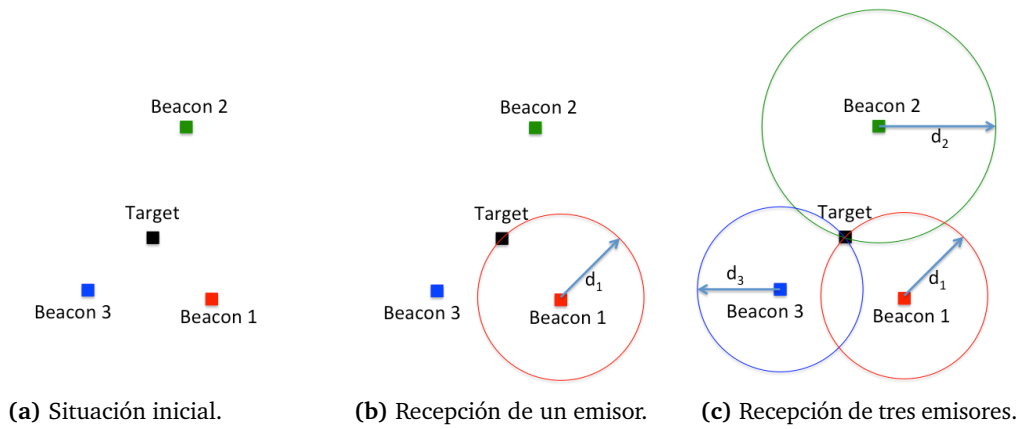


Figura 2.12. Pasos en el posicionamiento mediante TOA.

En [66] se analiza la viabilidad de un sistema de posicionamiento basado en luz visible mediante TOA. Se supone que que emisor y receptor están perfectamente sincronizados. En concreto, la habitación estudiada tenía las características que pueden verse en la figura 2.13.

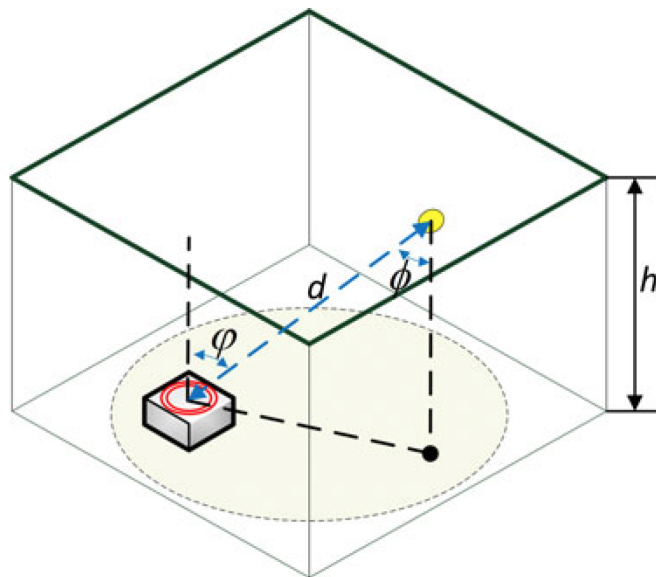


Figura 2.13. Sistema de posicionamiento mediante análisis de TOA [66].

Utilizando la cota de Cramér-Rao se llega a la conclusión de que el rango de error está entre 2 y 5 cm, dependiendo de la geometría de la habitación, de la frecuencia y la potencia de emisión del LED, y de características del PD como su superficie.

La principal conclusión que se puede sacar del uso de **TOA** en posicionamiento mediante luz visible (*Visual Light Positioning*, **VLP**) es que se trata de una aproximación demasiado compleja, para la que sólo existen modelos teóricos. La necesidad de que emisor y receptor estén perfectamente sincronizados supone un escollo, al igual que lo supone la velocidad de la luz. Posiblemente ésta sea la razón por la que los investigadores en este campo estén intentando solucionar el problema utilizando métodos más simples y con más posibilidades de utilizarse en aplicaciones prácticas.

TDOA

Como se ha comentado al hablar de **TOA** (ver la sección 2.4.2), se trata de una técnica de posicionamiento que requiere una sincronización perfecta entre emisor y receptor. Cuando esto no es posible, algunos sistemas de posicionamiento utilizan diferencia de tiempo de llegada (*Time Difference of Arrival*, **TDOA**). De hecho, [67] presenta **TDOA** como un método para mejorar **TOA**. Sin embargo, tal y como se menciona en [10], esto no significa que el sistema sea totalmente asíncrono: sigue siendo necesario que exista sincronización entre los emisores. Por lo general, un sistema **TDOA** necesita que dos receptores calculen la diferencia temporal en la adquisición de una señal. Sin embargo, el mismo resultado puede conseguirse utilizando un único receptor y varios emisores. Una vez obtenidas esas diferencias es posible utilizar multilateración [68] para determinar la posición del receptor.

En [69] se describe un sistema que utiliza **TDOA** para posicionar en 2D. Para ello, plantea aprovechar la infraestructura de iluminación existente. En el momento de escribir dicho artículo (2011), las luminarias **LED** se adivinaban como posibles sustitutos de los fluorescentes. El receptor utiliza un **PD**. Los datos proporcionados son resultado de simulaciones.

Las dimensiones del modelo son $5 \times 5 \times 3 \text{ m}^3$. Se utilizan tres luminarias **LED** situadas en las posiciones (1,5 1,5 3), (1,5 3,5 3) y (3,5 1,5 3). En la figura 2.14a puede verse un diagrama de la habitación simulada. Cada baliza emite a una determinada frecuencia (1 MHz, 3 MHz y 5 MHz). Se utilizan filtros de paso de banda para identificar cada luminaria.

Para comprobar la fiabilidad del sistema se establecen 81 puntos de medida en el plano que se encuentra en el suelo. El error máximo es de 4,5 mm, con una media de 1,8 mm. Estos resultados pueden verse en la figura 2.14b.

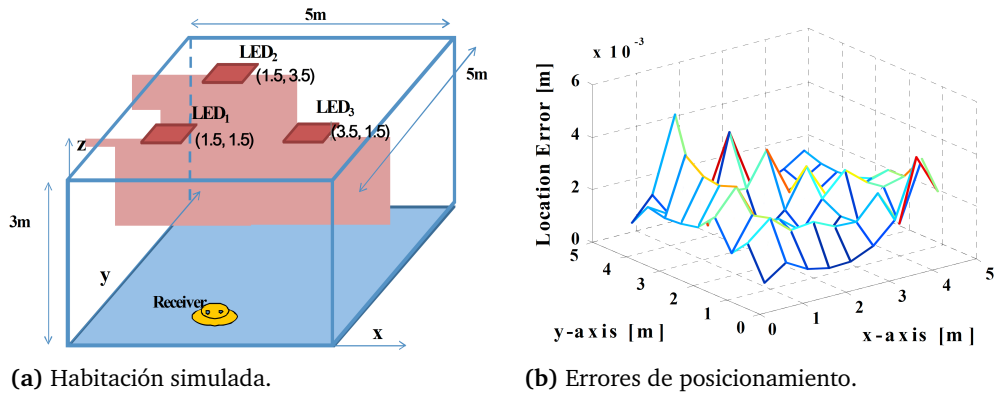


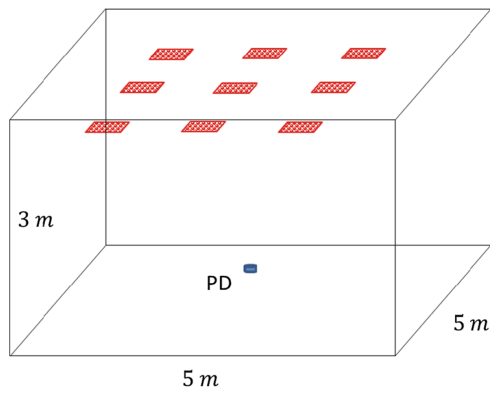
Figura 2.14. Posicionamiento mediante TDOA [69].

Este sistema no tiene en cuenta la influencia del ruido en el rendimiento del sistema de posicionamiento, algo que se subsana en [70]. Utilizando MATLAB se simula una habitación de las mismas dimensiones, aunque con tres luminarias como en el primero de ellos. El resultado es un sistema resistente al ruido, capaz de posicionar en 2D con menos de 2 cm de error.

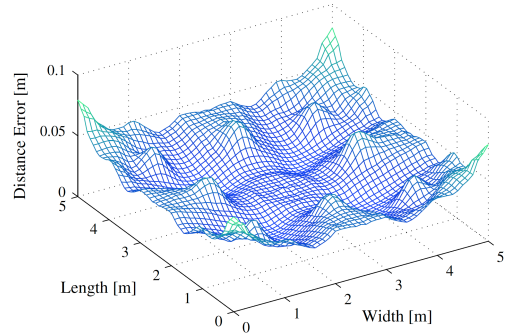
Tanto [69] como [70] presentan algunas limitaciones. En primer lugar, la imposibilidad de posicionar en 3D, que [71] sí ofrece. En segundo lugar, la técnica de codificación utilizada en [69] y [70] afectaba a las condiciones de iluminación de la sala, puesto que cada panel oscilaba en una frecuencia distinta, provocando apreciables diferencias en la potencia de emisión luminosa. Sin embargo, [71] utiliza un sistema de codificación sin dicho inconveniente que, además, presenta la ventaja de que su decodificación en el receptor es más sencilla. Estos resultados se consiguen en una sala de idénticas dimensiones aunque con más paneles LED en el techo, como puede verse en la figura 2.15a. Además, estos paneles están a 0,5 m del techo. Todos los paneles, menos el central, están a la misma distancia de las paredes.

Para poder posicionar en 3D [71] utiliza multilateración. Por su parte, [69] utiliza trilateración. Aunque la complejidad de este último algoritmo es menor, con tres fuentes de luz sólo es posible posicionar en 2D. En la figura 2.16 pueden compararse las representaciones gráficas de los valores obtenidos en ambos trabajos, que posteriormente se utilizarán para estimar la posición del receptor.

El esquema de codificación utilizado por [71] consiste en la emisión de señales piloto con diferentes potencias desde cada uno de los paneles LED, a intervalos regulares de duración 10 ns y con 40 ns de separación entre ellas. El impacto de dichas señales en las condiciones de iluminación de la sala es inapreciable por el ojo humano. El resultado es un sistema de posicionamiento con un error medio de 3,6 cm.

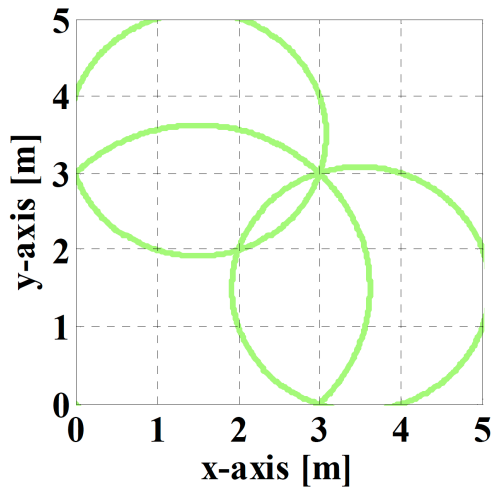


(a) Habitación simulada.

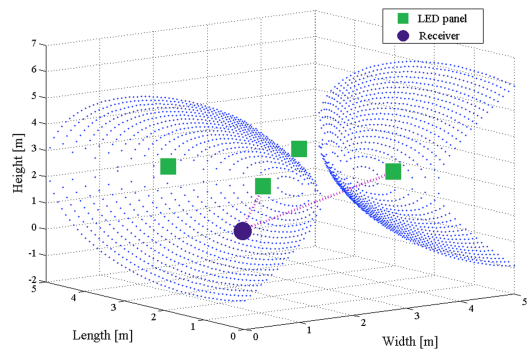


(b) Errores de posicionamiento.

Figura 2.15. Posicionamiento mediante TDOA [71].



(a) Trilateración [69].



(b) Multilateración [71].

Figura 2.16. Trilateración frente a multilateración.

AOA

Según [67], el posicionamiento mediante la técnica de ángulo de llegada (*Angle of Arrival*, AOA) se consigue obteniendo el ángulo entre emisor y receptor. Es necesario disponer de un mínimo de dos emisores. De esta manera se puede calcular la intersección de dos líneas utilizando las relaciones geométricas, siendo el punto de corte la posición del receptor. Este principio está descrito en la figura 2.17.

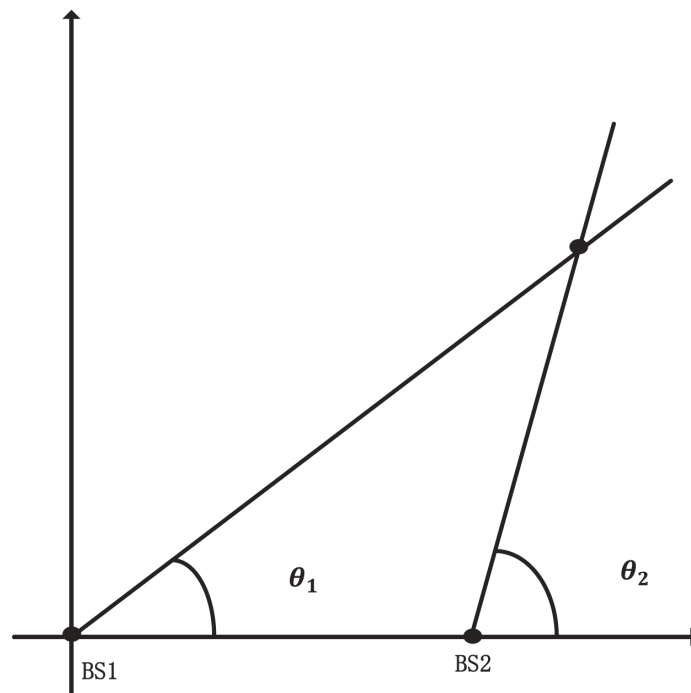


Figura 2.17. Sistema de posicionamiento mediante ángulo de llegada [67].

Suponiendo que los ángulos de incidencia entre la línea acimutal de la señal luminosa y las dos luminarias LED, llamadas BS1 y BS2, son θ_1 y θ_2 , respectivamente, que las coordenadas de las luminarias son (x_i, y_i) , mientras que las del receptor son (x_0, y_0) , se cumple la ecuación (2.5):

$$\tan \theta_i = \frac{(y_i - y_0)}{(x_i - x_0)}, i = 1, 2 \quad (2.5)$$

En [72] se utiliza una combinación de varios PD para poder determinar el ángulo de llegada de la luz. Dicho sensor puede verse en la figura 2.18.

En las simulaciones el rendimiento, este sistema tenía una precisión angular con errores medios de 2° y un error de 5 cm de media en el posicionamiento. Desafortunadamente, no se aportan resultados experimentales reales.

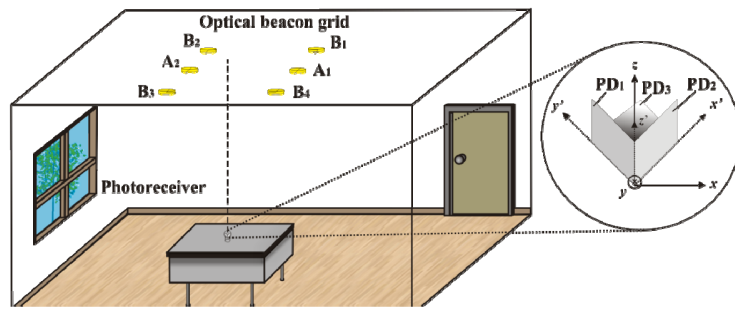


Figura 2.18. Diagrama del sistema QADA de A. Arafa *et al.* [72].

La mayoría de los trabajos citados como basados en AOA no lo son realmente, al menos desde el punto del vista del autor de este trabajo de tesis. Por sólo citar dos ejemplos, en [67] se mencionan [3] y [73] como sistemas AOA. Del primero de ellos se habla en la sección 2.4.6, y en este capítulo se clasifica dentro de los sistemas basados en el análisis de la visión. Por otra parte, [73] presenta un sistema de posicionamiento basado en esta técnica. En dicho sistema se cuenta con tres emisores LED situados en el techo de la sala de pruebas. El receptor está compuesto por un PD y un acelerómetro. El primero se utiliza para medir la potencia recibida de cada emisor. El segundo, para conocer la inclinación del receptor. Para llevar a cabo el posicionamiento son necesarias dos medidas, hechas en la misma posición pero con diferentes inclinaciones. En cada una de ellas es necesario tener visión directa de las tres luminarias. En las simulaciones realizadas se empieza por obtener resultados en 2D, para luego extender el sistema a 3D. El sistema tiene un error de posicionamiento por debajo de 0,5 m. Sin embargo, la medición del ángulo no se realiza en el PD, se estima a partir de la posición del móvil, por lo que la parte principal del sistema se basa en la medición de la intensidad de la señal, razón por la que se diría que pertenece a la categoría de RSS.

Un innovador sistema de posicionamiento basado en AOA mediante un cuadrante de fotodiodos con apertura angular diversa (*Quadrant Photodiode Angular Diversity Aperture, QADA*) se presenta en [74] en 2018. Se trata de un receptor cuyo diseño puede verse en la figura 2.19. En la parte inferior se encuentran cuatro PDs independientes con una separación mínima, capaces de realizar posicionamiento en 2D. Sobre ellos, una apertura situada a una altura conocida cuya superficie es menor que la expuesta por los cuatro PD. Dicha apertura se utiliza para que la luz incida de forma definida sobre los PDs, de manera que sea posible calcular el ángulo de llegada.

En el banco de pruebas desarrollado para determinar la capacidad del receptor en la estimación del ángulo de llegada de la fuente luminosa, el sistema tiene un error

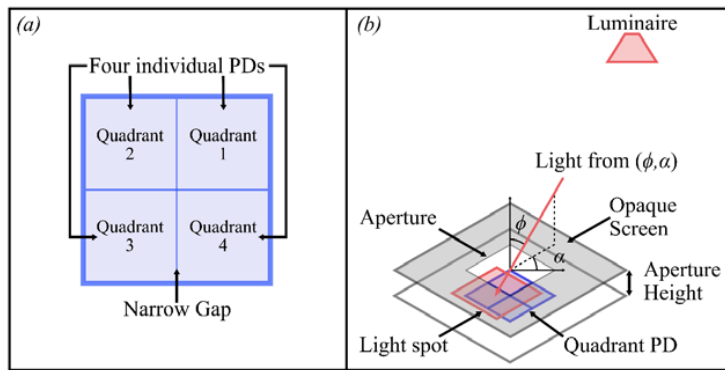


Figura 2.19. Diagrama del sistema QADA de S. Cincotta *et al.* [74].

cuadrático medio inferior a $0,11^\circ$, aunque el 90 % de las medidas tomadas estaba por debajo de $0,07^\circ$.

La evolución de este sistema es QADA-PLUS, que permite realizar posicionamiento en dos fases [75], presentado previamente en el Congreso Internacional de Posicionamiento y Navegación en Interiores (*International Conference on Indoor Positioning and Navigation, IPIN*) de 2018. Para probar la eficacia del sistema se realizó una simulación en una sala con las características que pueden verse en la figura 2.20.

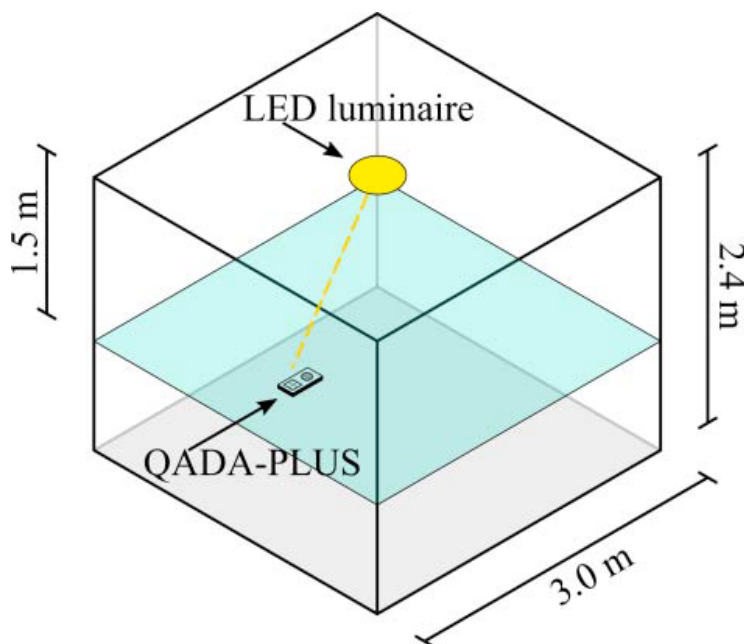


Figura 2.20. Diagrama del banco de pruebas simulado para el sistema QADA-PLUS [75].

La primera fase del sistema consiste en determinar el ángulo de incidencia del único panel LED necesario para realizar el posicionamiento. La segunda, en realizar un análisis de una imagen adquirida desde prácticamente el mismo punto en el

que se encuentra el receptor QADA utilizando la cámara frontal de un dispositivo móvil. QADA-PLUS presenta un error medio absoluto en la estimación del ángulo de incidencia de la luz de $0,0006^\circ$, mientras que para ángulos polares el error es de $0,005^\circ$. La utilización de dos fases diferenciadas supone una mejora substancial en los resultados obtenidos.

En el IPIN de 2019 se presentó un trabajo [76] desarrollado por la Universidad de Alcalá de Henares que utiliza un detector sensible a la posición (*Position Sensitive Detector*, PSD) como receptor, basado en los mismos principios que el QADA. Además del sensor, una de las aportaciones más interesantes de dicho trabajo es la incorporación de algoritmos genéticos al proceso de posicionamiento, logrando un error medio por debajo de 1 cm en el 80% de los puntos probados, con un error máximo por debajo de los 4,5 cm. En la figura 2.21a se puede ver el banco de pruebas utilizado, mientras que la figura 2.21b muestra los resultados obtenidos a 1 m de distancia.

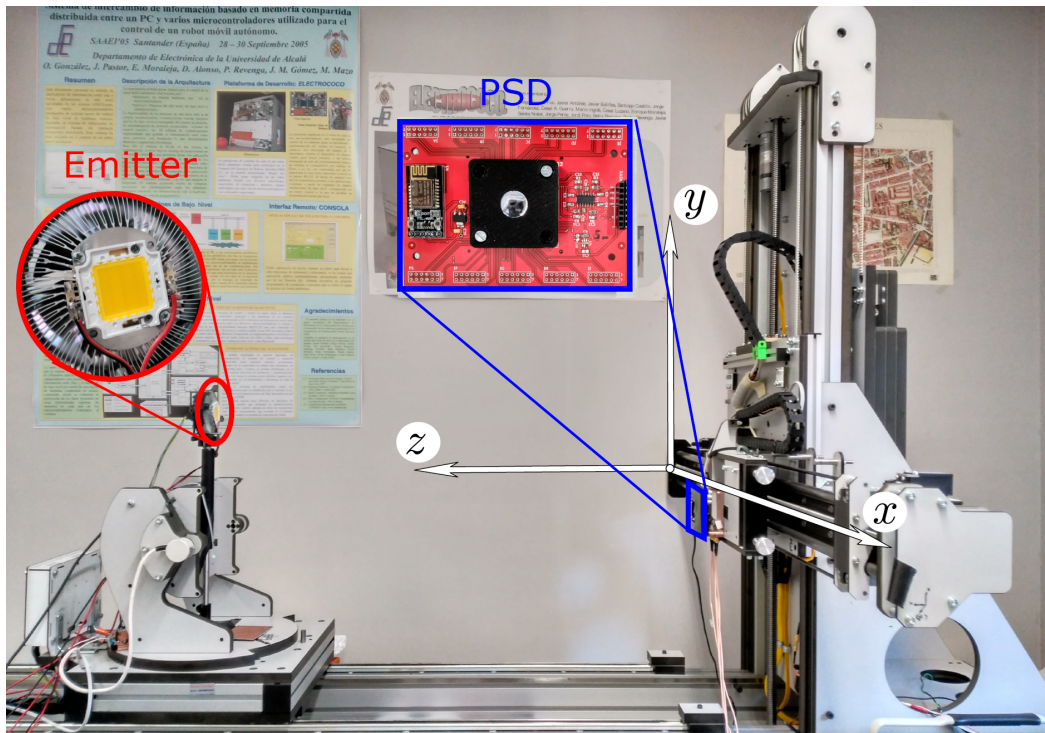
Métodos híbridos

Los cuatro sistemas de posicionamiento que se han discutido en las secciones anteriores (RSS en la sección 2.4.2, TOA en la sección 2.4.2, TDOA en la sección 2.4.2 y AOA en la sección 2.4.2) utilizan una única característica de la señal recibida para realizar la estimación de la posición. Como se menciona en [25], parece probable que la combinación de varias de estas técnicas permitiría un esquema de posicionamiento híbrido que sacase partido de las mejores características de cada señal en cada situación.

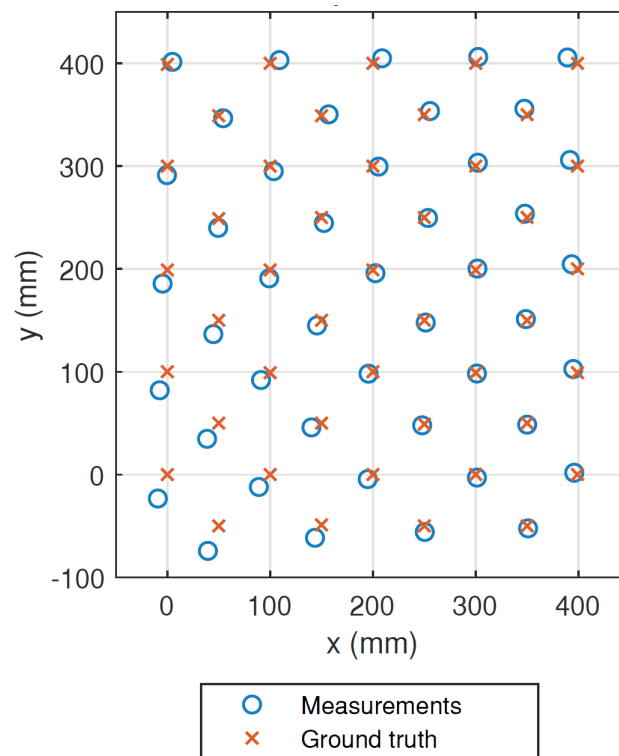
Una aproximación teórica a la técnica de posicionamiento mediante TOA puede encontrarse en [77], donde se estudian los límites y estimadores teóricos de sistemas de posicionamiento mediante VLC basados en TOA y RSS. Una de las aportaciones de [77] es la propuesta de un sistema híbrido TOA/RSS para la estimación de la distancia entre emisor y receptor, mejorando el rendimiento cuando existen limitaciones en la frecuencia de muestreo.

A lo largo de las secciones restantes de este capítulo tendremos ocasión de estudiar más ejemplos de sistemas híbridos que hacen uso de otras técnicas, además de las cuatro mencionadas hasta ahora. En la literatura existente, estos sistemas híbridos también se conocen como sistemas de fusión sensorial.

Tras revisar el estado en el que se encuentran los IPS basados en luz visible que realizan una estimación de parámetros, procedamos de forma similar con el segundo



(a) Banco de pruebas.



(b) Resultados.

Figura 2.21. Sistema propuesta por De-La-Llana-Calvo *et al.* en [76].

grupo, los estimadores de posición. Recordemos que estos últimos intentan determinar la posición de los emisores y, a partir de dichos valores, extrapolar la posición del receptor. Dicho grupo engloba las siguientes técnicas:

- **proximidad:** mediante los datos de identificación de las balizas más cercanas.
- **métodos geométricos:** aprovechan la interpretación geométrica de los datos obtenidos de las balizas.
- **métodos estadísticos:** más rigurosos matemáticamente que los geométricos.
- **huella digital/mapeo:** del inglés *fingerprinting/mapping*, se crea un mapa de características que, posteriormente, se utiliza para la localización.

Proximidad

En los **IPS** que utilizan técnicas de estimación de posición por proximidad, el receptor obtiene la identidad de la baliza más cercana. En caso de tener varias balizas en rango, escoge la más próxima entre ellas siguiendo criterios como, por ejemplo, su **RSS**. Una vez obtenida su identidad, puede utilizarla para obtener la ubicación de dicha baliza dentro del sistema de coordenadas de la sala y, a partir de ahí, deducir la posición del receptor. También existe la posibilidad de que las balizas transmitan su propia posición como parte de su identidad, lo que ahorra el paso extra de obtener la ubicación de la baliza a partir de su identidad. Si los requisitos del sistema no exigen demasiada precisión en la ubicación, identificar de una baliza y conocer el campo visual (*Field of Vision*, **FOV**) del receptor puede proporcionar la información necesaria para realizar el posicionamiento.

En [78] se presenta un **IPS** basado en proximidad que utiliza luminarias **LED** como emisores. Los receptores se han desarrollado *ex professo* combinando un **PD** y un módulo Zigbee [79] para la comunicación. En la figura 2.22 se pueden ver los diferentes elementos que conforman el prototipo desarrollado para las pruebas experimentales.

Cuando el receptor encuentra una baliza en las proximidades, intenta identificarla y se comunica con el nodo central para obtener los detalles de la baliza, entre los cuáles está la posición de la misma. La precisión de posicionamiento del sistema está entorno a 0,5 m, lo que se corresponde con el campo de visión del receptor.

El sistema propuesto en [80] es bastante similar al de [78], aunque los detalles del mismo varían. Su objetivo es proporcionar un sistema de guiado para personas con

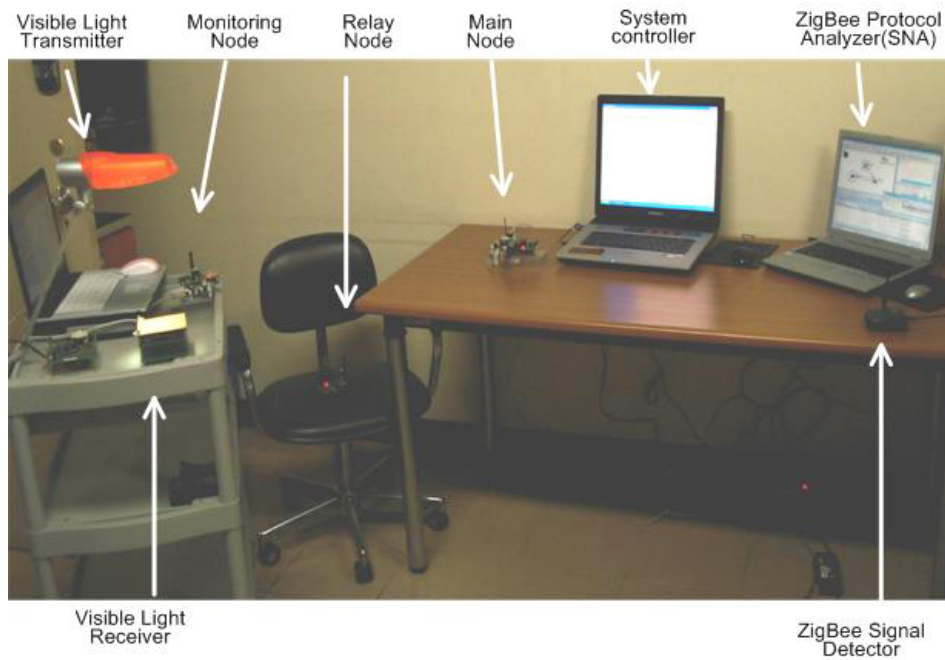
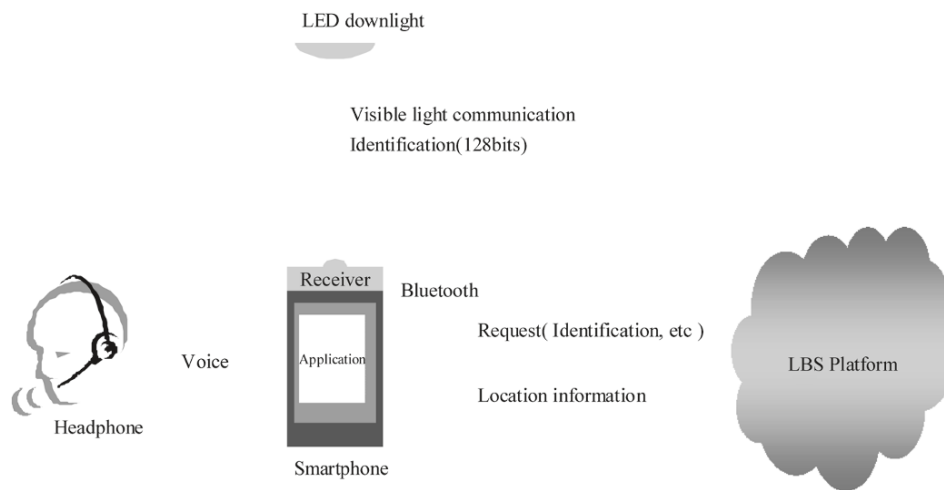


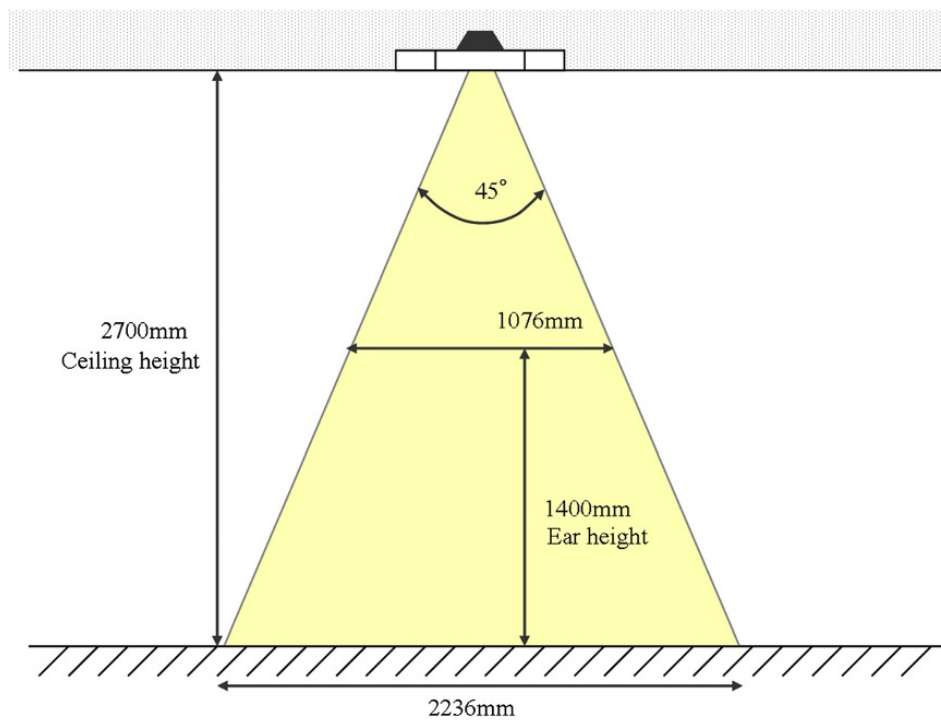
Figura 2.22. Prototipo [78].

discapacidad visual. En esta ocasión, el receptor está formado por un smartphone Android, un receptor de información lumínica conectado por Bluetooth al smartphone y unos auriculares que guían al usuario. En la figura 2.23a puede verse el prototipo utilizado para las pruebas experimentales. Igual que en [78], la precisión del sistema depende del campo de visión del receptor, como muestra la figura 2.23b. En este caso, dado que el prototipo está colgado del cuello del usuario, suponiendo una altura de aproximadamente 1,4 m, una baliza dentro de campo de visión del receptor permitiría alcanzar una precisión de 1,076 m. Este IPS supone un ejemplo más de sistema híbrido (ver la sección 2.4.2), puesto que a la señal óptica obtenida se incorporan los datos proporcionados por el sensor geomagnético del smartphone para obtener la ubicación del receptor de manera aún más precisa.

En la misma línea, mientras que los emisores [78] y [80] envían sus identificadores de forma ininterrumpida con una codificación OOK, [81] añade acceso múltiple por división de frecuencia (*Frequency-Division Multiplexing*, FDM) para poder recibir varios códigos de forma simultánea. En su propuesta juegan con diferentes zonas de cobertura y áreas de solapamiento para determinar cuál es la mejor distribución de balizas. El receptor es también un PD. El prototipo que muestra la figura 2.24a permite regular la altura de las balizas para estudiar tanto el área de cobertura como las zonas de solapamiento, en las que más de una baliza es detectada. Como puede verse en la figura 2.24b, la altura óptima es 1,5 m. A dicha altura, cada baliza



(a) Prototipo.



(b) Precisión.

Figura 2.23. Sistema propuesto por [80].

cubre una zona cuadrada de 1,5 m de lado, dentro de la que el sistema es capaz de determinar la posición del receptor con ese margen de error.

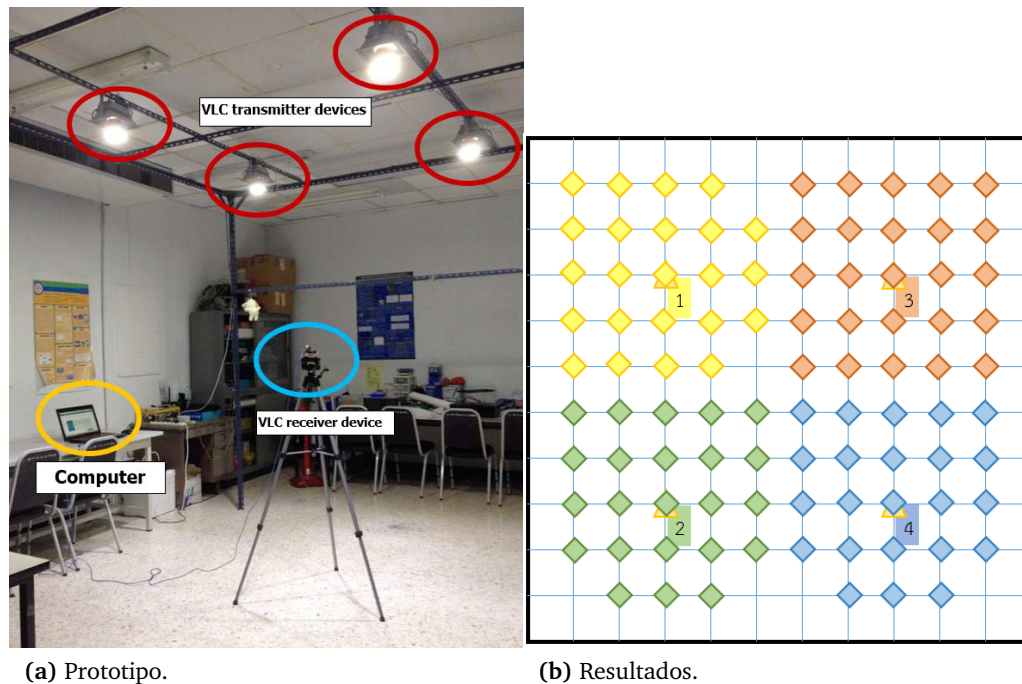


Figura 2.24. Sistema propuesto por [81].

Métodos geométricos

Al comienzo de esta sección vimos algunos de los trabajos que desarrollan **IPS** que aprovechan características observables de la señal luminosa como **RSS**, **TOA**, **TDOA** y **AOA** para determinar la posición del receptor dentro de una estancia en concreto. Los métodos geométricos de estimación de la posición combinan la obtención de estos parámetros observables con técnicas geométricas para determinar en qué posición se encontraba el receptor cuando dichos valores fueron obtenidos. Dependiendo del observable escogido, las técnicas geométricas aplicables se basan en cálculos de:

- **distancia:** utilizando **RSS** o **TOA** se puede estimar la ubicación mediante la obtención de lecturas de tres balizas, permitiendo describir tres círculos cuya intersección resulta ser la posición del receptor en dos dimensiones. Si, en lugar de tres balizas, son cuatro, se pueden describir cuatro esferas, siendo la intersección de ellas la posición del receptor, un punto en tres dimensiones. Esta técnica, conocida como trilateración, se trata en más detalle en la sección 2.4.5.

- **ángulo:** las técnicas de AOA descritas en la sección 2.4.2 se aprovechan de las características geométricas de los ángulos para estimar la ubicación del receptor.
- **diferencia de distancia:** como en la categoría anterior, ésta y las técnicas de TDOA están estrechamente relacionadas.

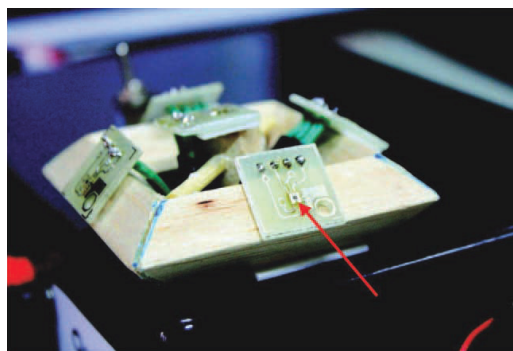
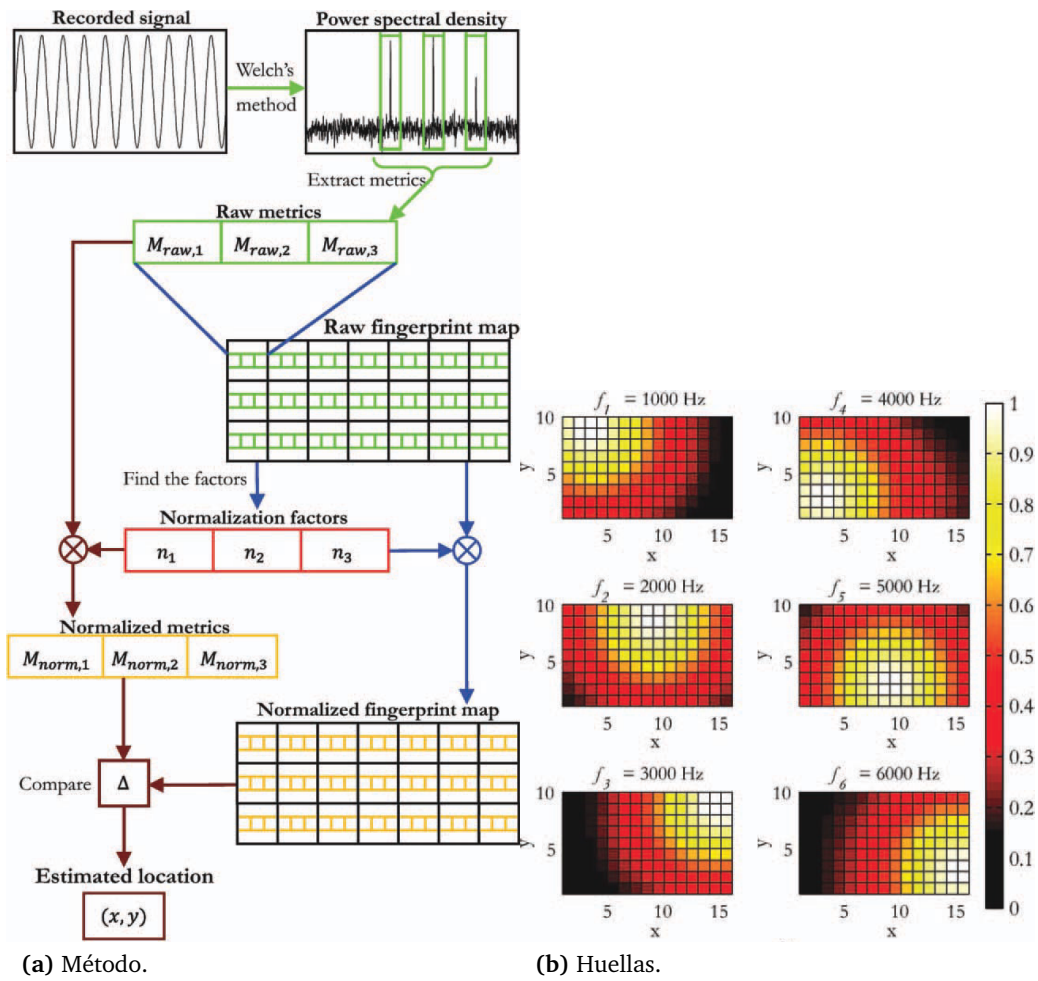
2.4.2.1. Métodos estadísticos

Los métodos de posicionamiento estadístico, según [25], utilizan las propiedades estadísticas de los observables empleados para estimar la posición del receptor. Este sistema es matemáticamente más riguroso que los métodos geométricos, ya que estos últimos dependen únicamente del conocimiento derivado de las operaciones geométricas. Así, las técnicas estadísticas suponen un sistema metódico capaz de proporcionar garantías de rendimiento y optimización, como se describe en [82]. Precisamente, la primera formulación genérica para la estimación de la posición basada en un modelo de medidas estadísticas se presenta en [25], una de las principales referencias de este estudio de la situación actual del posicionamiento en interiores basado en luz visible, donde se diferencia entre sistemas síncronos, casi síncronos y asíncronos.

2.4.2.2. Huella digital/mapeo

La técnica de huella digital/mapeo (del inglés *fingerprinting/mapping*) es de sobra conocida en el mundo del posicionamiento en interiores, sobre todo en sistemas basados en RF [83]. Se trata de un proceso en dos fases. La primera, *offline*, consiste en la toma de muestras por toda la superficie de la zona a mapear, recogiendo los valores observables más útiles para el posicionamiento en cada punto. Estos valores se utilizan para generar un mapa o base de datos. En la segunda fase, *online*, el receptor obtiene los mismos valores observables que en la primera, y los compara con los valores obtenidos en la primera fase. Como resultado, se obtiene la posición más probable del receptor.

En el trabajo desarrollado por Vongkulbhisal *et al.* [84] se utiliza la RSS como valor observable para construir la base de datos durante la fase *offline*. El método completo utilizado por este sistema, desde la creación del mapa hasta la evaluación de la ubicación del receptor, pasando por la normalización de los valores obtenidos, puede verse en la figura 2.25a. La RSS es el valor más frecuentemente utilizando en los IPS basados en luz visible que utilizan la técnica de huella digital.



(c) Prototipo.

Figura 2.25. Sistema propuesto por Vongkulbhisal *et al.* [84].

La principal desventaja de este tipo de sistemas es la necesidad de mantener actualizado el mapa o base de datos generado durante la fase offline. Cualquier alteración que implique cambios en los valores observables registrados tendrá como consecuencia una degradación en la calidad del posicionamiento. En las simulaciones realizadas se llega a la conclusión de que la precisión del sistema se reduce si la intensidad de la señal modulada deja de ser la misma capturada para la realización del radiomap. De hecho, durante las pruebas experimentales tanto los emisores como el receptor estaban completamente cubiertos por una tela negra para evitar aportaciones lumínicas fuera del sistema. Por otra parte, en las pruebas experimentales demuestran que la precisión del sistema es mayor cuanto menor sea la cuadrícula utilizada para generar el mapa, aunque la generación de dicho mapa llevará más tiempo. Mientras el mapa permanezca útil, la precisión del sistema puede llegar a ser centimétrica, con un error entre 14 y 27 cm dependiendo de las condiciones.

También utilizando huella digital, en [85] se crea un mapa con celdas de 0,5 m de lado en una superficie total de 15 m². En simulación se consigue una precisión de 1 m en las esquinas de la sala, y 0,1 m o menos en el resto de posiciones. El rendimiento del sistema es prácticamente similar sin importar el tamaño de las celdas: también se simulan con lados de 1 y 0,25 m. Además, aprovechan para comparar el rendimiento del sistema de huella digital con otras técnicas como la triangulación (ver la sección 2.4.4). Como puede verse en la figura 2.26, el rendimiento de la utilización de huella digital y mapa es mayor que el de triangulación.

En la sección 2.4.2 hablamos de las diferentes estrategias que pueden llevarse a cabo combinando la obtención de parámetros observables con técnicas geométricas para determinar en qué posición se encontraba el receptor cuando dichos valores fueron obtenidos. En la siguiente sección nos centraremos en utilizar una de las cámaras de un dispositivo portable para tomar una fotografía y, mediante técnicas de análisis de visión (que, en definitiva, incluirán algunas técnicas geométricas) estimar la posición del receptor.

2.4.3. Análisis de visión

Si comparamos utilizar un sensor de imagen de semiconductor complementario de óxido metálico (*Complementary Metal-Oxide-Semiconductor Image Sensor*, **CIS**) incluido en un dispositivo portable con utilizar un **PD**, estamos implícitamente diferenciando entre aquellos **IPS** que utilizan hardware desarrollado *ex professo* y los que aprovechan elementos ya existentes como, por ejemplo, alguna de las cámaras incluidas en los dispositivos portables. Los **PD** son más económicos, más

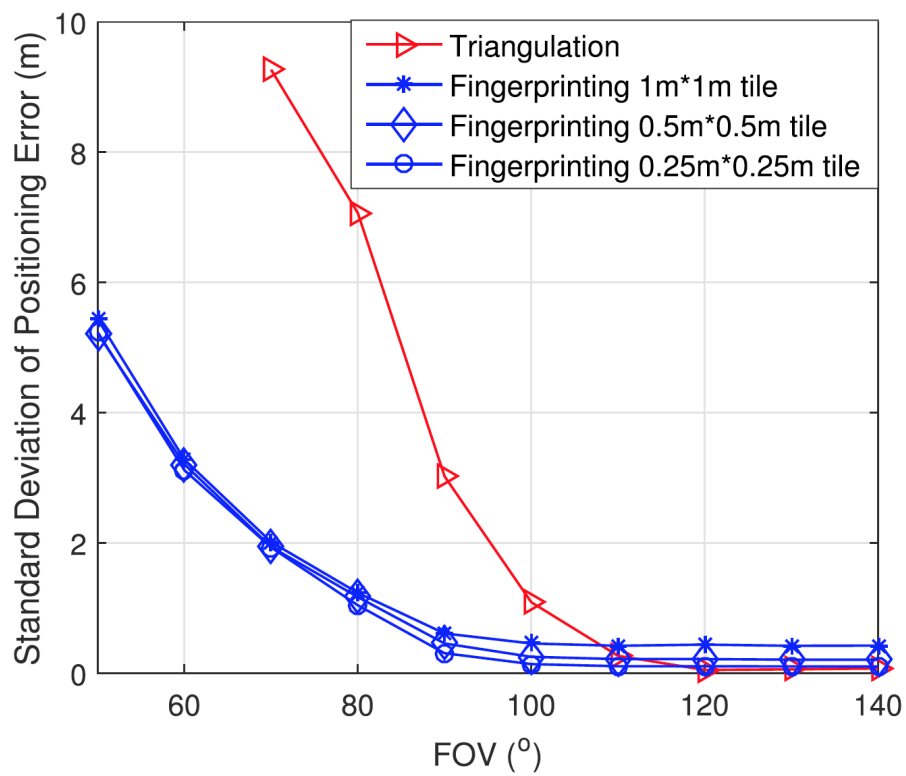


Figura 2.26. Desviación estándar del error de posicionamiento en [85].

eficientes energéticamente y con mayor ancho de banda. Las cámaras incorporadas en los dispositivos portables son omnipresentes y, además, permiten transmisión de datos, aunque disponen de menor ancho de banda que los PD. En ambos casos es necesaria visibilidad directa entre emisor y receptor. Sin embargo, la principal ventaja que supone disponer de una cámara es poder analizar la escena utilizando el procesador del dispositivo portable. También es interesante el hecho de que una cámara de tipo semiconductor complementario de óxido metálico (*Complementary Metal-Oxide-Semiconductor*, CMOS) no es más que una matriz de sensores, así que podríamos pensar en que estamos utilizando una enorme cantidad de sensores, aunque sólo son capaces de capturar información de color e intensidad luminosa de la escena, en lugar de sensores dedicados capaces de capturar información más precisa como únicamente la intensidad de la luz capturada.

Según [86], cuando se habla de posicionamiento mediante luz visible utilizando análisis de visión, de lo que se habla es de desarrollar un sistema capaz de establecer relaciones entre tres sistemas de coordenadas:

1. el sistema de coordenadas 3D del mundo real.
2. el sistema de coordenadas 2D de la imagen capturada.
3. el sistema de coordenadas 3D de la cámara.

La relación entre estos tres sistemas de coordenadas se puede ver en la figura 2.27.

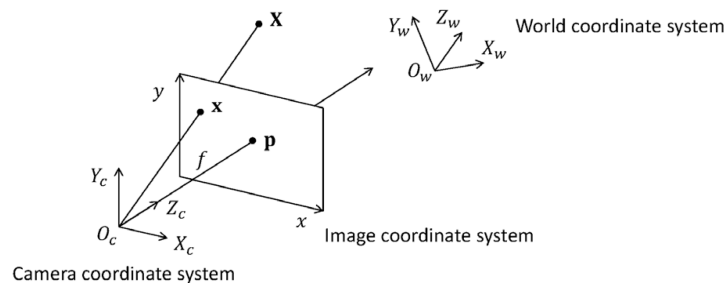


Figura 2.27. Relación entre sistemas de coordenadas para análisis de visión [86].

El punto X_w tiene coordenadas en el mundo real, mientras que el punto X_c las tiene con respecto a las coordenadas de la cámara. El punto $p = (x, y)$ es una proyección del punto X_w con respecto a la cámara situada en X_c .

La pose de la cámara (su posición y su orientación) se puede obtener a partir de la transformación descrita en la ecuación (2.6):

$$\tilde{\mathbf{X}}_c = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 1 \end{bmatrix} \tilde{\mathbf{X}}_w \quad (2.6)$$

donde $\tilde{\mathbf{X}}_c = (X_c, Y_c, Z_c, 1)^T$ son las coordenadas homogéneas de la cámara, $\tilde{\mathbf{X}}_w = (X_w, Y_w, Z_w, 1)^T$ son las coordenadas homogéneas de un punto en el mundo real) y \mathbf{R} es una matriz de rotación con dimensiones 3×3 . De aquí se puede deducir que la pose de la cámara es equivalente a $[\mathbf{R}|\mathbf{t}]$.

El sistema de coordenadas de la cámara se define como aquel cuyo origen está situado en el centro de la cámara, y en el que la dirección de Z_c es perpendicular al plano de la imagen. Este vector atraviesa la imagen capturada justo por su centro, partiendo del centro de la cámara. El punto de intersección del plano de la imagen y el eje Z_c , conocido como punto principal, es $\mathbf{p} = (p_x, p_y)$. En modelo de cámara estenopeica, un punto en tres dimensiones $\mathbf{X}_c = (X_c, Y_c, Z_c)^T$ se proyecta en un punto en dos dimensiones $\mathbf{x} = (x, y)^T$, dentro del sistema de coordenadas de la imagen, siguiendo la ecuación (2.7):

$$(x, y)^T = \left(f \frac{X_c}{Z_c} + p_x, f \frac{Y_c}{Z_c} + p_y \right) \quad (2.7)$$

donde la ecuación (2.7) f es la distancia focal de la lente. Sea \mathbf{A} la matriz de calibración de cámara. Se cumple la ecuación (2.8), verificándose que:

$$\mathbf{A} = \begin{bmatrix} f & 0 & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \quad (2.8)$$

Es posible describir la transformación de un punto en tres dimensiones con coordenadas en el mundo real a un punto en dos dimensiones dentro del sistema de coordenadas de la imagen como en la ecuación (2.9):

$$\tilde{\mathbf{x}} \sim \mathbf{A} [\mathbf{R}|\mathbf{t}] \tilde{\mathbf{X}}_w \quad (2.9)$$

siendo $\tilde{\mathbf{x}} = (x, y, 1)$ las coordenadas homogéneas de un punto de la imagen. Es posible simplificar esta ecuación tal y como muestra la ecuación (2.10):

$$\begin{aligned} \tilde{\mathbf{x}} &\sim \mathbf{P} \tilde{\mathbf{X}}_w \\ \mathbf{P} &= \mathbf{A} [\mathbf{R}|\mathbf{t}] \end{aligned} \quad (2.10)$$

en la que \mathbf{P} es una matriz de proyección de perspectiva con dimensiones 3×4 que, además, representa la pose de la cámara.

Para estimar la pose \mathbf{P} de la cámara resolviendo las ecuaciones anteriores es necesario obtener varias parejas de valores de $\tilde{\mathbf{x}}$ y $\tilde{\mathbf{X}}_w$. Por ejemplo, es posible calcular \mathbf{P} a partir de seis parejas porque \mathbf{P} tiene 12 incógnitas y cada pareja proporciona 2 ecuaciones. Si \mathbf{A} es conocida (calibrando la cámara, como en [87] [88]) se puede obtener la pose a partir de cuatro pares de valores [86]. Sin embargo, existen otras formas de solucionar este problema dependiendo de las condiciones de partida.

Para los más interesados en el análisis de visión, la estimación de la orientación, ubicación y distancia de la cámara utilizando VLC y cámaras se presenta en [86], donde se hace referencia al famoso libro de Szeliski [89] y se comenta que *existe mucha literatura al respecto de la estimación de la orientación de la cámara utilizando puntos ya que se trata de tema recurrente en la investigación.*

En general, los sistemas capaces de determinar la pose de la cámara a partir de la imagen capturada, conociendo la posición de las fuentes luminosas, se dividen en dos grupos: los de triangulación y los de geometría de vista única.

2.4.4. Triangulación

En los sistemas de posicionamiento que utilizan triangulación se usan dos cámaras para obtener la ubicación en tres dimensiones de un punto real, según [86]. Como se menciona en [90], se supone que la posición relativa y la orientación de una cámara con respecto a la otra son conocidas. Por lo tanto, la matriz de rotación \mathbf{R} y el vector de traslación \mathbf{f} de una puede expresarse en función de la otra. En la figura 2.28 se ilustra este caso.

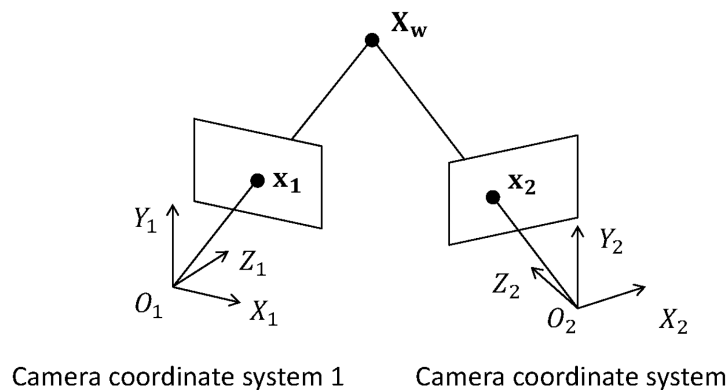


Figura 2.28. Triangulación [86].

Es posible describir la proyección del punto $\tilde{\mathbf{X}}_w$ del mundo real en tres dimensiones en dos cámaras de la forma presentada en la ecuación (2.11).

$$\begin{aligned}\tilde{x}_1 &\sim P_1 \tilde{X}_w \\ \tilde{x}_2 &\sim P_2 \tilde{X}_w\end{aligned}\tag{2.11}$$

En la ecuación (2.11) P_i es la matriz de proyección de perspectiva de cada cámara y \tilde{x}_i son las coordenadas homogéneas dentro de la imagen obtenida al proyectar el punto \tilde{X}_w en la cámara correspondiente. Tanto P_1 como P_2 son conocidas, puesto que trabajamos con cámaras cuyas características se conocen. El objetivo de la triangulación es calcular \tilde{X}_w a partir de los valores de \tilde{x}_1 y \tilde{x}_2 .

En el trabajo presentado por primera vez en el congreso ICCIT en 2011 [91] que, posteriormente fue publicado por la revista *International Journal of Smart Home* [92] se propone un sistema compuesto por cuatro LED y dos cámaras que puede verse en la figura 2.29.

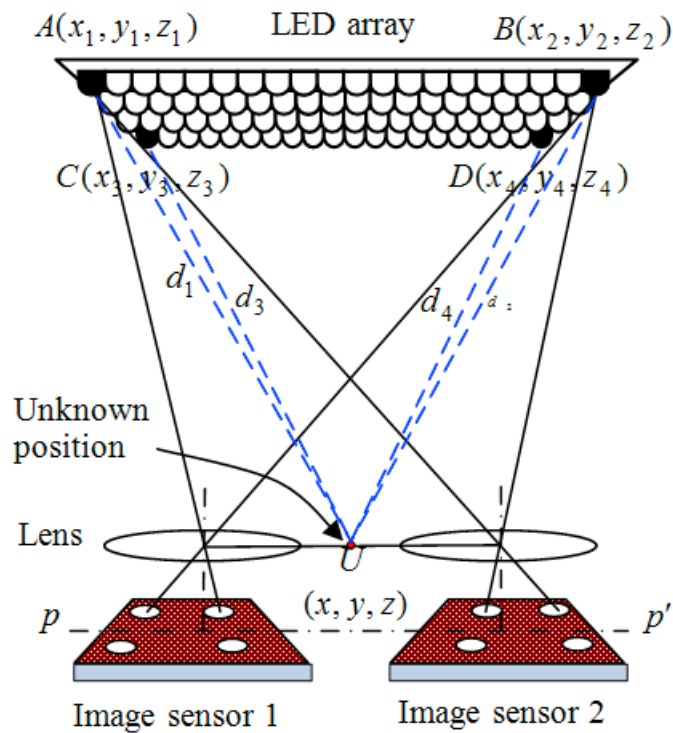


Figura 2.29. Diagrama para la simulación de un sistema de posicionamiento basado en triangulación [92].

Para poder estimar correctamente la posición 3D del receptor, este sistema necesita ver cuatro LED simultáneamente en cada una de las dos cámaras. El resultado de las ecuaciones cuadráticas mejora utilizando estimación por mínimos cuadrados. Una de las condiciones para poder resolver este tipo de problemas es que la matriz M tenga valores diferentes de cero en todas su celdas. Sin embargo, como todos los LED están

en el mismo plano, la condición no se cumple. Para resolverlo se añade una ligera diferencia en el cuarto LED, pero se obtiene una estimación incorrecta de la altura. El valor del eje Z se corrige utilizando estimación vectorial. El resultado en las simulaciones consigue una precisión de 15 cm sin necesidad de medida angular en simulaciones. El sistema mejora con más resolución en las cámaras o más separación entre ellas.

Posteriormente, en [93] se crea un prototipo a partir de los modelos presentados en [91] y [92], mejorando los resultados previstos en un 61,5 %, con una precisión de 8,5 cm. Parte de la mejora se atribuye al cálculo de las coordenadas del centro de cada LED como un valor en coma flotante, en lugar de utilizar un entero como se hacía en las simulaciones. Un diagrama de este prototipo puede verse en la figura 2.30.

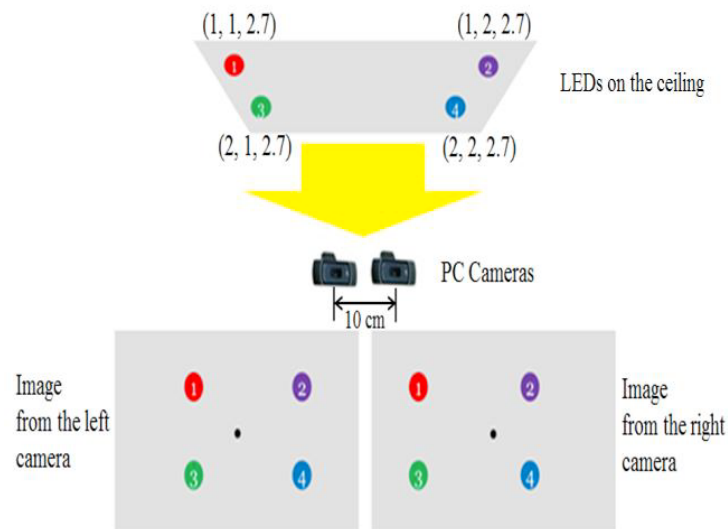


Figura 2.30. Diagrama del prototipo de un sistema de posicionamiento basado en triangulación [93].

El lado del cuadrado definido por los LED es de 90 cm, la distancia entre cámaras de 10 cm, los LED estaban a 2,7 m de altura. Las cámaras estaban conectadas a un ordenador personal (*Personal Computer, PC*), las dimensiones de sus sensores eran de 24 mm \times 24 mm, con una distancia focal de 26,94 mm. Las imágenes capturadas tenían 1280 px \times 960 px.

En [94] se da un paso más, permitiendo cambios arbitrarios en el acimut del receptor, tanto en un modelo como en un prototipo real. En la figura 2.31 se puede ver un diagrama del prototipo ideado. Sólo son necesarias dos luminarias emisoras de luz, situadas a 2,6 m de altura. La zona en la que se realiza el posicionamiento es un cuadrado de 60 cm de lado. Las cámaras están separadas 10 cm entre sí. El sensor

de las cámaras utilizadas tiene unas dimensiones de $3,2 \text{ mm} \times 2,4 \text{ mm}$, con una distancia focal de $4,4 \text{ mm}$. Las dimensiones de las imágenes capturadas van desde $160 \text{ px} \times 120 \text{ px}$ hasta $1280 \text{ px} \times 960 \text{ px}$. El acimut del receptor va desde los -180° hasta los 175° , con pasos de 5° .

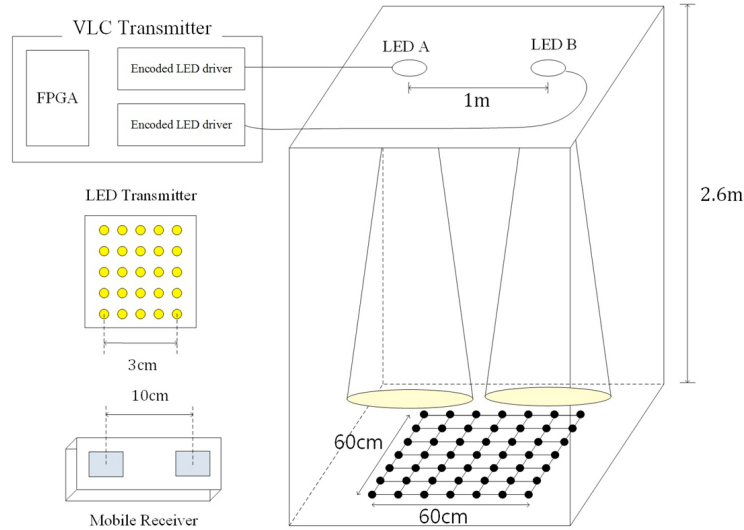


Figura 2.31. Diagrama del prototipo de un sistema de posicionamiento basado en triangulación [94].

En la resolución más alta, los errores de simulación en los ejes x , y y z son $0,07 \text{ cm}$, $0,02 \text{ cm}$ y $0,32 \text{ cm}$, respectivamente. Sin embargo, los resultados experimentales arrojan errores más altos ($2,26 \text{ cm}$, $5,51 \text{ cm}$ y $1,46 \text{ m}$). La gran discrepancia de errores entre simulación y experimento son achacados a la dificultad de encontrar el centro real de cada luminaria, entre otros factores. Que el error del eje z sea más significativo que el de los ejes x e y se debe a que el primero depende de los dos últimos.

Supongamos que un punto del mundo real P tiene las proyecciones P'_1 y P'_2 en dos cámaras. Según [86], dichos puntos pueden describirse como en la ecuación (2.12):

$$\begin{cases} (x_1, y_1)^T &= M_1 (X, Y, Z)^T \\ (x_2, y_2)^T &= M_2 (X, Y, Z)^T \end{cases} \quad (2.12)$$

donde M_i es la matriz de proyección de perspectiva de la cámara i , que puede verse en la ecuación (2.13).

$$M_i = C_i \begin{bmatrix} R_i & t_i \\ 0 & 1 \end{bmatrix} \quad (2.13)$$

La triangulación es el algoritmo más frecuentemente utilizado para obtener la posición del receptor cuando se utiliza AOA. Como ejemplo, ALTAIR [95] invierte el esquema más frecuentemente utilizado, con emisores LED en el techo y un receptor en la sala. En su lugar, las cámaras están montadas en el techo y cada usuario lleva puesta una etiqueta LED de IR que emite un código diferente. Cada una de estas etiquetas emite su señal de forma secuencial, controlada por un servidor central que también captura imágenes con las cámaras y estima la posición mediante triangulación. El error de posicionamiento medio es 9 cm, con un máximo de 30 cm, mientras que el tiempo de identificación del usuario es de 4s, con la capacidad de procesar dos imágenes por segundo. El diagrama del sistema puede verse en la figura 2.32.

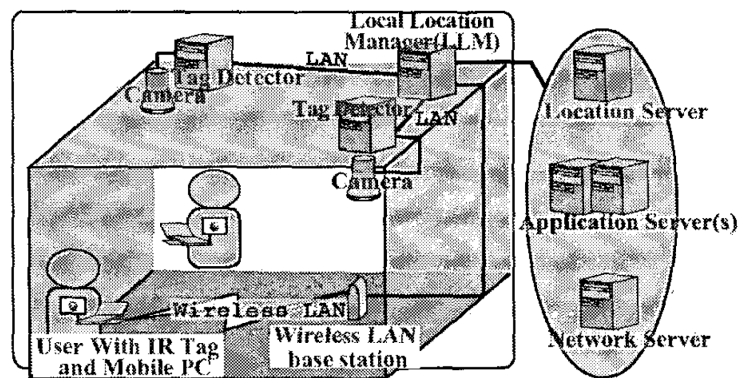


Figura 2.32. Esquema de funcionamiento de ALTAIR [95].

Otro trabajo que utiliza triangulación es [72], del que ya se habló en la sección 2.4.2.

2.4.5. Trilateración

En su versión más simple, el algoritmo de trilateración parte de la estimación de la distancia entre tres emisores y un receptor en dos dimensiones. En este caso la altura a la que se encuentran los tres emisores es igual, y la altura a la que se encuentra el receptor también es conocida. Las distancias obtenidas se utilizan como radios para describir tres círculos. El punto de intersección determina la posición del receptor, como puede verse en la figura 2.33a. Para encontrar dicho punto basta con resolver un sistema de ecuaciones como el presentado en la ecuación (2.14).

$$\begin{aligned}
(x - x_A)^2 + (y - y_A)^2 &= d_A^2 \\
(x - x_B)^2 + (y - y_B)^2 &= d_B^2 \\
(x - x_C)^2 + (y - y_C)^2 &= d_C^2
\end{aligned}
\tag{2.14}$$

En la ecuación (2.14) $[x_A, x_B, x_C]$ e $[y_A, y_B, y_C]$ son las coordenadas en dos dimensiones de las luminarias A, B y C, mientras que $[d_A, d_B, d_C]$ son las distancias horizontales entre estas luminarias y el receptor.

El caso anterior es el ideal. Sin embargo, los errores introducidos en el sistema por factores como la difusión de la luz, los ruidos introducidos en la señal por otras fuentes luminosas presentes, aberraciones en los sensores ópticos, una distribución inapropiada de los emisores, etcétera, impiden que la solución sea un único punto, convirtiéndose en un área que se puede calcular mediante una estimación de mínimos cuadrados [64]. En el caso de la figura 2.33b, el resultado es el área sombreada S .

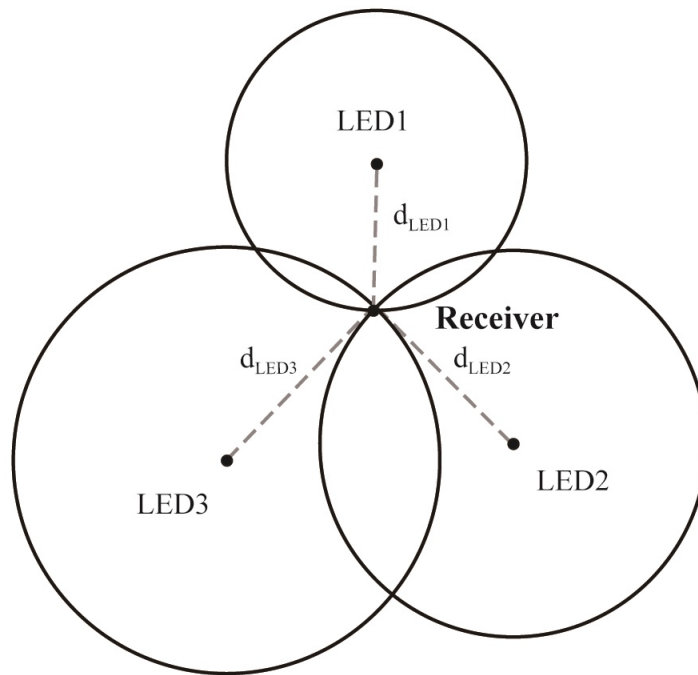
Partiendo del sistema presentado en la ecuación (2.14), es posible obtener una estimación de la posición del receptor solucionado el sistema que se muestra en la ecuación (2.15).

$$\begin{aligned}
2x(x_A - x_C) + x_C^2 - x_A^2 + 2y(y_A - y_C) + y_C^2 - y_A^2 &= d_C^2 - d_A^2 \\
2x(x_B - x_C) + x_C^2 - x_B^2 + 2y(y_B - y_C) + y_C^2 - y_B^2 &= d_C^2 - d_B^2
\end{aligned}
\tag{2.15}$$

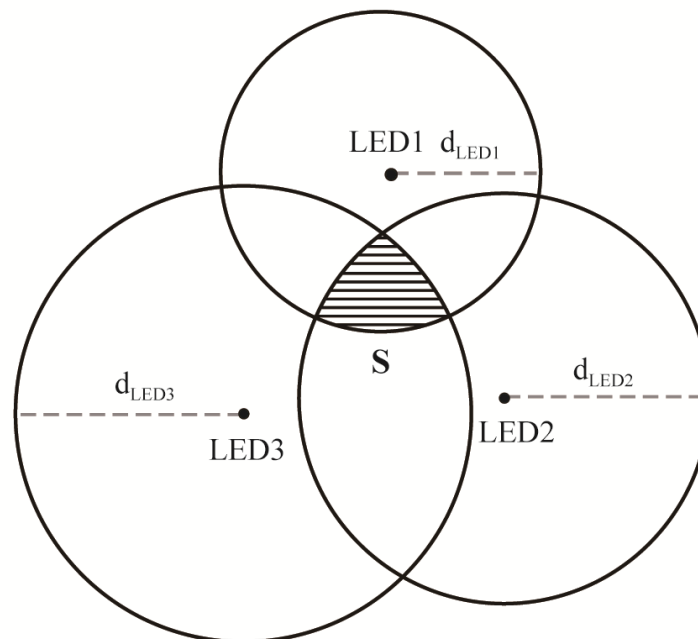
El sistema se complica un poco más cuando añadimos la altura a las incógnitas que es necesario despejar. En este caso necesitaremos al menos cuatro luminarias a la vista en lugar de tres. En lugar de encontrar la intersección de tres circunferencias (sea como punto, sea como área), estaremos buscando la intersección de cuatro esferas, esperando un resultado equivalente [96].

2.4.6. Geometría de vista única

Según [90], los sistemas de posicionamiento basados en geometría de vista única utilizan una sola cámara para capturar varias fuentes luminosas, como puede verse en la figura 2.34. Los LED utilizados en las fuentes luminosas utilizan un sistema de comunicación mediante luz visible (ya descritos en la sección 2.3) para enviar sus coordenadas 3D en el mundo real, o para enviar un identificador único que permita al receptor obtenerlas. La relación geométrica de estos datos con las coordenadas 2D



(a) Con punto de intersección.



(b) Con área de intersección.

Figura 2.33. Trilateración [10].

en la imagen capturada de las fuentes luminosas se utiliza para deducir la posición de la cámara.

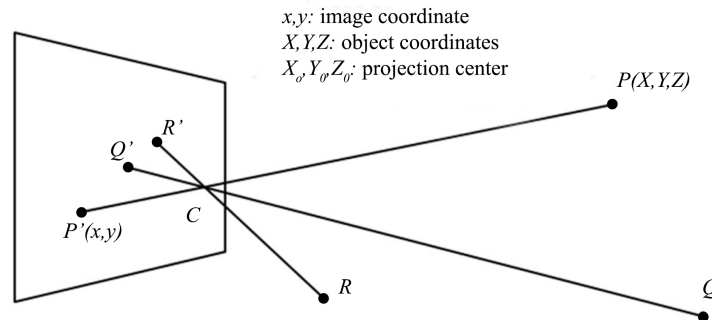


Figura 2.34. Geometría de vista única [90].

Uno de los trabajos con más impacto en este ámbito es el pionero Luxapose² [3], presentado en 2014. El objetivo del mismo consiste en estimar la ubicación y orientación de un smartphone a partir de tres o más fuentes luminosas, suponiendo que se conocen sus coordenadas 3D y las características de la cámara en el receptor (es decir, el tamaño del píxel, su distancia focal, las dimensiones de la imagen capturada, etcétera). Su algoritmo de posicionamiento se basa el modelo de cámara estenopeica descrito en la figura 2.35 y en la utilización de un factor de escala para estimar la pose de la cámara.

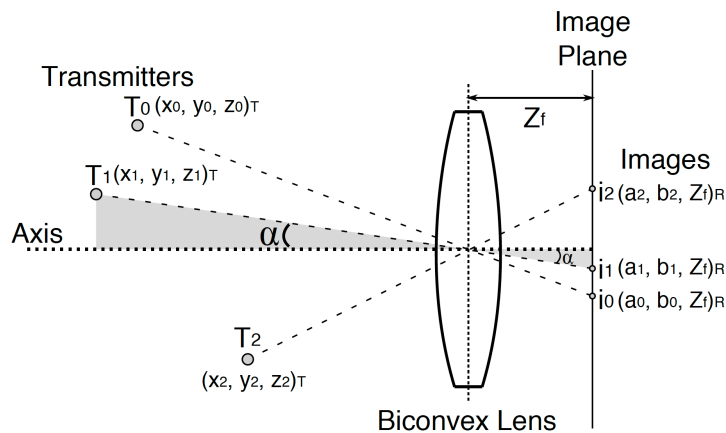


Figura 2.35. Modelo de cámara estenopeica utilizado en Luxapose [3].

Se supone que las coordenadas del emisor T_0 en el marco de referencia global son $(x_0, y_0, z_0)_T$. Dicho transmisor ha sido capturado en la imagen i_0 , y sus coordenadas en dicha imagen son $(a_0, b_0, Z_f)_R$, donde Z_f es la distancia en píxeles de la lente al

²Al contrario de lo que uno pensaría tras ver un trabajo tan influyente, no parece que ninguno de los autores del mismo haya continuado esa línea de investigación o, al menos, no públicamente.

sensor, es decir, la distancia focal. La posición del emisor T_0 está en la línea que va de $(0, 0, 0)_R$ a $(a_0, b_0, Z_f)_R$. Utilizando las reglas geométricas de similitud de triángulos es posible definir un factor de escala K_0 para el emisor T_0 que permita describir la ubicación del mismo en el sistema de referencia del receptor como $(u_0, v_0, w_0)_R$, como en la ecuación (2.16).

$$\begin{aligned} u_0 &= K_0 \times a_0 \\ v_0 &= K_0 \times b_0 \\ w_0 &= K_0 \times Z_f \end{aligned} \quad (2.16)$$

Al ser conocidas las posiciones de los emisores, es posible expresar la distancia entre ellas tanto en el sistema de referencia del emisor como en el del receptor. Así, la distancia entre los emisores T_0 y T_1 , expresada como $d_{0,1}^2$, puede verse en la ecuación (2.17):

$$\begin{aligned} d_{0,1}^2 &= (u_0 - u_1)^2 + (v_0 - v_1)^2 + (w_0 - w_1)^2 \\ &= (K_0 a_0 - K_1 a_1)^2 + (K_0 b_0 - K_1 b_1)^2 + Z_f^2 (K_0 - K_1)^2 \\ &= K_0^2 |O_{i_0}|^2 + K_1^2 |O_{i_1}|^2 - 2K_0 K_1 (O_{i_0} \cdot O_{i_1}) \\ &= (x_0 - x_1)^2 + (y_0 - y_1)^2 + (z_0 - z_1)^2 \end{aligned} \quad (2.17)$$

donde O_{i_0} y O_{i_1} son los vectores que van del centro de la lente a las coordenadas (a_0, b_0, Z_f) de la imagen i_0 y (a_1, b_1, Z_f) de la imagen i_1 , respectivamente. Las únicas incógnitas son K_0 y K_1 . Cada emisor proporciona tres ecuaciones cuadráticas con tres incógnitas, lo que permite estimar K_0 , K_1 y K_2 y calcular así la ubicación de los mismos en el marco de referencia del receptor.

En la práctica, [3] afirma que estimar a la perfección la ubicación del receptor no es realista debido a las imperfecciones ópticas del receptor y a la falta de precisión en la obtención de la ubicación de los emisores en las imágenes capturadas. Para intentar mitigar estos factores se intenta afrontar el problema buscando el mínimo error cuadrático medio de una serie de factores de escala, tal y como muestra la ecuación (2.18):

$$\sum_{m=1}^{N-1} \sum_{n=m+1}^N \left\{ K_m^2 |\vec{O}_{i_m}|^2 + K_n^2 |\vec{O}_{i_n}|^2 - 2K_m K_n (\vec{O}_{i_m} \cdot \vec{O}_{i_n}) - d_{mn}^2 \right\}^2 \quad (2.18)$$

donde N es el número de emisores que aparecen en la imagen capturada, con lo que se tienen $\binom{N}{2}$ ecuaciones.

Una vez estimados los factores de escala es posible obtener la posición de los emisores en el marco de referencia del receptor y, a partir de ahí, calcular las distancias entre ambos. La relación entre ambos dominios puede expresarse como se ve en la ecuación (2.19):

$$\begin{bmatrix} x_0 & x_1 & \dots & x_{N-1} \\ y_0 & y_1 & \dots & y_{N-1} \\ z_0 & z_1 & \dots & z_{N-1} \end{bmatrix} = \mathbf{R} \times \begin{bmatrix} u_0 & u_1 & \dots & u_{N-1} \\ v_0 & v_1 & \dots & v_{N-1} \\ w_0 & w_1 & \dots & w_{N-1} \end{bmatrix} + \mathbf{T} \quad (2.19)$$

donde la ecuación (2.19) \mathbf{R} es una matriz de rotación de dimensiones 3×3 y \mathbf{T} es una matriz de traslación de dimensiones 3×1 . \mathbf{T} representa la posición del receptor según el marco de referencia de los emisores. Los factores de escala son desconocidos, por lo que se emplean distancias equivalentes para obtener la posición del receptor en el marco de referencia de los emisores, tal y como muestra la ecuación (2.20).

$$(T_x - x_m)^2 + (T_y - y_m)^2 + (T_z - z_m)^2 = K_m^2 (a_m^2 + b_m^2 + Z_f^2) \quad (2.20)$$

En la ecuación (2.20) (x_m, y_m, z_m) son las coordenadas del emisor m en el marco de referencia de los emisores y (a_m, b_m) es la proyección del emisor m en el plano de la imagen capturada. Por último, la posición del receptor se estima encontrando el conjunto de valores (T_x, T_y, T_z) que minimiza la ecuación la ecuación (2.21).

$$\sum_{m=1}^N \left\{ (T_x - x_m)^2 + (T_y - y_m)^2 + (T_z - z_m)^2 - K_m^2 (a_m^2 + b_m^2 + Z_{fm}^2) \right\}^2 \quad (2.21)$$

Conocida \mathbf{T} , el paso restante para obtener la pose de la cámara es determinar su orientación, la matriz \mathbf{R} con dimensiones 3×3 que se puede ver en la ecuación (2.22).

$$\mathbf{R} = \begin{bmatrix} r_1 & r_2 & r_3 \end{bmatrix} \quad (2.22)$$

En la ecuación (2.22) los vectores columna \mathbf{r}_1 , \mathbf{r}_2 y \mathbf{r}_3 son los componentes de los vectores unitarios \hat{x}' , \hat{y}' y \hat{z}' , respectivamente, proyectados en los ejes x , y y z del marco de referencia de los emisores.

El prototipo desarrollado para probar el planteamiento matemático anterior puede verse en la figura 2.36.



Figura 2.36. Prototipo de Luxapose para pruebas [3].

Cada uno de los cinco emisores está compuesto por una luminaria LED comercial modificada para emitir un identificador único de 8 bits utilizando codificación Manchester con una frecuencia en el rango de 2 a 4 kHz. Estos emisores se encuentran a 246 cm de altura; cuatro de ellos están en cada esquina de un rectángulo de $71,1 \times 73,7 \text{ cm}^2$, con el quinto situado en el centro. En el suelo se ha colocado una malla cuadrada, alineada con el rectángulo de emisores, en la que cada celda tiene 2,54 cm de lado. Los resultados experimentales demuestran que Luxapose es capaz de lograr precisión decimétrica y un error de 3° de orientación.

Sin embargo, Luxapose presenta una carencia fundamental: el procesamiento de las imágenes capturadas no se realiza en el propio dispositivo que las captura. El dispositivo móvil (un Nokia Lumia 1020) toma la fotografía y la envía a un servidor (un MacBook Pro con un procesador Core i7 a 2,7 GHz) dedicado en exclusiva a la tarea de procesarlas y estimar la pose del receptor. El proceso completo (tomar una fotografía, enviarla al servidor, procesarla y estimar la pose del receptor) lleva unos 9 s de media.

En [97] se describe un sistema de posicionamiento en interiores que supone una evolución de Luxapose en varios aspectos. En primer lugar, y probablemente el más importante: el procesamiento se realiza en el propio dispositivo móvil utilizado como receptor. El tiempo de procesamiento está entre 22,7 y 35,7 ms, dependiendo de las condiciones. La precisión del sistema llega a los 7,5 cm, por debajo del decímetro de Luxapose. Además, incluye un sistema para disminuir el parpadeo y la atenuación de la luz que se provoca al modular el medio luminoso para emitir datos a través de él, algo que Luxapose no tenía en cuenta.

Para determinar la pose del receptor se utiliza el algoritmo propuesto en [98], que sólo precisa dos luminarias. En los resultados experimentales de este último el error máximo era de 6,5 cm dentro de una zona rectangular de 110 cm × 80 cm, con las luminarias situadas a 245 cm de altura. En la figura 2.37 puede verse dicho banco de pruebas.

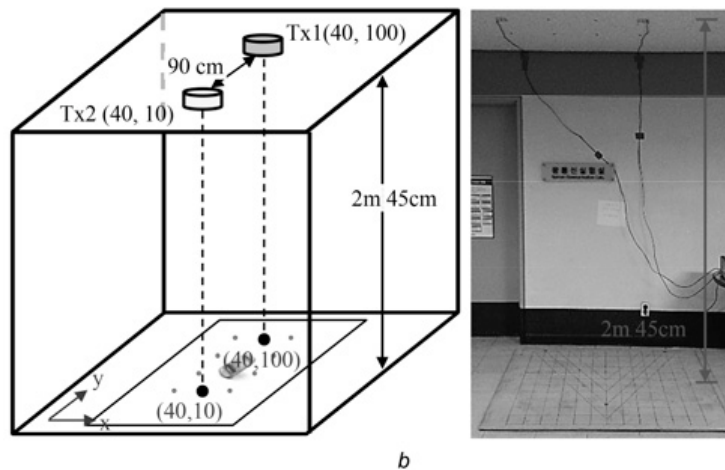


Figura 2.37. Banco de pruebas de [98].

Poder utilizar sólo dos LEDs para realizar el posicionamiento es especialmente interesante en zonas angostas como, por ejemplo, pasillos.

Esta sección se cierra con el trabajo de Y. Ji *et al.* [99], probablemente el más interesante de los vistos hasta ahora en lo que a esta tesis se refiere. En él se propone un sistema de posicionamiento basado en luz visible con una única luminaria LED capaz de realizar posicionamiento con precisión centimétrica en 3D. Utilizando un panel LED comercial de 7 W y 7,3 cm de diámetro a 1 m de altura, este sistema logra posicionar en 2D con 2,04 cm de error, y en 3D con 3,96 cm.

El trabajo está dividido en dos partes: en la primera se describe el sistema y en la segunda se realizan pruebas experimentales para evaluar la eficacia del mismo. La descripción del sistema se centra en varios aspectos:

- la transmisión de datos.
- la detección de la baliza.
- la estimación de la posición del receptor.

Cada baliza está compuesta por una luminaria LED y una FPGA para controlar su funcionamiento. Dicha FPGA tiene almacenada en su memoria de solo lectura (*Read-Only Memory*, ROM) el código que tiene que emitir la baliza, y modula la señal

emitida por la luminaria, transmitiendo así el mensaje correspondiente. El código, de 25 bits, contiene detalles sobre la naturaleza de la baliza como su ubicación y su forma. En la figura 2.38 puede verse la estructura de datos emitida por la baliza.

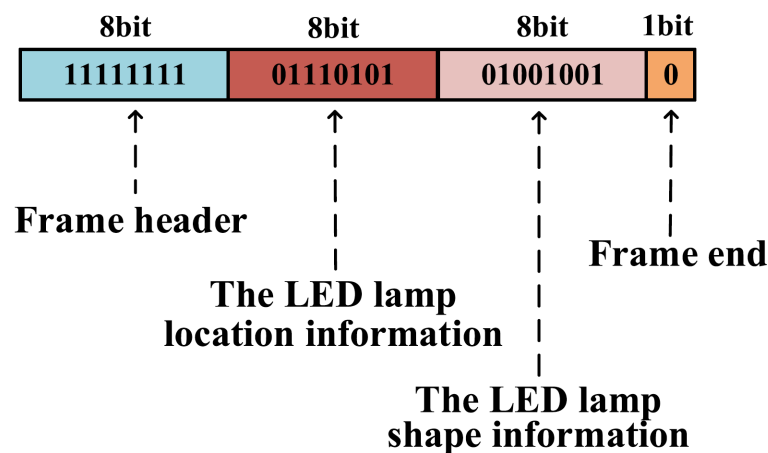


Figura 2.38. Estructura de datos emitida por la baliza de [99].

Para mejorar la detección del código emitido por la baliza, el receptor utiliza curvas de umbralización dinámicas mediante ajuste polinómico de segundo orden. En la figura 2.39 puede verse el valor medio en escala de grises de cada columna de píxeles de la imagen original (en azul). En rojo, la curva obtenida mediante el ajuste polinómico de segundo orden. La tasa de acierto en el proceso de decodificación es del 100 % a 1 m, del 95 % a 1,2 m, del 80 % a 1,4 m y del 10 % a 2 m.

Una vez detectado el código emitido por la baliza, el sistema pasa a buscar la propia baliza en la imagen capturada por el smartphone. Para ello, se vale de un ajuste por mínimos cuadrados combinado con el operador de Sobel [100]. La detección del contorno de la baliza dentro de la imagen capturada es importante, puesto que de la precisión de esta operación depende la posterior estimación de la posición del receptor.

Combinando los datos del contorno de la baliza dentro de la imagen capturada con los datos de inclinación y orientación del smartphone, es posible utilizar los principios de geometría tridimensional y de formación de imágenes utilizando lentes para estimar la ubicación del receptor. En la figura 2.40 se pueden ver todos los elementos necesarios para estimar la posición del receptor, incluido el proceso de corrección de inclinación previo.

Si el smartphone no se encuentra inclinado con relación al plano horizontal, y está a una distancia de 1 m de la baliza, el error medio de posicionamiento en 2D/3D es de 2,04 cm/3,96 cm, respectivamente. Por otra parte, con inclinaciones de hasta 15°, el

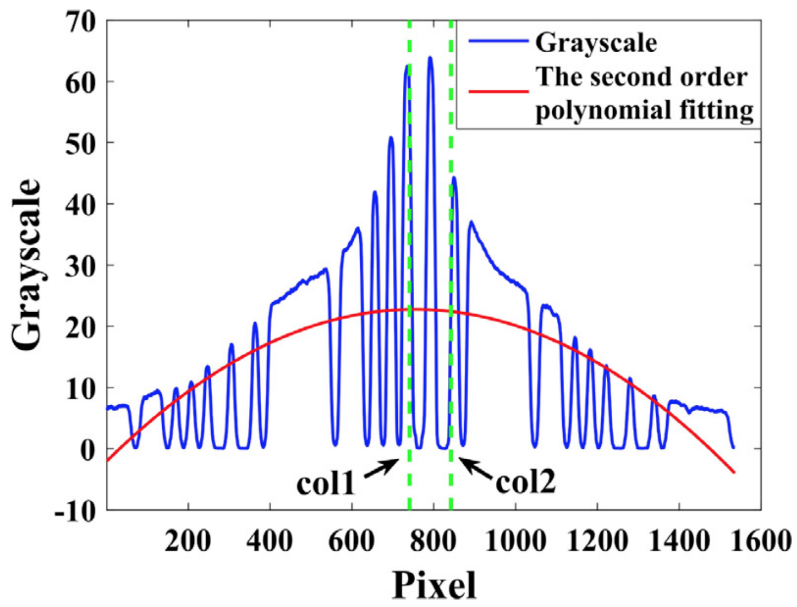


Figura 2.39. Mejora en la obtención de los umbrales para la extracción del código emitido por la baliza de [99].

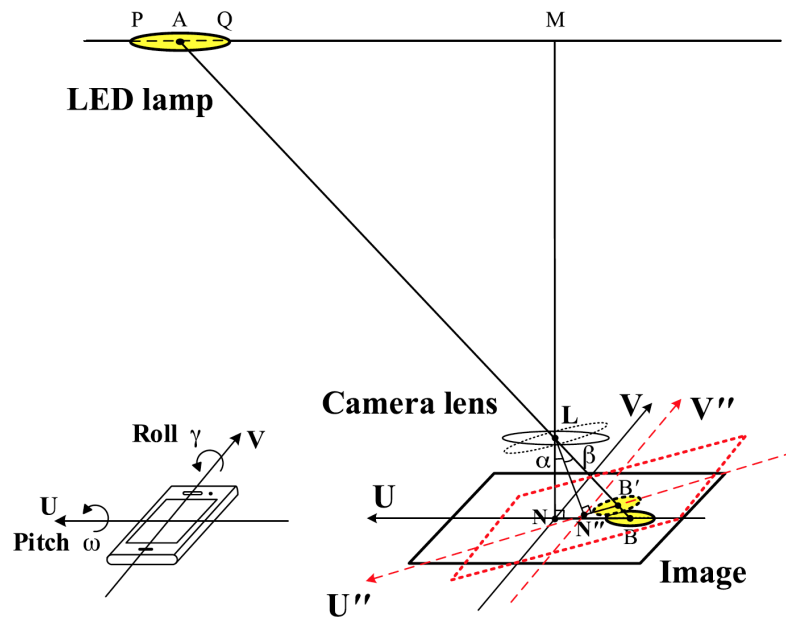


Figura 2.40. Elementos necesarios para la estimación de la posición del receptor en [99].

error medio se mantiene entorno a los 4 cm. En la figura 2.41 se pueden observar los resultados obtenidos con diferentes ángulos de cabeceo (*pitch*) y alabeo (*roll*) del receptor.

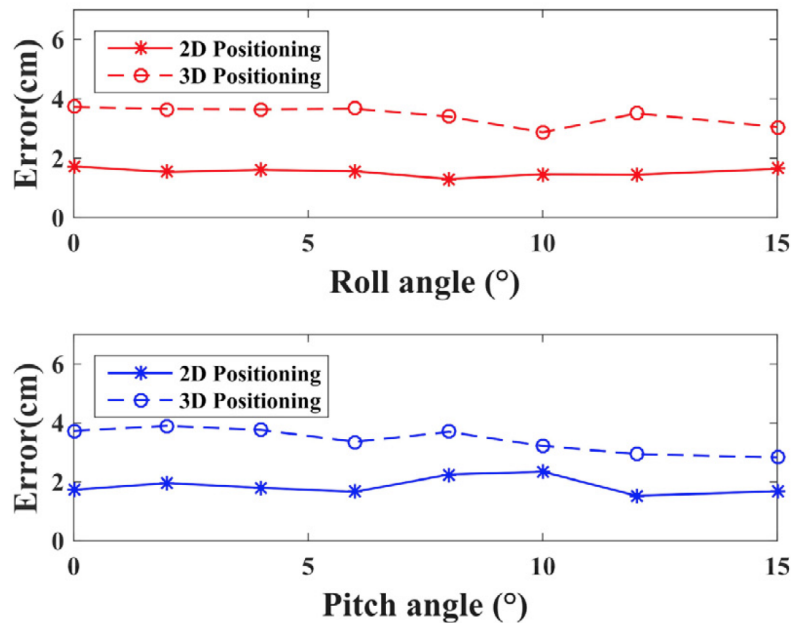


Figura 2.41. Errores medios de posicionamiento en función de la inclinación del receptor en [99].

Dadas las premisas presentadas en la introducción de este trabajo de tesis la sección ??, la aproximación con más garantías de éxito parece ser utilizar análisis de visión de geometría de vista única para posicionar un dispositivo portable en interiores utilizando la ayuda de luminarias LED, dado que la mayoría de los dispositivos portables actuales sólo ofrecen una cámara frontal al usuario. Ahora bien, cabe preguntarse cómo identificará el sistema cada una de las luminarias y cómo obtendrá las coordenadas de las mismas en el mundo real. Para conocer la respuesta es necesario analizar la situación actual y los últimos avances en VLC.

2.5. Objetivos

Tras estudiar la situación actual en el mundo de la comunicación y el posicionamiento en interiores con luz visible nos encontramos en disposición de marcar los objetivos que este trabajo de tesis debería cumplir. El primer paso consiste en desarrollar un sistema de transmisión de datos entre emisor y receptor. Parece claro que la alternativa más viable consiste en utilizar paneles LED comerciales. Es

la opción más asequible y resulta posible hacerlos oscilar a frecuencias de varios kHz (ver la sección 2.3). Además, una codificación OOK puede ofrecer tasas de transmisión aceptables, aunque para poder aprovechar luminarias ya desplegadas como infraestructura de iluminación para nuestros propósitos será necesario incorporar algún mecanismo que permita mantener la intensidad de la luz idéntica, independientemente del código emitido.

Sería deseable diseñar una baliza en consonancia con las características de las luminarias LED, es decir, con reducido impacto medioambiental, requisitos energéticos mínimos y bajo coste. La tecnología LED es moderna y la baliza también debería serlo. Lo ideal sería que, una vez desplegadas, las balizas fuesen configurables sin necesidad de desmontarlas o acceder físicamente a ellas. Por último, el diseño de las mismas debería ser replicable con facilidad, para facilitar el trabajo propio y permitir que cualquiera pudiese crear su propia versión de las mismas.

En cuanto al posicionamiento, como se comenta en [71], cuando el receptor utiliza PD con la técnica de AOA se puede estimar la posición con gran precisión, pero a costa de utilizar una matriz de sensores en el receptor, lo que puede resultar muy costoso. Por otra parte, resulta complicado alcanzar resultados precisos mediante la técnica de RSS, principalmente por los efectos negativos de los reflejos cerca de las paredes. El tiempo de viaje de la luz en interiores es demasiado rápido debido a la corta distancia que separa emisor y receptor, lo que complica la utilización de la técnica de TOA por la necesidad de una sincronización muy precisa entre ambos. Sin embargo, TDOA sólo precisa sincronización entre los emisores, algo fácil de conseguir puesto que los paneles LED pueden compartir el mismo reloj en la misma habitación. La diferencia en el tiempo de llegada puede obtenerse con precisión si se tiene un esquema de transmisión de datos adecuado. Es más, un sistema con TDOA sólo requiere un PD en el receptor, que es relativamente asequible. Por lo tanto, la técnica de TDOA parece la más adecuada para un sistema de posicionamiento en interiores mediante luz visible basada en el uso de PD.

Sin embargo, esta tesis pretende llevar el posicionamiento en interiores a la mayor cantidad de público posible, razón por la que se utiliza como receptor la cámara de un dispositivo portable. Por lo tanto, se ha de descartar la utilización de PDs. Trabajos como [99] confirman que es posible lograr conseguir posicionamiento preciso, del orden de los centímetros, con una única baliza como referencia.

El último objetivo es, a la vez, un objetivo global: desarrollar todas las herramientas software necesarias para que el sistema funcione y hacerlas públicas. Así podrán ser examinadas y, en el mejor de los casos, aprovechadas por quienes quieran sacarles

partido. Se optará por utilizar herramientas ya existentes y, por qué no, extenderlas para que este trabajo de tesis pueda realizar todas las aportaciones a su alcance.

Comunicación de datos

3.1. Introducción

Como se comentó al finalizar el capítulo anterior (ver la sección 2.5), el objetivo de este trabajo de tesis consiste en el desarrollo de un sistema de posicionamiento en interiores en el que las balizas utilizan luz visible generada mediante LED y el receptor es un dispositivo portable que se apoya en los diferentes sensores a su disposición para poder estimar la posición del usuario. Un requisito indispensable para cumplir este objetivo es disponer de un sistema de comunicación robusto, capaz de transmitir pequeñas cantidades de datos desde la baliza hasta el receptor. Sólo entonces se podrá identificar de forma única cada baliza, sentando las bases para la finalidad última de este trabajo: el desarrollo de un sistema de posicionamiento.

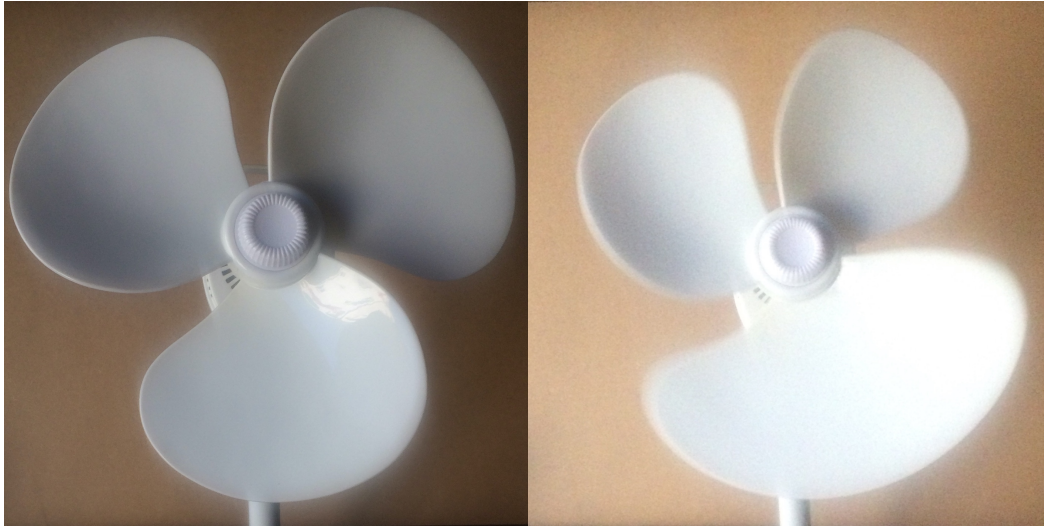
Para conseguir un sistema de comunicación robusto entre emisor y receptor se ha trazado un plan de trabajo muy concreto. En primer lugar, se elegirá un sistema de codificación adecuado al medio de transmisión y a los elementos emisor y receptor. Dado que se ha escogido como sistema de transmisión una luminaria LED para interiores, la codificación no debe interferir con la misión principal de las luminarias. Es decir, las personas que se encuentren en las proximidades de las balizas no deben percibir que una luminaria está emitiendo un código y, además, no deben apreciarse diferencias entre luminarias emitiendo códigos diferentes.

A continuación se explicará el método seguido para la elección de la codificación utilizada, así como la caracterización del sistema completo.

3.2. El efecto *rolling shutter*

La inmensa mayoría de los dispositivos portables actuales utiliza una cámara de tipo CMOS susceptible de presentar el efecto *rolling shutter*. Esto significa que el sensor utilizado para capturar la imagen lo hace línea a línea, en lugar de capturar toda la escena de una vez [101]. Como consecuencia, cualquier elemento en movimiento dentro de la escena puede aparecer distorsionado en la fotografía resultante. Cuanto

mayor sea la rapidez del movimiento, mayor la distorsión. Este efecto se puede apreciar en la figura 3.1, donde un objeto en rápido movimiento no tiene el mismo aspecto que estando en reposo. En dicha figura ambas fotografías fueron tomadas con la cámara trasera de un iPhone 5s.



(a) Ventilador en reposo.

(b) Ventilador en movimiento.

Figura 3.1. El efecto *rolling shutter*.

Sin embargo, es posible sacar partido de este defecto. Si se construye un sistema capaz de emitir luz que parpadee con una frecuencia superior a la que el sensor **CMOS** pueda capturar, se podría codificar un mensaje utilizando la ausencia o presencia de luz. Posteriormente la muestra tomada podría analizarse para intentar extraer el mensaje [3][102][103].

La elección de la frecuencia de oscilación supone un desafío interesante. Si está por debajo de la frecuencia de captura del sensor **CMOS** será apreciable por el ojo humano, e incluso puede llegar a ser perjudicial para aquellos con problemas de fotosensibilidad. Por otra parte, si es muy superior a la frecuencia de captura del sensor **CMOS**, éste no será capaz de obtener ningún dato a partir de la señal. Si ambas frecuencias son exactamente iguales, el sensor detectará siempre la señal o su ausencia, dependiendo la sincronización entre emisor y receptor.

Antes de poder decodificar los códigos emitidos por las balizas es necesario tomar fotografías adquiridas por la cámara del dispositivo portable. Dichas fotografías deben tomarse en ciertas condiciones; en concreto, el tiempo de exposición se ajusta al mínimo mientras que la sensibilidad de la cámara lo hace al máximo. El propósito de estos cambios es doble:

1. En el contenido de las imágenes adquiridas se da prioridad a la parte más iluminada de la escena, que debería ser la que contenga luminarias. En la figura 3.2a se puede observar el resultado de tomar una fotografía en condiciones normales, mientras que en la figura 3.2b se ve el obtenido al ajustar la cámara, como se acaba de mencionar. Se puede comprobar que en la segunda de las imágenes el entorno es menos prominente que la luminaria.
2. Estos ajustes permiten que en las muestras adquiridas sea posible diferenciar los cambios de estado de la fuente emisora. Por las características de las cámaras CMOS que incorporan la inmensa mayoría de los dispositivos portables actuales, dichos cambios de estado aparecen convertidos en barras.



(a) Configuración predeterminada.

(b) Exposición y sensibilidad ajustadas.

Figura 3.2. Resultado de ajustar tiempo de exposición y sensibilidad de la cámara.

Este efecto se aprovechará para crear el sistema de codificación utilizado en este trabajo de tesis. En la sección 4.4 se describe la aplicación de escritorio desarrollada para realizar el análisis *offline* de las muestras, mientras que en la sección 4.5 se describe la aplicación Android desarrollada para la toma de muestras y el análisis *online* de las mismas.

3.3. Elección del sistema de codificación

Un detalle crucial al escoger un sistema de codificación es que no haya grandes diferencias en la luminosidad de las balizas dependiendo del código emitido [14]. Es necesario comprender que cada 1 emitido implica que la baliza está encendida, mientras que para cada 0 la baliza está apagada. Aunque estos cambios se realicen a muy altas frecuencias, el resultado final es que si hay más unos que ceros en el código identificador de una baliza, ésta se percibirá más brillante que otra con un número inferior de unos. El objetivo es lograr que la iluminación de todas las balizas sea homogénea. En la sección 2.3 del capítulo dedicado a describir la situación actual en el mundo de la comunicación y el posicionamiento con luz visible se describe en detalle todo lo relacionado a la utilización de la luz visible para la transmisión de datos, prestando especial atención a la tecnología LED.

Para alcanzar este objetivo se utilizará un código bifásico de marca (*Biphase Mark Code*, BMC). Esta codificación, también conocida como Manchester, se utiliza para la representación digital de datos [104]. Cada valor transmitido utiliza el estado anterior (alto o bajo) como entrada. El valor cero mantendrá ese estado durante dos ciclos, mientras que el valor uno mantiene el valor opuesto al anterior durante un ciclo, cambiando al anterior en el siguiente ciclo. En la figura 3.3 se pueden ver ejemplos de esta técnica, que será descrita con más detalle a continuación.

Si el emisor utilizase dos bits para la codificación podría enviar cuatro valores diferentes, entre 0 y 3 en base 10. Es decir, los valores serían 00, 01, 10 y 11 en base 2. Tras convertir esos valores en base 2 a la codificación seleccionada, la baliza los emitiría de forma cíclica. En estas circunstancias al receptor le resulta imposible determinar dónde comienza el mensaje. Este problema puede evitarse introduciendo un separador o cabecera, usando un patrón que no esté presente en los datos transmitidos. Un candidato ideal podría ser el valor 01110, puesto que las reglas de la codificación Manchester impiden que una secuencia de tres unos aparezca de forma natural. De esa manera, la cabecera siempre puede diferenciarse del resto de la transmisión. En la figura 3.3 se muestra cómo codificar estos cuatro valores, separador incluido. A la derecha de cada valor se incluye un ejemplo del resultado.

3.4. Proceso de decodificación

El proceso de decodificación de cada muestra está compuesto por dos fases:

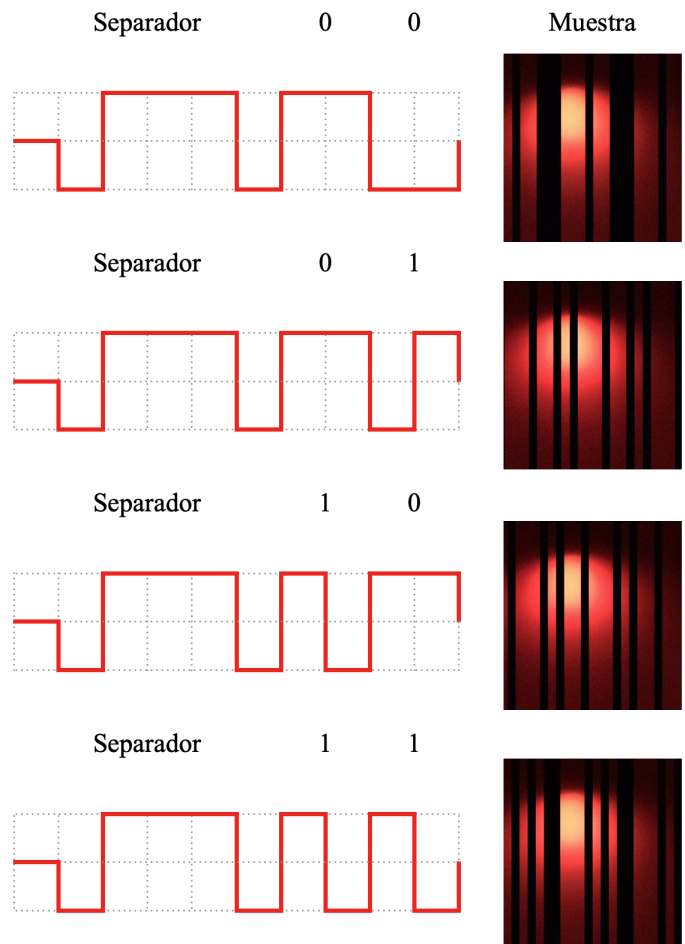


Figura 3.3. Valores del 0 al 3 codificados con dos bits usando Manchester, separador incluido. Se proporciona un ejemplo de cada secuencia [105].

- **procesamiento de la imagen:** el resultado es una serie de datos de los que se obtiene el código emitido.
- **procesamiento de los datos:** el resultado es una serie de códigos, asociados a un índice de redundancia asignado al proceso de decodificación.

El procesamiento de la imagen consta, a su vez, de varios pasos:

- **carga de la imagen:** la muestra a analizar se encuentra almacenada en un archivo, en disco. El primer paso del proceso de decodificación consiste en cargar dicha imagen.
- **giro de la imagen sobre el eje vertical:** por el modo de funcionamiento de las cámaras con sensor **CMOS**, el código emitido avanza de derecha a izquierda. La mejor manera de conseguir que el procesamiento de la imagen sea fácil de comprender es rotar la imagen sobre el eje vertical. A partir de ese momento el código se leerá de izquierda a derecha.
- **conversión de la imagen a escala de grises:** las muestras adquiridas están en formato **RGB**. Sin embargo, los componentes cromáticos de la imagen no aportan nada al proceso de decodificación. Para simplificarlo se convierte la imagen a escala de grises, quedando sólo los componentes de intensidad de la imagen.
- **suavizado de la imagen:** un leve suavizado eliminará ruidos en la imagen, optimizando el resultado de los siguientes pasos.
- **primera umbralización de la imagen:** para poder decodificar la imagen es necesario trabajar con ceros y unos, en lugar de con valores entre 0 y 255. Utilizando la umbralización de Otsu [106] es posible realizar esta conversión de forma óptima gracias a su análisis del histograma de la imagen, sin necesidad de establecer dicho umbral de forma manual.
- **reducción de la imagen:** la imagen que llega a este paso es un código de barras en blanco y negro. Las líneas de este código, por la manera en la que se combina la oscilación del emisor y el efecto *rolling shutter* del receptor, son perfectamente verticales. La mayoría de artefactos producidos por el ruido habrán desaparecido ya. Al aplicar un proceso de reducción por columnas a la imagen utilizando valores medios, el resultado será una única fila en la que cada punto será resultado de una operación realizada sobre todos los puntos de una columna. Aquellas columnas con muchos valores blancos tendrán una intensidad más alta que aquellas en las que hubiese muchos valores oscuros.

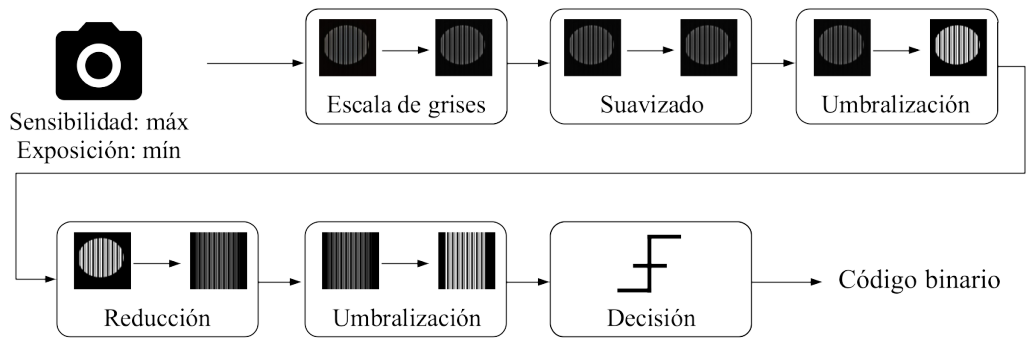
Para facilitar su visualización, esta fila se ha expandido para verticalmente en la figura 3.4b.

- **segunda umbralización de la imagen:** como resultado del proceso de reducción la imagen volverá a estar en escala de grises. Esta vez, puesto que la imagen sobre la que se va trabajar está más controlada (es decir, no se trata de la muestra tomada), no se utilizará umbralización de Otsu sino binaria, resultando otra vez en una imagen en blanco y negro. El punto de corte está en la mitad de la intensidad máxima menos uno. Como en el paso anterior, la fila se ha expandido para verticalmente en la figura 3.4b para facilitar su visualización.

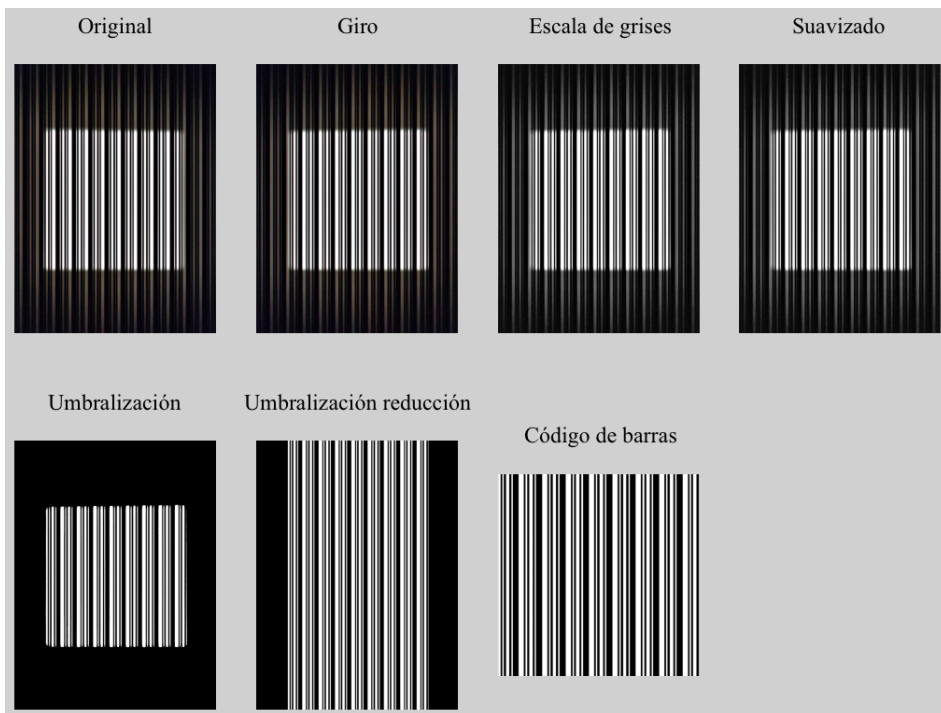
Estos pasos pueden apreciarse mejor en el diagrama de funcionamiento del receptor mostrado en la figura 3.4a. El resultado de procesar una muestra se puede ver en la figura 3.4a, donde se pasa de una muestra tomada con la cámara de un dispositivo portable a un vector con ceros y unos que se podrá procesar para extraer el mensaje emitido.

La fase de procesamiento de datos también consta de varios pasos:

- **obtención de bits y frecuencias:** se recorre la lista de unos y ceros, guardando para cada valor la frecuencia con la que aparece cada uno.
- **rellenado de los huecos:** por las características de las cámaras de los dispositivos portables, en ocasiones y durante breves instantes, el sensor no captura ningún valor, aunque sí que exista información luminosa. Así, por ejemplo, en la figura 3.5 se pueden apreciar algunos de estos huecos. Dos de ellos han sido marcados en rojo. Para evitar que estos huecos interfieran en la detección del código se rellenarán con el valor encontrado anterior y posteriormente, es decir, el blanco.
- **creación de una imagen representativa:** este paso no es obligatorio, pero sirve de ayuda para el posterior estudio del proceso de decodificación. A partir de la serie de ceros y unos obtenida en el paso anterior se crea una imagen que se corresponderá con el código de barras capturado por la cámara del dispositivo portable tras todos los pasos de procesamiento dados hasta ahora. Examinar dicha imagen a simple vista resultará mucho más sencillo que hacer lo propio con la serie de ceros y unos obtenida hasta el momento.
- **obtención de la secuencia de bits:** a partir de la serie de ceros y unos se obtendrá la secuencia de bits emitida por la baliza. No existe una correspondencia directa entre el valor emitido por la baliza y el captado por la cámara



(a) Diagrama de funcionamiento.



(b) Pasos del procesamiento de una muestra.

Figura 3.4. Funcionamiento del receptor.



Figura 3.5. Huecos en la muestra.

del dispositivo portable. Será necesario buscar un valor de referencia que permita determinar cuántos de los bits recibidos se corresponden con cada bit emitido por la baliza. Este ancho se obtiene a partir de la cabecera: la secuencia de unos más larga se corresponde con 111 (de la cabecera 01110).

- **obtención del código emitido:** dentro de la secuencia de bits con la que estemos trabajando se buscará la cabecera. A partir de la cabecera se buscarán tantos bits como se estén emitiendo y se procederá a decodificarlos. El valor obtenido será el código emitido por la baliza.

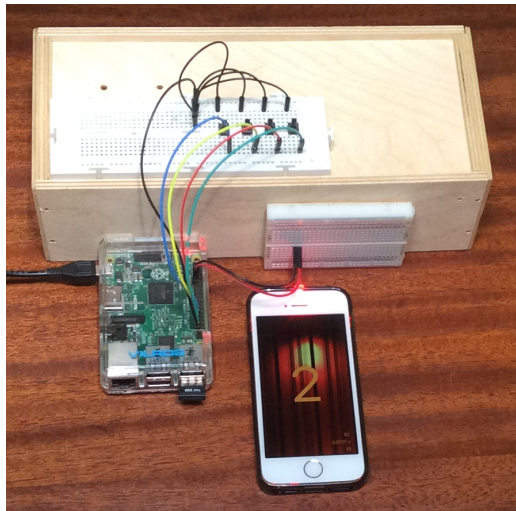
3.5. Baliza

En este apartado se describe la baliza utilizada durante la fase de desarrollo del sistema de comunicación de datos. Aunque para llegar a su diseño final ha sido necesario pasar por varios prototipos, todos ellos tienen en común la utilización de la tecnología de iluminación **LED**. Las razones para elegirla se describieron en la sección 1.2 del capítulo de introducción, siendo una de las más importantes para los fines de este trabajo su capacidad de oscilación a alta frecuencia, algo imposible con otras tecnologías ya en desuso, como la de filamento incandescente.

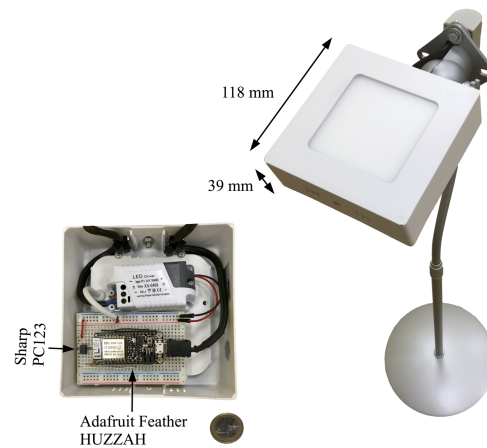
3.5.1. Prototipos

En el ámbito del proyecto Limus [107], dentro del grupo Grupo de Investigación en Sistemas Sensoriales (GISS) [108] de la Universidad de Extremadura (UEX), se llevó a cabo el desarrollo de un sistema de detección de marcas basado en luz visible que da origen a este trabajo de tesis [105]. Uno de los componentes de este sistema puede verse en la figura 3.6a: era el primer prototipo de la baliza.

Esta primera versión se compone de una placa de prototipado con cuatro interruptores que permiten que el usuario seleccione el código que el sistema va a emitir (valores del 0 al 3), una Raspberry Pi [109] encargada del control del sistema, y un **LED** de 5 mm de diámetro, con una intensidad luminosa entre 150 y 200 mcd, que se hace oscilar a una frecuencia de 1 kHz. Esta frecuencia se escogió tras experimentar con diferentes valores, para conseguir que no fuese perceptible por el ojo humano, pero suficiente para que el receptor fuese capaz de decodificar el mensaje.



(a) IPIN 2018.



(b) IPIN 2019.

Figura 3.6. Prototipos.

De nuevo dentro del ámbito del proyecto Limus [110], y presentado en la edición de 2019 del IPIN, que se celebró en Pisa, Italia, se publicó un artículo cuyo objetivo era comparar las tres tecnologías de posicionamiento estudiadas para dicho proyecto: ultrasónica, Bluetooth de baja energía (*Bluetooth Low Energy*, BLE) y luz visible. La baliza basada en luz visible de este segundo prototipo puede verse en la figura 3.6b.

En dicha figura se muestra el prototipo tal y como se montó para su presentación en público. La parte posterior de la baliza muestra el circuito de control e interrupción que modula la luz emitida por la baliza para la transmisión de mensajes. Está compuesto por una placa de desarrollo Feather Huzzah [111] de Adafruit con capacidad Wi-Fi, lo que permite cambiar el código emitido y realizar otras operaciones de forma remota. La combinación del efecto *rolling shutter* de las cámaras con tecnología CMOS, la capacidad de modular la intensidad de la luz emitida por las luminarias LED a alta frecuencia, y la incapacidad del ojo humano para detectar este tipo de oscilaciones es la base de funcionamiento del sistema [105]. De este modo resulta posible emitir un mensaje a través del medio iluminado sin que resulte molesto a las personas que lo atraviesen.

La motivación tras la creación del primer prototipo era muy simple: disponer de una prueba de concepto, comprobar que la transmisión de datos utilizando luz visible era factible. En este primer prototipo se utilizó la cámara frontal de un iPhone 5s [112]. La razón que llevó a la elección de este dispositivo fue meramente práctica: se disponía de uno y resultaba más rápido desarrollar una aplicación para su sistema operativo. Para el segundo prototipo la sección ?? se planteó una meta

más ambiciosa: desarrollar una baliza capaz de integrarse con el entorno gracias a la utilización de elementos que pueden adquirirse en tiendas minoristas. Con más tiempo y datos disponibles para tomar decisiones, se escogió Android para el receptor en lugar de iOS por dos razones:

- se trata de la plataforma móvil con más usuarios, de manera que el trabajo aquí presentado puede tener más impacto.
- Android proporciona herramientas más versátiles para controlar la cámara del dispositivo portable, lo que supone una ventaja como desarrollador dados los objetivos de este trabajo.

3.5.2. Diseño final

En la figura 3.7a puede verse el diagrama de funcionamiento de la baliza. El diseño final debe ser fácilmente replicable, configurable y versátil. Para lograrlo, su diseño se ha dividido en tres partes: la carcasa, la luminaria y el sistema de control.

Carcasa

El primer prototipo tenía todos sus componentes a la vista. El segundo estaba montado en el interior de la caja de una luminaria comercial. Para el tercero se ha querido desarrollar una carcasa que se pudiese fabricar y alterar, de manera que fuese sencillo crear nuevas versiones del mismo para luminarias de diferentes tamaños, así como para albergar diferentes componentes en su interior. La carcasa se ha creado utilizando una cortadora láser Perez Camps PC10/80 [113] proporcionada por Smart Open Lab (SOL), perteneciente a la red de Fab Labs [114] de Extremadura, que se encuentra en la Escuela Politécnica de Cáceres. Para su diseño se ha utilizado el modelo “Electronics Box” de Boxes.py [115].

Luminaria

La luminaria utilizada para las pruebas de este capítulo se adquirió en una superficie comercial. Su precio no superó los 20 €. Se trata de un panel de 18 W, de luz fría (6500 K) y 1350 lm. La parte exterior del panel, la que se presenta al usuario, tiene 225 mm de lado. La parte lista para empotrarse tiene 205 mm de lado y utiliza unas pinzas para mantenerse sujeta.

Sistema de control

En la figura 3.7b se puede ver el interior de la mayor de las balizas, así como los componentes utilizados para conformar el sistema de control.

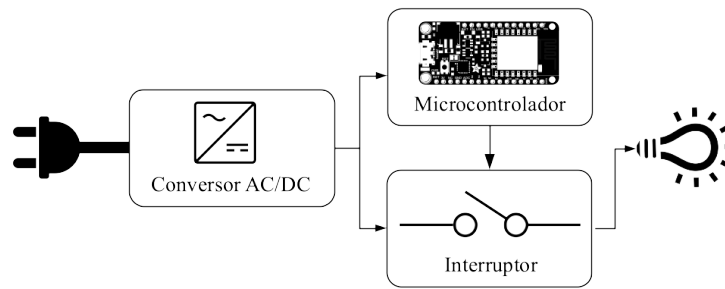
El módulo transmisor necesita dos convertidores AC/DC, el primero para el panel LED y el segundo para la placa de desarrollo Amica de NodeMCU [116]. Dicha placa se basa en el módulo ESP8266 [117], que se utiliza para interrumpir la fuente de alimentación con la ayuda del relé de estado sólido Crydom CMX200D3 [118]. La parte software del sistema de control se describe en la sección 4.3. En la sección 4.4 se describe la herramienta complementaria al generador de códigos Manchester, capaz de decodificar el código emitido por la baliza que acabamos de ver.

PCB

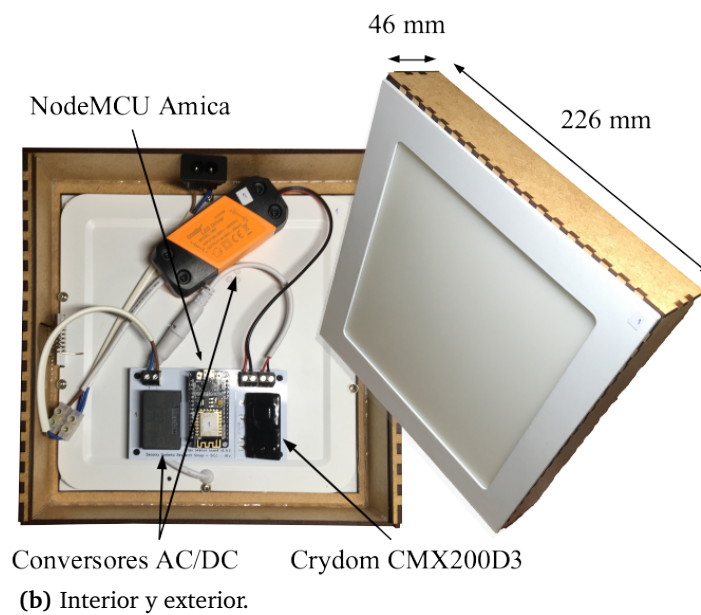
El diseño final incluye una placa placa de circuito impreso (*Printed Circuit Board*, PCB) que facilita el montaje de los componentes dentro de la carcasa. Para crearla se utilizó KiCAD [119], una aplicación multiplataforma y de código abierto para el diseño de componentes electrónicos. En la figura 3.7c se puede ver esta PCB, sin componentes. Las perforaciones de las esquinas de la PCB se utilizan para montarlas en la parte trasera de los paneles LED utilizando separadores de nylon, aislantes, para evitar así cualquier conexión eléctrica inesperada. De esta manera se ahorra espacio y no es necesario utilizar parte de la carcasa para el montaje. Como muestra la figura 3.7b, el interior de la baliza está mucho más organizado gracias a la utilización de la PCB para integrar los diferentes componentes. Como consecuencia de esta mejor organización de los componentes en el interior de la baliza ha sido posible reducir el perfil de la misma, permitiendo disponer de una prototipo más manejable.

3.6. Pruebas realizadas

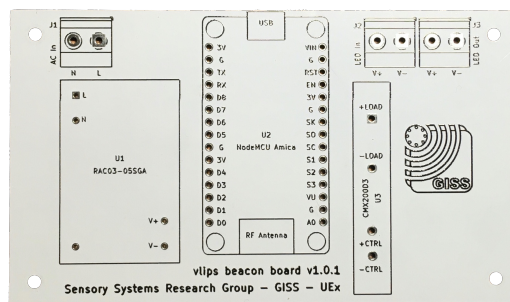
Para caracterizar el funcionamiento del sistema de comunicación de datos desarrollado se han llevado a cabo pruebas de alcance, cobertura e interferencia.



(a) Diagrama de funcionamiento del emisor.



(b) Interior y exterior.

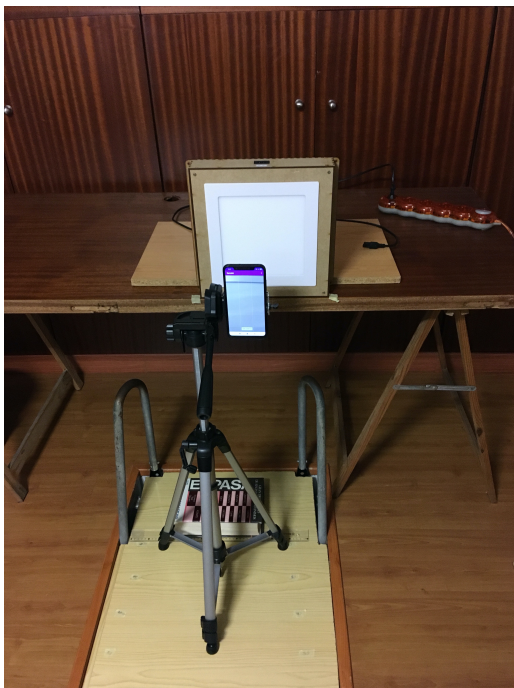


(c) placa de circuito impreso (*Printed Circuit Board, PCB*) diseñada para el montaje de la baliza.

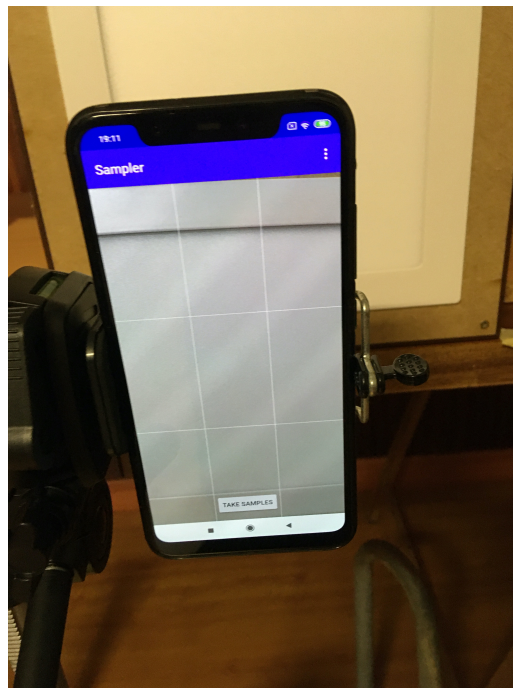
Figura 3.7. Prototipo 3.

3.6.1. Alcance

La primera de las pruebas realizadas para evaluar el funcionamiento del sistema permite determinar el alcance del mismo y así conocer la distancia en línea recta entre emisor y receptor a partir de la cual deja de ser posible la comunicación de datos. Para ello se ha situado la baliza emisora a una altura tal que su centro está a 0,895 m del suelo. De igual forma, el dispositivo portable se ha montado en un trípode, de manera que el centro de la cámara trasera está también a 0,895 m. De esta forma tanto emisor como receptor están a la misma altura. El suelo se ha marcado a intervalos regulares de 10 cm hasta llegar a 2,5 m. En la figura 3.8a se puede ver cómo están distribuidos todos los elementos del experimento.



(a) Distribución.



(b) Distancia mínima.

Figura 3.8. Diseño de las pruebas de alcance.

Debido a las características físicas de la baliza y la cámara, la distancia de comienzo de las pruebas debe ser de 30 cm. En la figura 3.8b se puede apreciar que a 20 cm la cámara no es capaz de capturar la totalidad de la superficie de la luminaria.

La baliza estará emitiendo el código 10 que, incluyendo la cabecera, tiene esta forma:

0111010110011

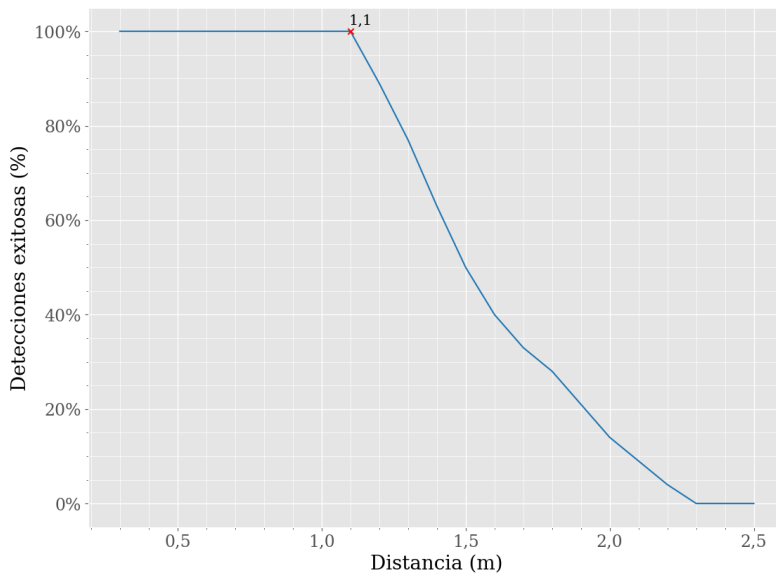
Este código no ha sido elegido al azar. Se trata de uno de los dos códigos menos favorables (el otro es el 5) para el proceso de detección porque contiene las cuatro combinaciones de dos bits posibles al trabajar con la codificación Manchester: 00, 01, 10 y 11. Si alguna combinación resulta desfavorable en el proceso de decodificación, afectará a este proceso de pruebas. En la tabla 3.1 se pueden ver los 16 valores posibles utilizando cuatro bits, con 01110 como cabecera. Marcados en negrita, los códigos con todas las combinaciones de parejas. La tercera columna de la tabla indica cuántas parejas diferentes hay en cada secuencia de bits.

Tabla 3.1. Parejas de bits diferentes para cuatro bits.

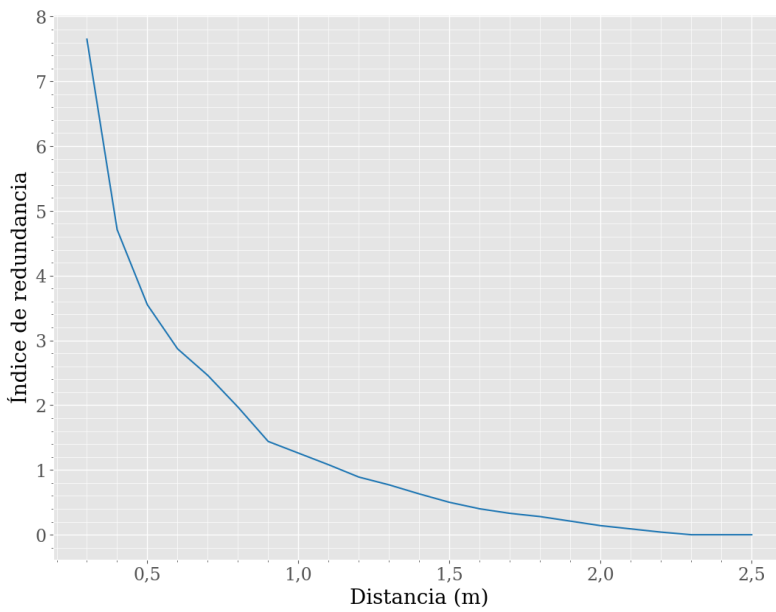
Código	Codificación Manchester	Parejas distintas
0	01110 11 00 11 00	2
1	01110 11 00 11 01	3
2	01110 11 00 10 11	3
3	01110 11 00 10 10	3
4	01110 11 01 00 11	3
5	01110 11 01 00 10	4
6	01110 11 01 01 00	3
7	01110 11 01 01 01	2
8	01110 10 11 00 11	3
9	01110 10 11 00 10	3
10	01110 10 11 01 00	4
11	01110 10 11 01 01	3
12	01110 10 10 11 00	3
13	01110 10 10 11 01	3
14	01110 10 10 10 11	2
15	01110 10 10 10 10	2

Las muestras obtenidas serán procesadas por el decodificador que, como ya se ha comentado, analizará los resultados obtenidos y generará informes de manera automática. Entre otros resultados, se dispondrá de gráficos como los que pueden verse en la figura 3.9.

Como se puede apreciar, el porcentaje de detección se mantiene al 100% hasta llegar a 1,1 m, distancia a la que comienza a descender paulatinamente hasta llegar a los 2,3 m; a esa distancia las detecciones dejan de ser posibles. El índice de redundancia indica cuántas veces se ha detectado el código correcto en una única muestra. Por ejemplo, en el código de barras de la figura 3.10 se puede apreciar



(a) Detecciones exitosas (%) en función de la distancia entre emisor y receptor.



(b) Índice de redundancia.

Figura 3.9. Resultados de las pruebas de alcance.

que el mismo patrón completo se repite en varias ocasiones, concretamente 7 veces (aunque aparece dos veces incompleto en los límites izquierdo y derecho de la superficie luminosa de la baliza). En ese caso, el índice de redundancia sería 7. Así, en la figura 3.9b se puede ver que el índice de redundancia desciende desde aproximadamente 7,5 hasta aproximadamente 1 cuando se pasa de 0,30 m a 1,20 m. La detección se realiza de forma correcta en cada ocasión, pero la fiabilidad de las mismas va decreciendo.



Figura 3.10. Muestra con código repetido.

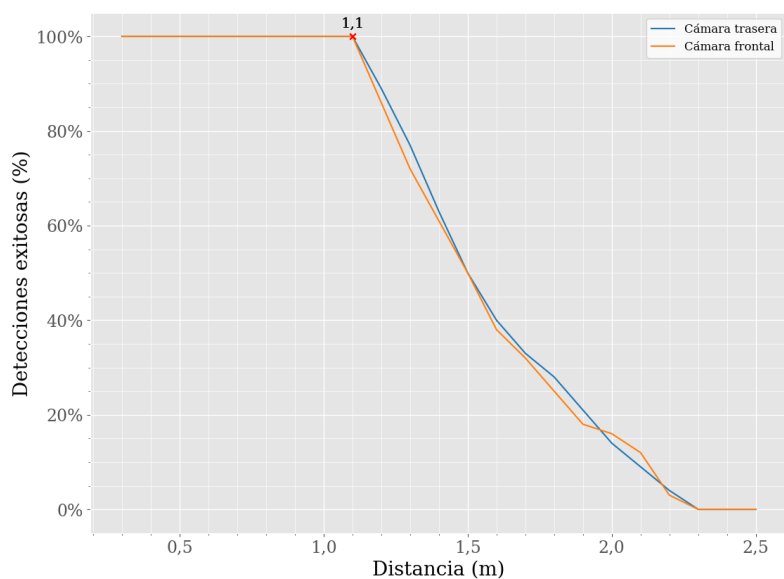
Comparación de cámaras

Las primeras pruebas experimentales del alcance del sistema de comunicación de datos se han realizado utilizando la cámara trasera del dispositivo portable. El smartphone Android utilizado dispone, como la práctica totalidad de los disponibles en el mercado en la actualidad, de una cámara frontal y una cámara trasera. Es necesario conocer si el rendimiento del sistema es similar utilizando la cámara trasera. Para resolver esta incógnita se ha repetido la misma prueba utilizando la cámara frontal en lugar de la trasera. Los resultados obtenidos con ambas cámaras se pueden comparar en la figura 3.11.

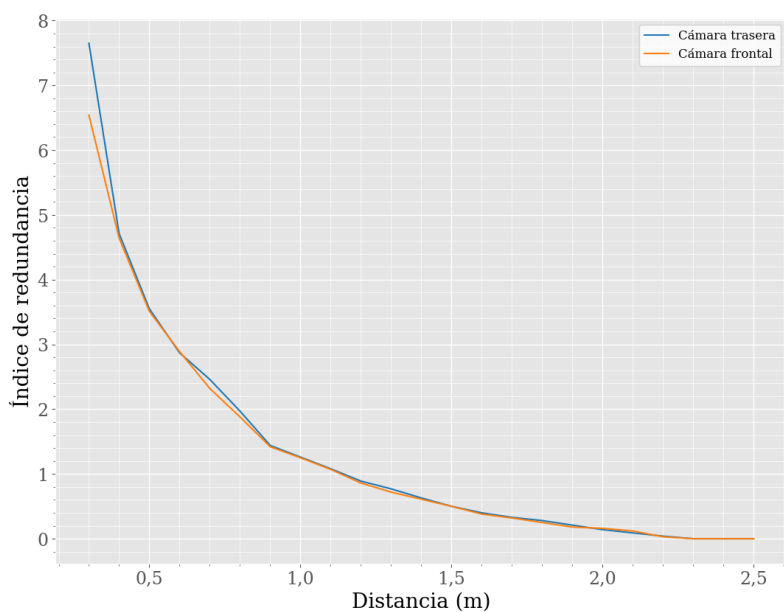
Como se puede comprobar, los resultados obtenidos son prácticamente idénticos. Con esta información, dado que el rendimiento del sistema no se ve influido por la cámara del smartphone utilizado, es posible decantarse por cualquiera de las dos cámaras para realizar el resto de las pruebas. Se ha elegido la cámara trasera porque la pantalla queda en el lado opuesto de la baliza, como puede verse en la figura 3.8. Así, ajustar la posición del receptor para que esté perfectamente alineado con el emisor resulta mucho más sencillo.

3.6.2. Cobertura

El objetivo de esta prueba es determinar la zona de cobertura dentro de la cual emisor y receptor son capaces de operar. En la figura 3.12 se pueden ver los resultados de realizar la prueba descrita en la sección 3.6.1, pero desplazando la luminaria



(a) Detecciones.



(b) Índice de redundancia.

Figura 3.11. Comparación de los resultados de las pruebas de alcance utilizando la cámara trasera y la frontal.

transversalmente con respecto al receptor en pasos de 0,10 m a la izquierda, desde $-1,20$ m hasta $1,20$ m. En todo momento la alineación relativa de ambos elementos se mantiene perpendicular. De esta manera, el efecto del campo de visión de la cámara del receptor también tiene influencia en el resultado de la prueba. Los resultados obtenidos con el desplazamiento hacia la izquierda de la luminaria se han reflejado sobre el eje Y.

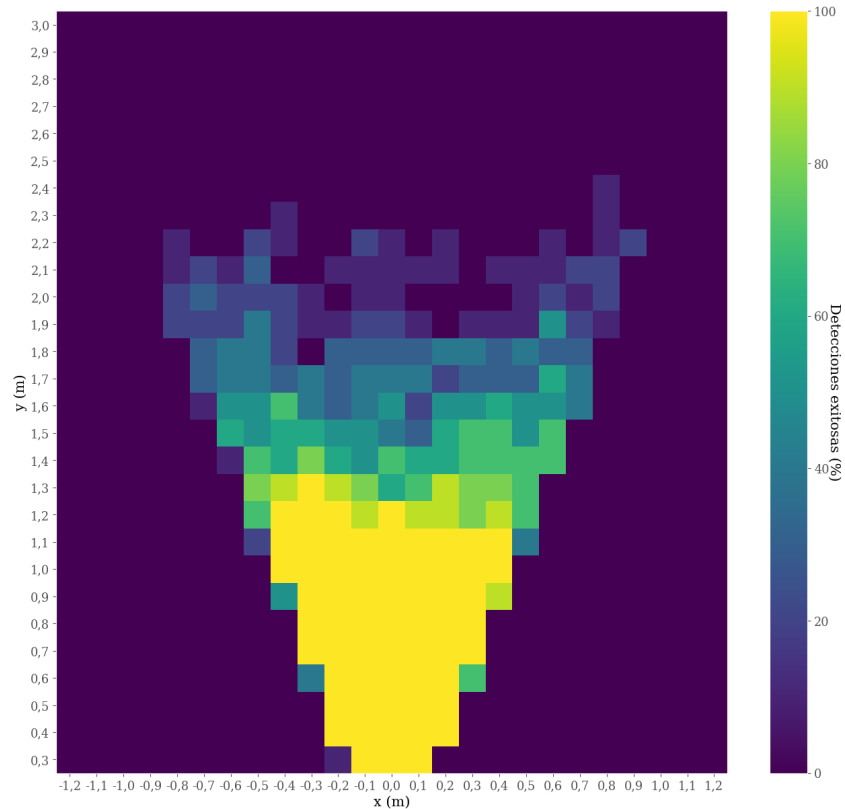


Figura 3.12. Prueba de alcance sobre una superficie.

Como se puede comprobar, la zona de cobertura óptima está formada por el triángulo que parte de la ubicación de la luminaria, con una profundidad de unos $1,1$ m y una anchura de unos $0,8$ m (desde $-0,4$ m hasta $0,4$ m). La capacidad de detección se va perdiendo paulatinamente a medida que emisor y receptor se separan.

Resulta interesante comprobar que estos datos experimentales se corresponden con los datos teóricos obtenidos a partir de las características físicas de la cámara trasera del Xiaomi Mi 8 [120] utilizada para tomar las muestras. La intención es calcular el FOV de la cámara, superponiéndolo entonces dentro de la prueba de alcance presentada en la figura 3.12 para comprobar que ambos valores concuerdan.

De acuerdo con lo establecido en [121] el FOV de cámaras y lentes puede determinarse como sigue:

$$FOV = 2 \cdot \arctan\left(\frac{S}{2 \cdot f}\right) \quad (3.1)$$

donde S es una matriz de 2×1 cuyos componentes son el ancho y el alto del sensor de la cámara y f es la distancia focal, ambos en mm.

Sólo nos interesa el ancho del campo de visión, puesto que estamos moviendo el smartphone en el plano horizontal. Las variables, entonces, el ancho del sensor y la distancia focal de la cámara. Gracias a las especificaciones del Mi 8 [122] se sabe que la distancia focal de la cámara trasera es de 4,22 mm. De ahí también podemos conocer el modelo del sensor utilizado. Se trata del IMX363 Exmor RS desarrollado por Sony [123]. Sus especificaciones indican que las dimensiones del sensor son 4032 px \times 3024 px. Además, el tamaño del píxel es de 1,40 μ m. El móvil se utiliza en posición vertical (o retrato) al tomar las muestras, con lo que el sensor está rotado 90°. Por lo tanto, se debe utilizar 3024 px como medida del ancho. Es necesario conocerlo en milímetros, valor que se obtiene en la ecuación (3.2):

$$\text{ancho sensor} = 3024 \text{ px} \cdot 1,4 \mu\text{m/px} = 4233,6 \mu\text{m} = 4,2336 \text{ mm} \quad (3.2)$$

Además de en las especificaciones del sensor, la distancia focal también aparece en los datos de formato de archivo de imagen intercambiable (*Exchangeable image file format*, Exif) [124] que se incluyen en cada fotografía tomada. En concreto, la distancia focal de la cámara trasera del Xiaomi Mi 8 es de 4,216 mm.

Con estos valores resulta posible calcular el FOV utilizando la ecuación (5.37), que resulta ser de 53,32°. Es posible utilizar ese valor para superponer un triángulo que parta de la distancia máxima entre receptor y emisor en la que aún se tiene una recepción del 100 %, es decir, 1,2 m. En ese punto se sitúa el vértice de un triángulo. El ángulo en dicho vértice será el del FOV. El resultado puede verse en la figura 3.13a.

Situando la cámara a 1,2 m como en la figura 3.13a, perpendicular a la baliza, se puede comprobar que la cobertura máxima coincide con dicho punto. Sin embargo, si se desplaza a la derecha, justo en el primer punto en el que la recepción deja de ser del 100 % (aproximadamente a 0,5 m, tal y como puede verse en la figura 3.13b), se puede comprobar que se está trabajando en el límite del campo de visión.

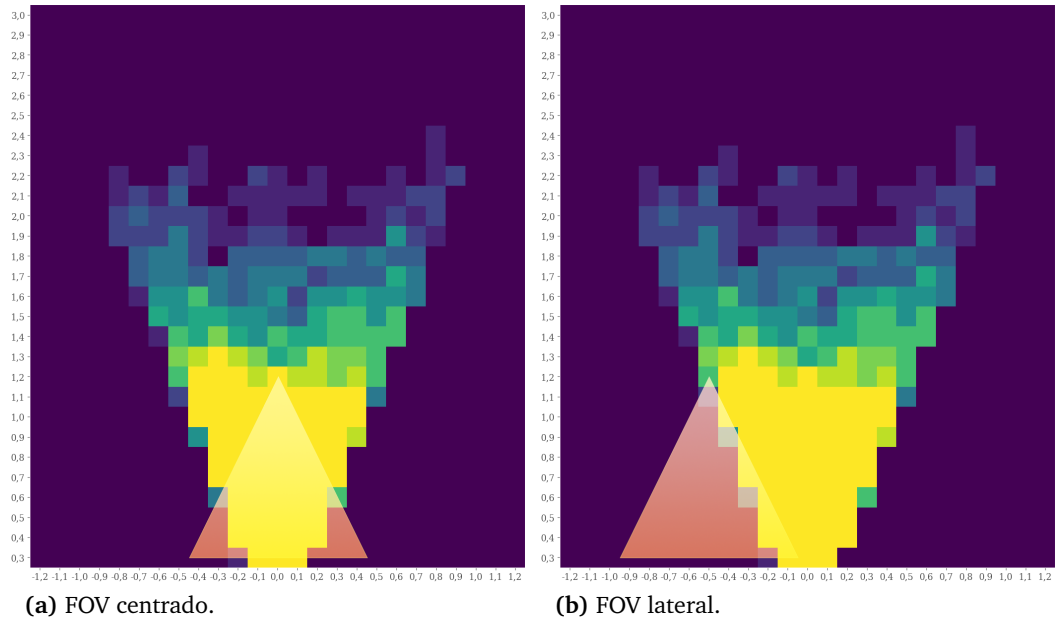


Figura 3.13. Correspondencia entre cobertura y FOV.

3.6.3. Interferencia

Para poder caracterizar el funcionamiento de este sistema resulta interesante conocer cómo se comporta el proceso de decodificación cuando más de una baliza está en el campo de visión del receptor. Para ello se ha preparado un experimento del que forman parte dos balizas.

La baliza principal emite el código 10 que, como se comentó en la sección 3.6.1, tiene la peculiaridad de contener todas las posibles parejas de dos bits. La baliza secundaria emite el otro código que contiene todas las parejas posibles, el código 5. El receptor se ha situado a 1,2 m de la baliza principal. El centro de la cámara trasera del mismo está a 0,895 m de altura, al igual que el centro de la baliza principal. Ambos están perfectamente alineados.

La baliza secundaria se ha situado a la izquierda de la principal, alineadas. Inicialmente, la separación entre el centro de ambas balizas es de 0,7 m, lo que deja a la baliza secundaria fuera del campo de visión de la cámara del receptor. Esta situación puede verse en la figura 3.14.

En cada fase del experimento se toman 100 muestras. Al terminar, la baliza secundaria se acerca a la principal en pasos de 5 cm. El experimento concluye cuando ambas balizas están a 30 cm de distancia, momento en el que ya no se pueden aproximar más entre sí debido a las características físicas de las balizas. En ese momento ambas



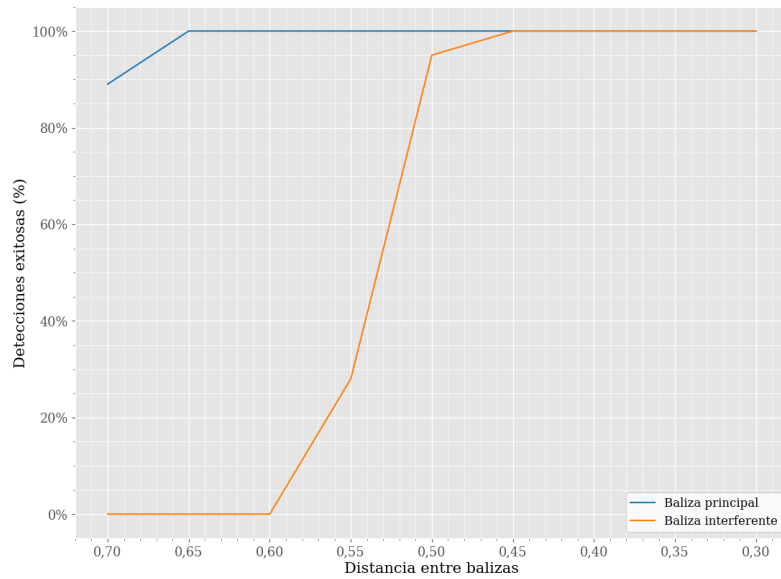
Figura 3.14. Prueba de interferencia.

están dentro del campo de visión de la cámara del receptor. Los resultados del experimento pueden verse en la figura 3.15.

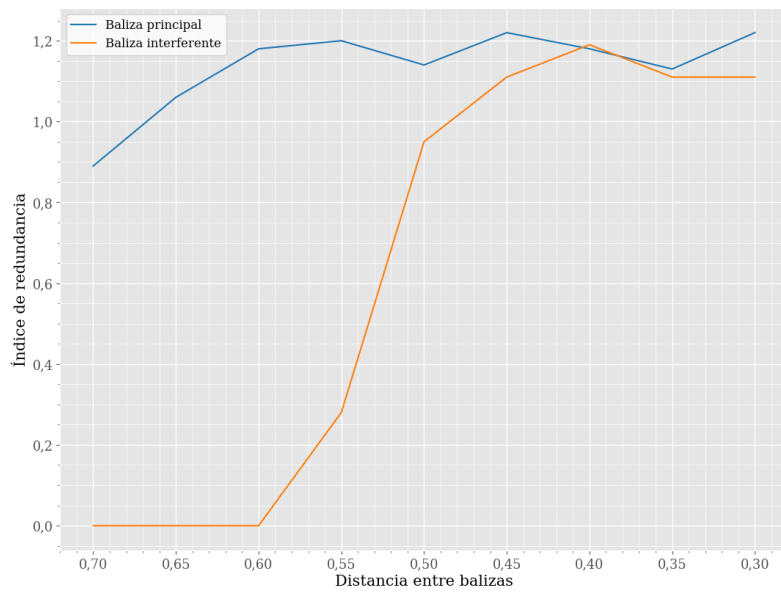
En la figura 3.15a se compara el porcentaje de detecciones de ambas balizas a medida que la baliza secundaria se aproxima a la principal. La baliza principal siempre es detectada un 100 % de las veces. La secundaria también a partir de los 0,45 m de distancia entre ambas. Esto permanece así hasta el final del experimento.

El porcentaje de detección de la baliza secundaria aumenta a medida que ambas balizas se acercan. Esto se debe a que, a medida que acercamos la secundaria a la principal, más superficie de la baliza secundaria está dentro del campo de visión de la cámara del receptor y, por tanto, más sencillo resulta detectarla.

En la figura 3.15b se compara el índice de redundancia de la detección de cada baliza a medida que la baliza secundaria se aproxima a la principal. La baliza principal tiene un índice de redundancia mayor que la secundaria mientras la distancia entre ambas es superior a 0,4 m. Cuando dicha distancia disminuye, el índice de redundancia de la detección de ambas se mantiene aproximadamente igual, aunque la principal se mantiene por delante de la secundaria.



(a) Aciertos y fallos en la detección de la baliza de referencia.



(b) Índices de redundancia en la detección de la baliza de referencia.

Figura 3.15. Resultados de las pruebas de interferencia.

3.7. Optimización

En esta sección se abordan una serie de mejoras que permiten aumentar el alcance del sistema y optimizar la utilización de recursos. Además, se complementa con un estudio del consumo de batería del smartphone en el proceso de decodificación.

3.7.1. Elección del formato de imágenes

El formato JPEG [125], utilizado para guardar las muestras utilizadas en las pruebas de la sección 3.6, contiene datos sobre los tres canales RGB. Como se comenta en la sección 3.4, entre los primeros pasos del proceso de decodificación está convertir la imagen a escala de grises, puesto que sólo los datos de intensidad luminosa son de interés, no los cromáticos. Se puede ahorrar un valioso tiempo de procesamiento guardando las muestras en formato YUV [126], que permite acceder directamente a los datos de intensidad luminosa sin necesidad de realizar ninguna operación sobre los datos originales.

En una prueba realizada en el dispositivo móvil utilizando el sistema descrito en la sección anterior se comprobó que el tiempo de procesamiento de cada muestra pasaba de 354 ms a 146 ms de media, es decir, casi un 60 % menos. Por lo tanto, queda claro que el camino a seguir pasa por trabajar con formato YUV en lugar de JPEG, siempre y cuando no sea necesario compartir las imágenes. Para compartirlas o realizar procesamiento offline se seguirá utilizando el formato JPEG, puesto que resulta más conveniente al ser más simple visualizar su contenido en la mayoría de sistemas operativos. Además, en esa fase el tiempo de procesamiento no es crítico.

3.7.2. Rotación de bits

La tasa de detección de este sistema de comunicación baja del 100 % al superar 1,1 m de distancia entre emisor y receptor. Esto está relacionado con varios factores que definen del sistema, como el ancho de la luminaria, la frecuencia de emisión del código, el número de bits del mensaje y la distancia entre la baliza y el receptor. A cierta distancia, en la muestra tomada el código ocupa prácticamente el ancho de la luminaria (pueden verse ejemplos de ello en la sección 3.7.3). Si a esta limitación añadimos el hecho de que el mensaje emitido no tiene por qué comenzar justo en el primer punto de la muestra, sino que puede estar a la mitad de la misma, la posibilidad de decodificar correctamente el mensaje decrece.

En la figura 3.16 se puede ver un ejemplo del resultado de procesar una de las muestras tomadas a 1,2 m de distancia.

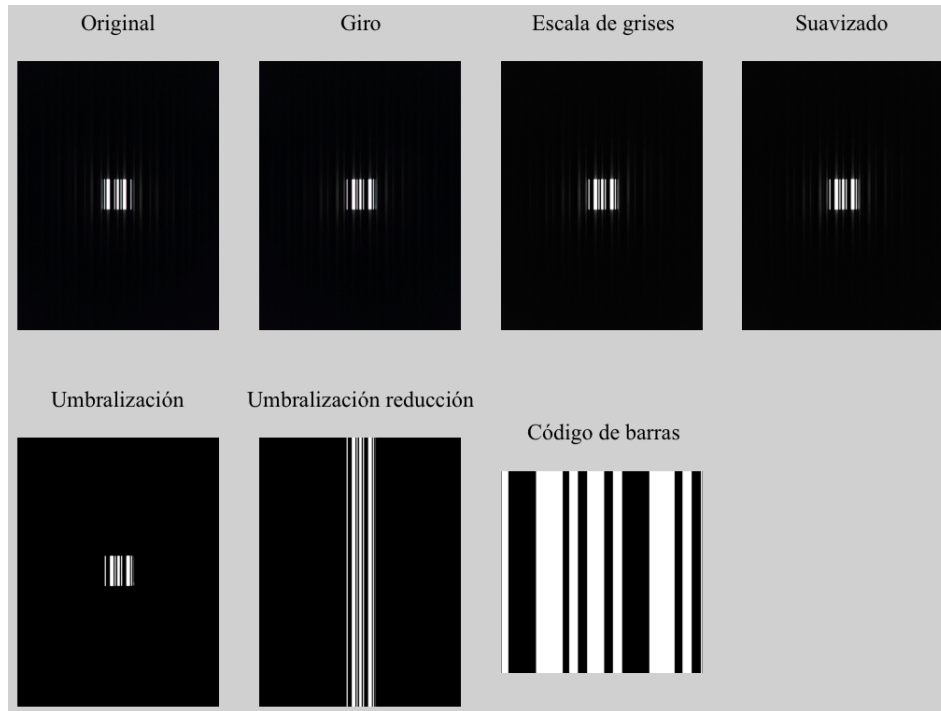


Figura 3.16. Resultado del procesamiento de una muestra tomada a 1,2 m.

El código emitido en esa muestra es el 10. La representación en Manchester de dicho valor es 0111010110100. Resulta sencillo identificar dicho código en la figura 3.16, basta con localizar la banda blanca más ancha en el código de barras.

El proceso de decodificación logra identificar el código 1 con un índice de redundancia 1, es decir, sólo ha sido capaz de encontrar el código correspondiente 1 vez. Sin embargo, surge la pregunta de si sería posible reconstruir otra copia del código a partir de lo que hay tras el primer código encontrado y lo que queda antes de la primera cabecera. Quizá sea más sencillo de explicar examinando la cadena de bits en la que se convierte el código de barras:

```
01000111010110100011101010
```

El proceso de decodificación busca la primera ocurrencia de la cabecera 01110, separando la secuencia de bits en varias partes:

```
0100 01110 10110100011101010
```

Lo que hay antes de la cabecera se descarta, los ocho bits que hay a continuación de la misma serán decodificados en busca del mensaje enviado:

0100 01110 10110100 011101010

De esa manera, 10110100 se convierte en el código 10. El proceso se repite con lo que queda de la secuencia de bits:

01110 1010

Por desgracia, los bits que hay a continuación de la segunda cabecera encontrada no son suficientes para buscar un mensaje. Afortunadamente, el código se emite una y otra vez, de forma periódica. Antes de la primera cabecera hay cuatro bits. Esos cuatro bits podrían ser el final del mensaje que no termina tras la segunda cabecera, y sumados a los cuatro bits que hay tras la segunda cabecera podrían servir para decodificar una segunda vez el mensaje:

01110 1010 0100

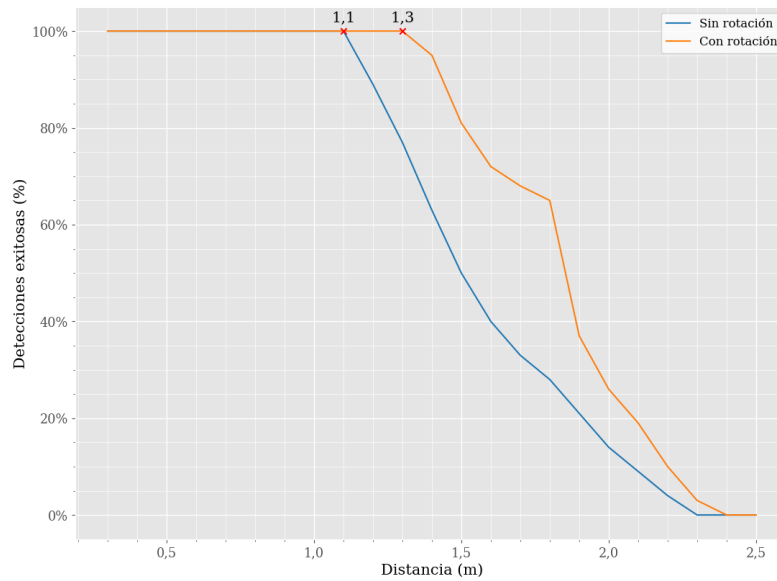
La secuencia de bits 10100100 no se corresponde con ningún código Manchester de cuatro bits, de manera que no se obtiene ninguna mejora en este caso. Sin embargo, el ejemplo visto hasta ahora es uno de los muchos que pueden darse. Existe la posibilidad de que en ciertas circunstancias los pasos descritos sí supongan una mejora en el proceso de decodificación.

Para comprobarlo se ha realizado una modificación en el proceso de decodificación que utiliza esta rotación de la secuencia de bits, lo que va a permitir comparar el rendimiento de ambas versiones del algoritmo. La primera comparación se hará con la prueba de alcance. En la figura 3.17a se puede ver el resultado de comparar el éxito en las detecciones sin y con la rotación de los bits.

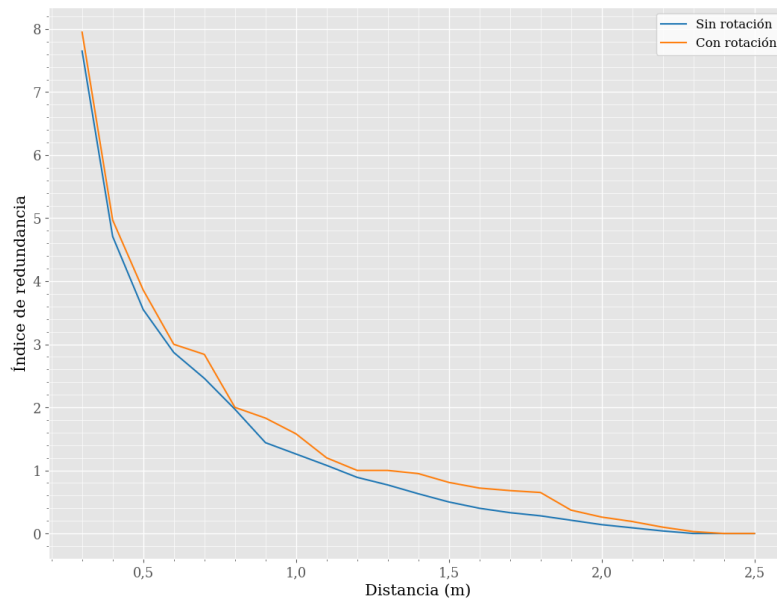
Como se puede apreciar, sin rotación de bits el 100 % de detecciones se mantiene hasta llegar a 1,1 m. Sin embargo, con rotación de bits este valor se mantiene hasta llegar a 1,3 m. En este caso en particular, la rotación de bits supone una mejora de 0,2 m en el alcance del sistema de comunicación de datos.

Otro de los parámetros que permite comparar el rendimiento de esta mejora es el índice de redundancia de la detección, es decir, el número de veces que el código se ha detectado correctamente a partir de una de las muestras. Esta comparación puede observarse en la figura 3.17b, donde se aprecia una ligera mejoría al utilizar la rotación de bits.

Como queda de manifiesto, sin rotación de bits el 95 % de las detecciones ocurren hasta 1,7 m. Sin embargo, con rotación este porcentaje de detecciones se mantiene hasta 1,8 m. En este caso en particular, la rotación de bits supone una mejora de 0,1 m en la curva de distribución acumulativa del sistema de comunicación de datos.



(a) Detecciones exitosas (%) en función de la distancia entre emisor y receptor.



(b) Índice de redundancia.

Figura 3.17. Resultados de las pruebas de alcance, sin y con rotación de bits.

Antes de incorporar esta mejora al sistema de comunicación de datos resulta fundamental comprobar que, efectivamente, el porcentaje de detecciones mejora con independencia del código emitido. Para ello se han tomado 100 muestras por cada uno de los 16 códigos posibles a una distancia en línea recta de de 1,5 m, con emisor y receptor enfrentados y paralelos. Esta comparación se presenta en la figura 3.18.

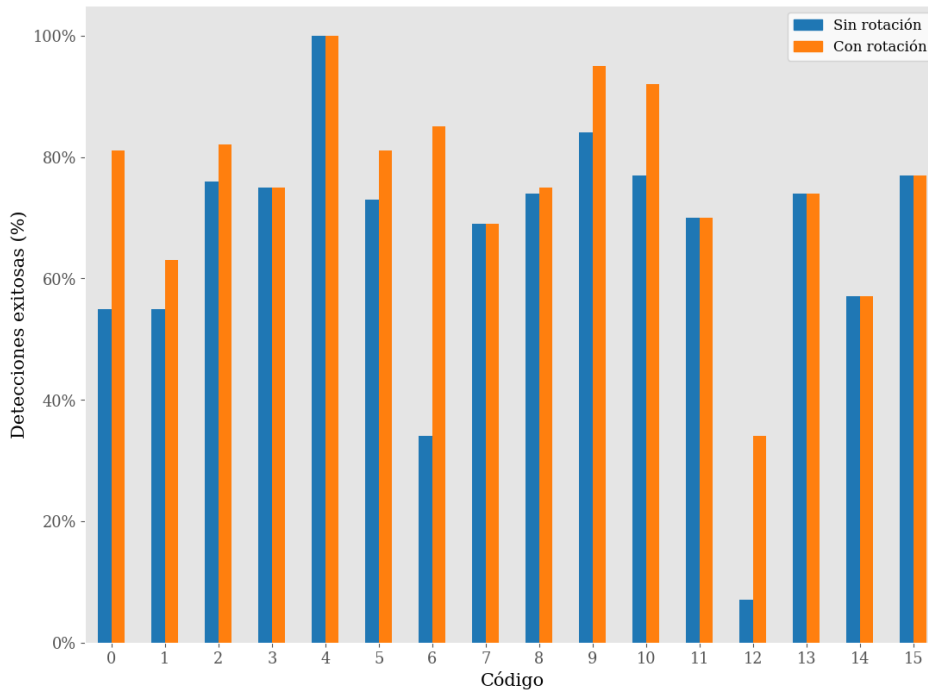


Figura 3.18. Éxito en la detección de códigos a 1,5 m, sin y con la rotación de bits activada.

Los resultados obtenidos sin la rotación de bits son peores que los obtenidos con la rotación activa. En concreto, el porcentaje de aciertos con la rotación desactivada es del 66 %, mientras que con la rotación activada llega al 75 %, un incremento del 9 %, una mejora muy significativa si tenemos en cuenta que estamos trabajando en el límite del alcance del sistema.

Los resultados de comparar la cobertura sin y con la rotación de bits también son interesantes, como se puede apreciar en la figura 3.19.

En este caso, la zona de cobertura óptima está formada por el triángulo que parte de la ubicación de la luminaria, con una profundidad de unos 1,3 m (frente a los 1,1 m sin rotación) y un fondo de 1 m (frente a los 0,8 m anteriores). Este fondo va desde los -0,5 m hasta los 0,5 m (sin rotación iba desde -0,4 m hasta 0,4 m).

Con estos datos, la superficie de cobertura ha pasado de 0,44 m² a 0,65 m², lo que supone un incremento de aproximadamente el 32 % en la zona de cobertura óptima.

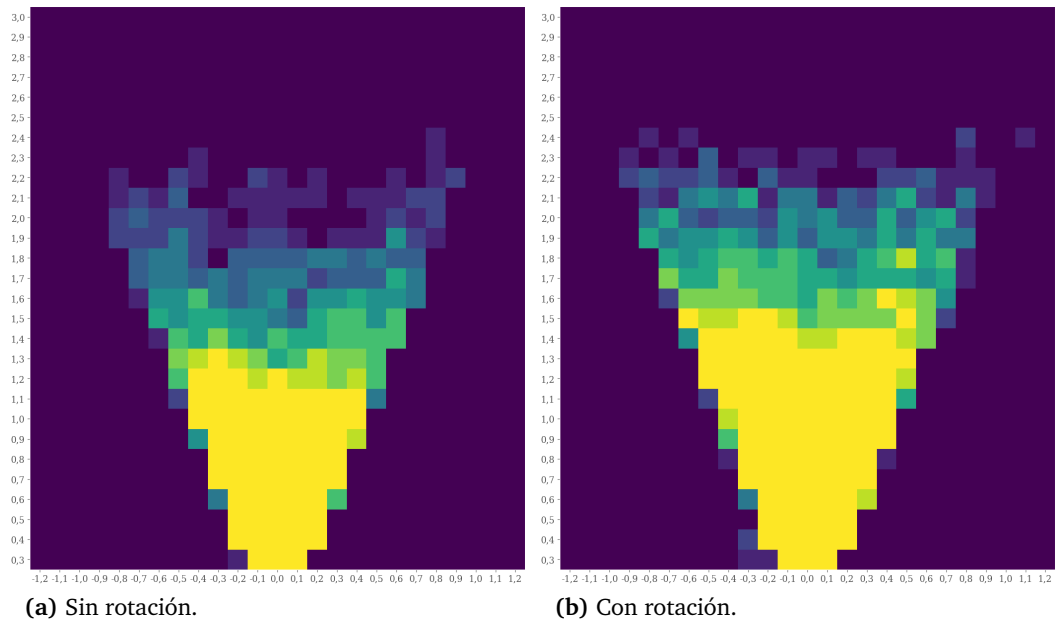


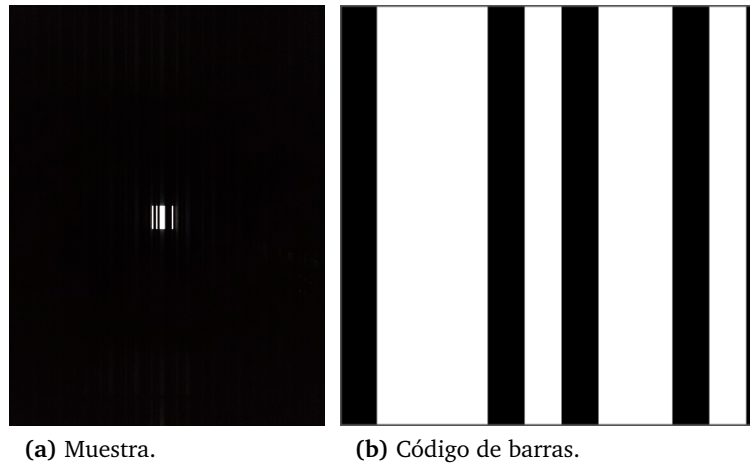
Figura 3.19. Cobertura sin y con rotación de bits.

En vista de los resultados obtenidos, se decide incorporar esta mejora al algoritmo de decodificación.

3.7.3. Toma en ráfagas

En la figura 3.17a se puede ver que el sistema de comunicación con rotación de bits funciona con un 100 % de aciertos hasta llegar a 1,3 m, momento en el que este porcentaje comienza a descender hasta llegar al 0 % a una distancia de 2,4 m.

Una de las razones por las que el sistema actual tiene esta limitación de alcance es la relación entre el número de bits que se quieren enviar y el ancho del panel LED que impide que, a cierta distancia, todos los bits quepan en él. Por ejemplo, en la figura 3.20 se puede ver la muestra tomada a 2,4 m y el código de barras correspondiente. La representación en Manchester del valor decimal 10 (binario 1010) se muestra en la figura 3.20b: 0111010110100. La primera barra negra es el 0, la siguiente barra blanca el 111 y la siguiente barra negra es el 0. Esos 5 valores conforman la cabecera. Los valores que aparecen a continuación son 10 (1), 11 (0) y 01 (1). De la última pareja, 00 (0), sólo aparece un fragmento del primer 0. Por lo tanto, no resulta posible encajar el código completo en el panel LED a esa distancia.

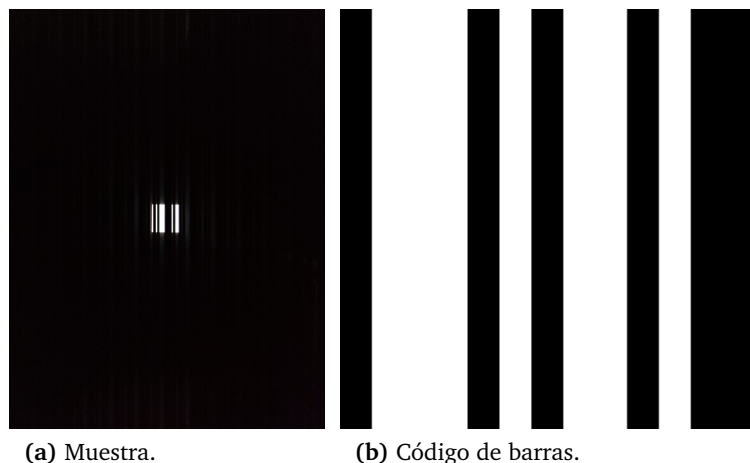


(a) Muestra.

(b) Código de barras.

Figura 3.20. Muestra y código de barras a 2,4 m.

En la figura 3.21 se puede ver la muestra tomada a 2,0 m y el código de barras correspondiente. El código emitido es el mismo que en la figura 3.20, con la diferencia de que esta vez sí caben todos los bits enviados. De hecho, hasta sobra un poco de espacio al final. Aunque el mensaje cabe dentro del panel LED tal y como lo percibe la cámara del smartphone, el receptor no siempre captura el mensaje con la cabecera justo al principio. De hecho, tanto en la figura 3.20a como en la figura 3.21a dicha cabecera aparece a mitad de la muestra.



(a) Muestra.

(b) Código de barras.

Figura 3.21. Muestra y código de barras a 2,0 m.

La forma óptima de conseguir que la cabecera esté siempre al principio de la muestra es sincronizar el emisor y el receptor. En las pruebas realizadas esto no ha sido posible. Ambos elementos pueden estar trabajando a frecuencias compatibles, pero eso no garantiza que la cabecera se encuentre al principio de la captura. Sin

embargo, existe una solución que se encuentra a medio camino entre lograr una sincronización perfecta y tomar una única muestra sin sincronización: el modo ráfaga (*burst mode* en inglés).

El concepto de las ráfagas de fotografías no es nuevo, aunque sí que se ha incorporado recientemente al mundo de la fotografía digital móvil. Tomar una ráfaga de fotografías no es más que aprovechar que todo el hardware está listo para tomar fotografías y, en lugar de tomar una sola, tomar una serie de ellas tan rápido como el bus de datos del dispositivo lo permita. Se trata de una operación crítica, ya que no sólo están involucrados los límites de velocidad de transferencia del sistema, también su velocidad de procesamiento y capacidad de almacenamiento.

En el caso del sistema de comunicación de datos de este trabajo, se parte de la hipótesis de que tomar una ráfaga a suficiente velocidad puede aumentar la probabilidad de que en una de las muestras la cabecera esté al principio de la misma. Para comprobar si esta hipótesis es cierta se ha modificado la aplicación, incluyendo un nuevo experimento que, en lugar de tomar una única muestra, toma 10. De esta forma, si en un determinado punto se repitiese este experimento 100 veces, se terminarían obteniendo 1000 muestras, 10 por cada iteración del experimento.

En esta primera fase las muestras no se analizan en el dispositivo sino que son trasladadas a un ordenador en el que serán procesadas. Para ello se utilizarán una serie de scripts que procesarán las muestras dadas las nuevas condiciones de funcionamiento. El lector puede encontrar más información sobre estas herramientas de análisis offline en la sección 4.4.

En primer lugar, se ha modificado el script **decode.py**, que toma una muestra e intenta encontrar el código emitido por la baliza. Este script es capaz de trabajar con una única muestra o con una carpeta en la que se encuentren varias de ellas. La modificación consiste en que, además de todos los parámetros que devuelve para cada muestra, se incluya uno más: la distancia que existe desde la cabecera al comienzo de la muestra. Esto permite determinar la calidad de la muestra utilizando este parámetro: cuanto menor sea esa distancia, mejor será la muestra.

En segundo lugar, se han modificado **decoder.py** y **sample.py** para que tengan en cuenta las ráfagas. Combinados, las procesan, ordenan los resultados por código encontrado, de manera que aquella que más veces se haya encontrado aparece la primera, y escogen aquella que más veces aparezca y cuya cabecera esté más cerca del comienzo de la muestra.

Para comprobar el funcionamiento de este nuevo algoritmo se ha utilizado el mismo sistema de toma de muestras descrito en la sección 3.6.1, adquiriendo 100 muestras

en cada punto. En este caso cada muestra consiste 10 fotografías tomadas en ráfaga, resultando en 1000 fotografías por punto, con un total de 23 000 fotografías disponibles para el posterior análisis. En la figura 3.22 se puede observar el alcance de este sistema comparado con el descrito en la sección 3.7.2.

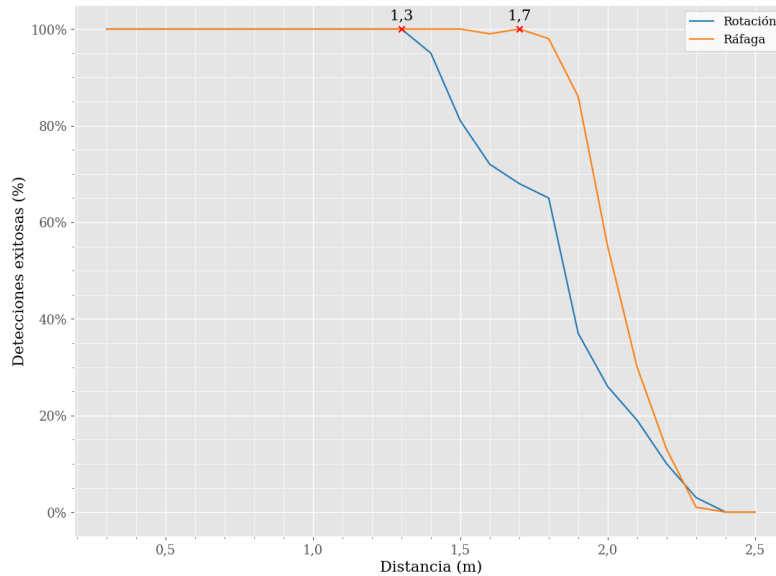


Figura 3.22. Comparación del sistema de ráfaga con el de rotación de bits.

El sistema de ráfaga permite extender el rango en el que el porcentaje de aciertos es del 100 % desde los 1,3 m del sistema con rotación de bits hasta 1,7 m, un aumento ligeramente por encima del 30 %.

Para determinar la cobertura conseguida con este algoritmo y poder compararla se ha realizado el mismo experimento descrito en la sección 3.6.2. El resultado puede verse en la figura 3.23b. Como referencia, se incluye el resultado del sistema de rotación de bits la figura 3.23a.

La cobertura mostrada en la figura 3.23b llega hasta 1,7 m de profundidad, frente a los 1,3 m del sistema de rotación de bits. Además, el ancho del sistema de ráfaga cubre 1,4 m (desde $-0,7$ m hasta $0,7$ m), mientras que el sistema rotación de bits cubre 1,0 m (desde $-0,5$ m hasta $0,5$ m). El incremento en la superficie cubierta por este nuevo algoritmo está ligeramente por encima del 80 % con respecto a la superficie cubierta por el algoritmo de rotación de bits.

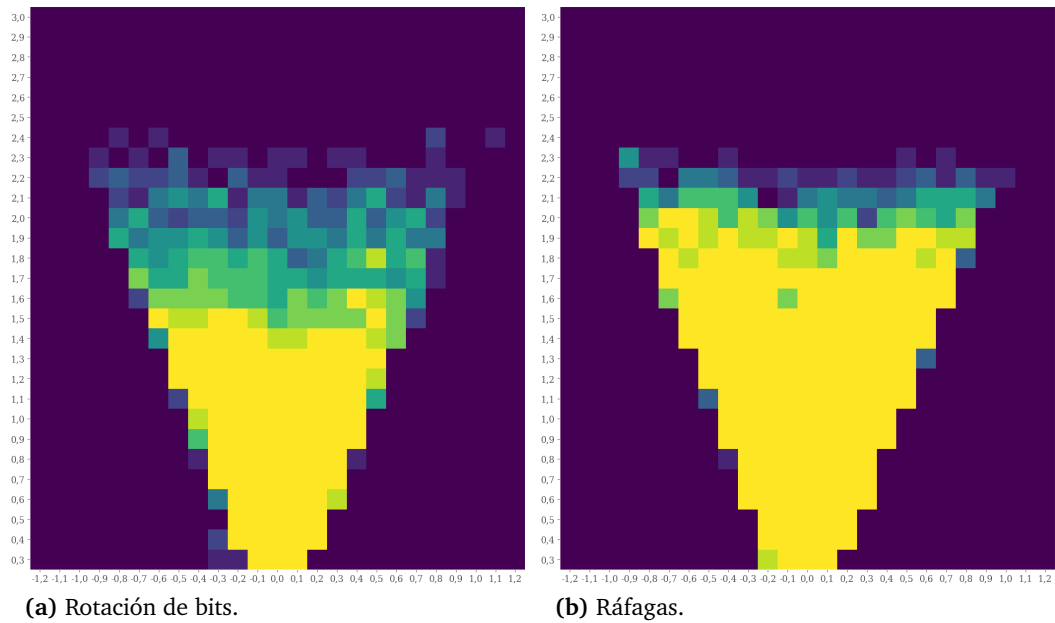


Figura 3.23. Cobertura del sistema de ráfaga.

3.7.4. Doble toma

A pesar de su rendimiento en el proceso de decodificación, con un alcance que llega a 1,7 m con un 100 % de aciertos, el sistema de detección de códigos descrito en la sección 3.7.3 tiene una debilidad que puede ser contraproducente en dos importantes aspectos futuros:

1. cuando este sistema se utilice dentro de un sistema de posicionamiento para estimar la distancia entre emisor y receptor.
2. cuando haya más de una baliza presente en la escena y sea necesario detectar los códigos emitidos por ambas balizas.

Esto se debe a una de las premisas del sistema de detección de códigos actual: que el código empieza y termina donde hay una larga secuencia de ausencia de luz. Uno de los pasos del procesamiento de imágenes al que se somete a cada muestra es la umbralización, tras la que en la muestra sólo quedan dos valores: ausencia de luz o presencia de la misma. Sin embargo, podría darse el caso de que el código emitido por la baliza contuviese franjas negras (es decir, ceros) a izquierda, a derecha, o a ambos lados. En la figura 3.24 se puede ver un caso en el que esta situación ocurre. Este código fue capturado a 1,2 m de distancia. Con el procedimiento de decodificación actual, ambas franjas negras se perderían, y el código no empezaría y terminaría con ceros sino con unos.

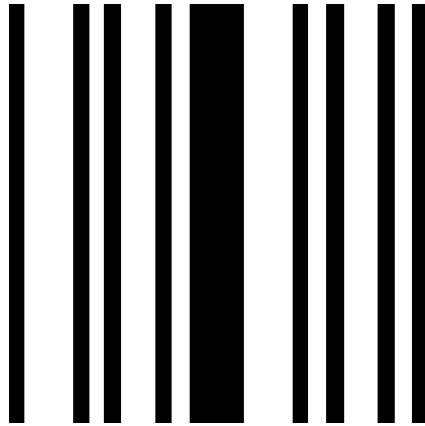


Figura 3.24. Muestras con ceros a ambos lados.

Existe una técnica básica en la industria de los efectos especiales, conocida desde finales de 1880, llamada *matte* [127]. Utilizando diferentes procedimientos, se crea una máscara que, aplicada a la película, permite sustituir con precisión partes del fotograma por otras. Dicha técnica se viene aplicando desde el cine de los hermanos Lumière hasta la actualidad. Posiblemente el lector esté familiarizado con este efecto y lo conozca como el efecto *fondo verde*, nombre con el que se ha hecho popular en la actualidad, o *chroma key*.

Es posible aplicar esta técnica con un propósito diferente: localizar el contorno de la fuente luminosa. Para ello se tomarán dos fotos:

- la primera de ellas, configurando la cámara de manera que la sensibilidad tome el mínimo valor posible. Además, el tiempo de exposición se ajustará para que sea de 0,01 s, puesto que se ha comprobado experimentalmente que con ese valor se consigue que las barras del código provocadas por la combinación del efecto rolling shutter y el parpadeo a alta frecuencia de la luminaria sea prácticamente inapreciable.
- para la segunda, como ya se ha comentado, la sensibilidad tomará el máximo valor posible, mientras que el tiempo de exposición tomará el valor mínimo. De esta forma, el código de barras que se busca será lo más apreciable posible dadas las características de la cámara utilizada.

Ambas fotos, tomadas a una distancia de 1,2 m del emisor, pueden verse en la figura 3.25a y en la figura 3.25b, respectivamente.

Los pasos necesarios para determinar la ubicación del contorno son similares a los descritos en la sección 3.4. Dicho contorno (la figura 3.25a) se utiliza para generar una máscara (la figura 3.25c) que, aplicada al código de barras (la figura 3.25b),

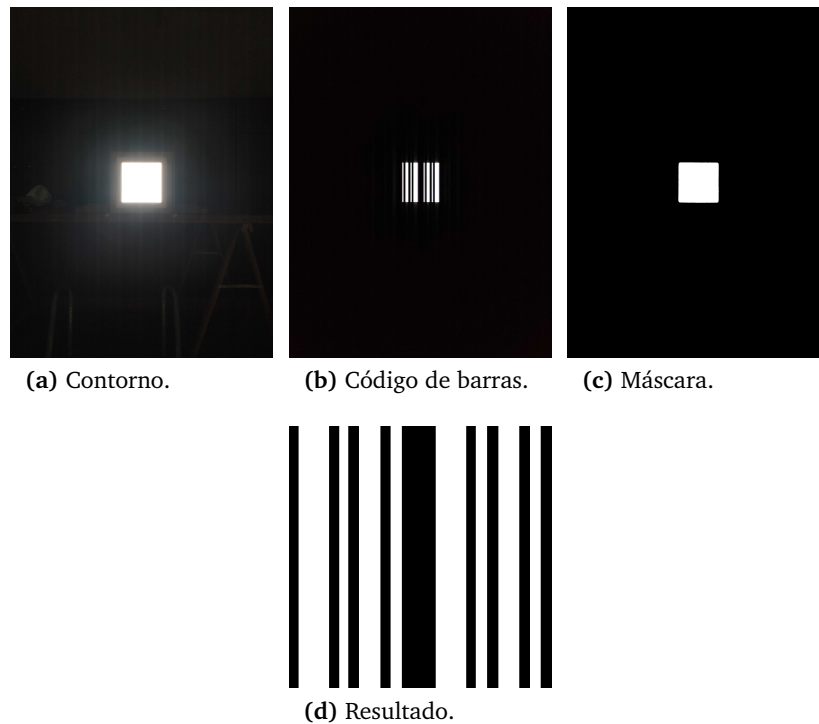


Figura 3.25. Ejemplo de toma de dos muestras.

permite obtener el resultado final (la figura 3.25d) tras seguir los mismos pasos de procesamiento de imágenes que en los algoritmos anteriores. Gracias a este proceso se obtienen varias ventajas directas:

- **ubicación precisa de la baliza:** gracias a la utilización de la máscara no se depende de la presencia de luz en el código de barras para determinar su extensión.
- **imagen reducida:** tras utilizar la máscara para recortar la imagen se trabaja con una imagen de dimensiones más reducidas, disminuyendo así parte del tiempo de procesamiento.
- **posibilidad de trabajar con varias balizas:** la utilización de una máscara abre la posibilidad a trabajar con tantas balizas como sea necesario en la misma imagen, puesto que resulta más precisa la detección de los límites de las mismas.
- **posibilidad de posicionar:** dadas las dimensiones de la baliza, las características físicas de la cámara utilizada para tomar una muestra (dimensiones del sensor, tamaño del píxel y distancia focal), y las dimensiones de la proyección de la baliza en la muestra, resulta posible estimar la distancia entre la cámara

y la baliza, lo que abre la puerta al posicionamiento en interiores, objetivo de este trabajo de tesis.

También presenta algunos inconvenientes:

- **tiempo de proceso:** es necesario tomar dos muestras en lugar de una, lo que implica multiplicar por dos el tiempo necesario de captura de las muestras.
- **consumo energético:** por la misma razón que en el punto anterior, este algoritmo de decodificación tendrá unos requisitos energéticos más altos que el algoritmo utilizado hasta ahora.
- **resultados imprecisos:** si el usuario mueve el dispositivo portable con mucha velocidad en el momento en el que se toman las dos muestras, el desplazamiento del mismo desde que se tomó la muestra para el contorno hasta que se toma la muestra para el código de barras puede provocar que la máscara no coincida con la ubicación de la baliza.

En [99] se utiliza el método de mínimos cuadrados para localizar la circunferencia que mejor se ajuste a la muestra obtenida. Este método, como se reconoce en las conclusiones de dicho artículo, sólo es adecuado para luminarias redondas. El algoritmo propuesto en este trabajo de tesis supera esa limitación, pudiendo procesar con éxito luminarias de cualquier forma y tamaño. El precio a pagar es la necesidad de tomar dos muestras consecutivas con la cámara, utilizando diferentes ajustes.

Para comprobar si la utilización de este algoritmo supone una ventaja con respecto al resto de mejoras se ha modificado la aplicación **Sampler** para Android, incorporando un nuevo experimento que toma las dos muestras necesarias. También se ha modificado el script Python **decoder**. En la figura 3.26 se puede comparar el porcentaje de aciertos en función de la distancia de los cuatro algoritmos de decodificación estudiados hasta el momento. Los resultados mostrados en ella se corresponden con el procesamiento de las muestras a posteriori. Tanto el algoritmo de ráfaga como el del doble toma incorporan la mejora de rotación de bits.

El primero de los algoritmos se limita a tomar muestras y procesarlas. Es la referencia con respecto a la que se mide el rendimiento del resto de algoritmos. Su alcance máximo con un 100 % de aciertos es de 1,1 m. La primera mejora sobre este algoritmo rota los bits del código obtenido, tratando de recuperar información para completar códigos incompletos. Su alcance máximo con 100 % de aciertos es de 1,3 m. Teniendo en cuenta este parámetro, el rendimiento del algoritmo de doble toma es comparable con el de rotación de bits: ambos tienen el mismo alcance con 100 % de aciertos. Sin embargo, el algoritmo de doble toma mejora a partir de los 1,8 m, siendo su tasa de acierto en las detecciones mejor hasta que la señal se pierde por completo.

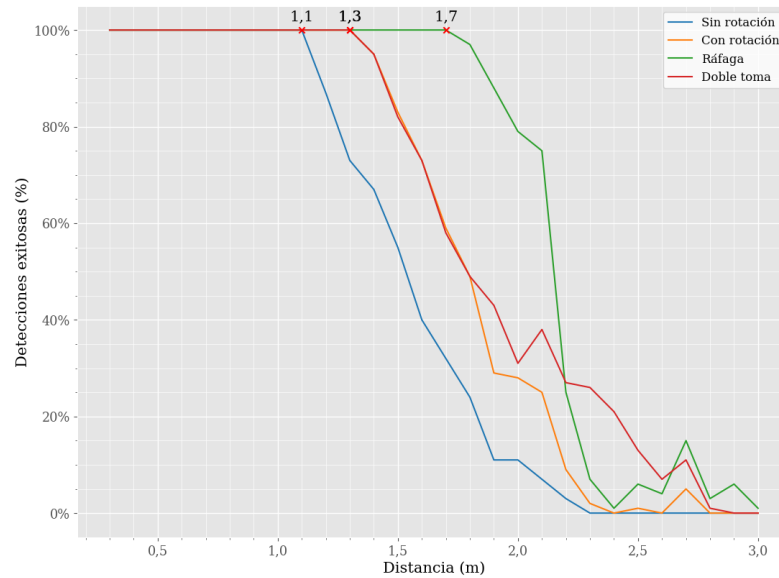


Figura 3.26. Rendimiento con la distancia del algoritmo de toma de dos muestras, comparado con el resto.

El alcance del algoritmo de doble toma es peor que el algoritmo de ráfaga hasta los 2,2 m, momento en el que el de doble toma le supera, aunque se trata de algo poco relevante, puesto que el porcentaje de aciertos empieza en un 25 % hasta la pérdida completa de la señal.

Para poder decidir qué algoritmo utilizar existe un argumento más a tener en cuenta: el tiempo de procesamiento. En la figura 3.27 se puede ver el tiempo medio de procesamiento de muestras de cada uno de los cuatro algoritmos de los que se dispone por el momento.

El algoritmo de ráfagas tarda aproximadamente 2 s en procesar cada conjunto de muestras. Sin embargo, el de doble toma sólo tarda 345 ms de media, por lo que se encuentra dentro de un margen de tiempo aceptable. Por sus múltiples ventajas frente al algoritmo de toma de muestras con rotación de bits, éste parece ser una elección razonable a pesar de tardar el doble en procesar cada conjunto de muestras.

3.7.5. Doble toma + ráfagas

El algoritmo de doble toma implica ventajas muy interesantes. Además, resulta sencillo combinarlo con el algoritmo de ráfaga: en una primera fase se toma la muestra para la obtención del contorno, en la segunda se toma una ráfaga de

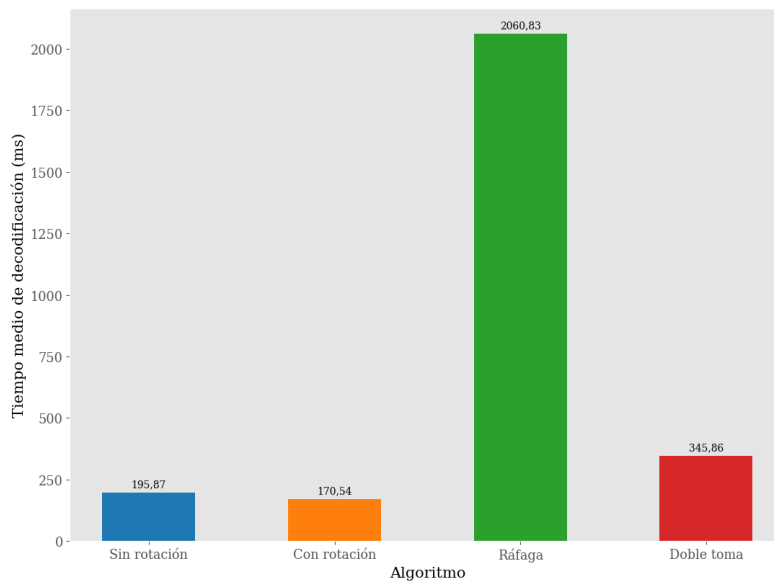


Figura 3.27. Tiempo medio de procesamiento del algoritmo de dos muestras, comparado con el resto.

muestras. El contorno obtenido se puede combinar con las muestras para obtener el código emitido. En la figura 3.28 se puede comparar el porcentaje de aciertos en función de la distancia de los cinco algoritmos de decodificación estudiados hasta el momento.

De todos los algoritmos estudiados hasta ahora, el de doble toma más ráfaga es el que mejores resultados obtiene en las pruebas de alcance, manteniéndose por encima del 90 % de aciertos hasta 2,1 m, y por encima del resto a partir de ese momento.

Atendiendo a la figura 3.27, parece evidente que el precio a pagar seguirá siendo el tiempo de procesamiento, dado que cada ráfaga implica la toma de 10 muestras que posteriormente es necesario procesar. Sin embargo, dado que la utilización de la máscara permite trabajar con imágenes de menor tamaño, cabe preguntarse si se reduce el tiempo medio de procesamiento. En la figura 3.29 se puede ver este tiempo para cada uno de los cinco algoritmos de los que se dispone por el momento. Como se puede comprobar, el tiempo total de procesamiento se ha reducido en un 5 %, aproximadamente.

En el caso del algoritmo de ráfaga, el tiempo medio de procesamiento por muestra es de unos 206 ms, mientras que en el de doble toma más ráfaga es de unos 196 ms. Para averiguar la influencia que tiene el número de muestras en el rendimiento de ambos algoritmos se han tomado ráfagas de 20 muestras desde los 0,3 m hasta los 3 m de distancia. A continuación se han procesado utilizando una versión modificada

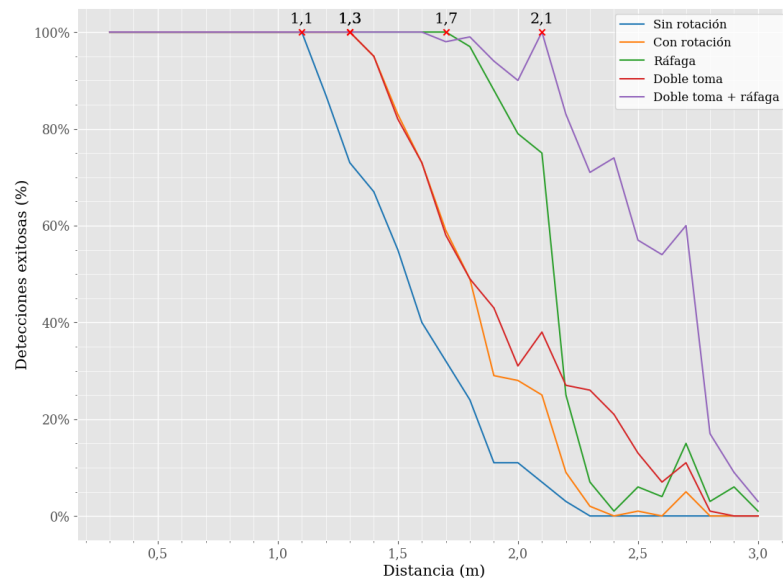


Figura 3.28. Alcance del algoritmo de toma de dos muestras más ráfaga, comparado con el resto.

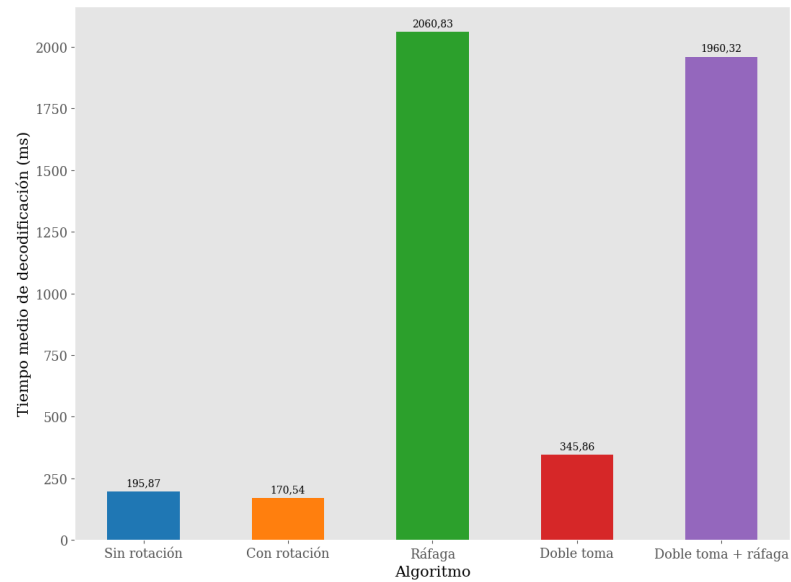


Figura 3.29. Tiempo medio de procesamiento del algoritmo de dos muestras más ráfaga, comparado con el resto.

de este último algoritmo que permite seleccionar cuántas muestras de la ráfaga se quieren utilizar. Dicho algoritmo ha procesado las ráfagas tomadas desde 1 a 20 muestras. El resultado se puede ver en la figura 3.30.

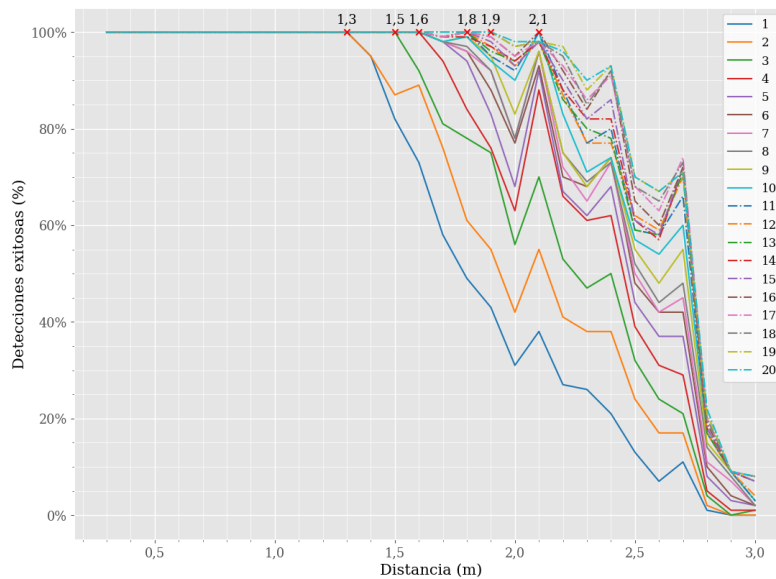


Figura 3.30. Impacto de la distancia en el rendimiento del algoritmo de ráfaga con diferente número de muestras.

Como era de esperar, dado que a mayor número de muestras mayor es la probabilidad de conseguir una en la que la decodificación del código sea posible, el alcance del algoritmo de doble toma más ráfaga es mejor cuanto mayor sea el número de muestras procesadas. Es decir, a partir de cierta distancia existe un factor de probabilidad que determina la factibilidad de decodificación. Sería ideal lograr disminuir este factor probabilístico al mínimo.

3.7.6. Consumo de batería

Llegados a este punto se dispone de una versión del algoritmo de decodificación que funciona de forma nativa en un dispositivo Android, y es capaz de decodificar las muestras a medida que las toma. Resulta interesante conocer el consumo energético del proceso de decodificación por dos razones:

- para conocer el coste en la duración de la batería de la toma de muestras y la decodificación de las mismas.
- para determinar cómo afecta a este parámetro algún futuro cambio en el algoritmo de decodificación.

Con el fin de poder caracterizar el impacto que la decodificación de muestras tiene en la vida de la batería se registrará la evolución la carga de la misma, partiendo de una carga completa, en tres situaciones:

- **referencia:** utilizando una aplicación diseñada específicamente para medir la carga de la batería, sin realizar ninguna otra tarea.
- **sin utilizar:** modificando la aplicación **Sampler** para que mida la carga de la batería sin realizar ninguna otra tarea. Dicha aplicación exhibirá únicamente su comportamiento predeterminado, esto es, mostrar una previsualización de lo que ve la cámara seleccionada, así como la rejilla alineación y el nivel.
- **decodificando muestras:** modificando la aplicación **Sampler** para que tome muestras a intervalos regulares y registre, al mismo tiempo, el nivel de carga de la batería. En este caso, la muestra tomada no se utiliza en el proceso de decodificación. En su lugar, se dispone de una muestra previamente obtenida cuyo resultado se conoce, para que el resultado del proceso de decodificación sea previsible.
- **decodificando ráfagas:** modificando la aplicación **Sampler** para que tome ráfagas a intervalos regulares y registre, al mismo tiempo, el nivel de carga de la batería. En este caso, las muestras tomadas no se utilizan en el proceso de decodificación. En su lugar, se dispone de una muestra previamente obtenida cuyo resultado se conoce, para que el resultado del proceso de decodificación sea previsible.

En los cuatro casos, cada 5 s se guardarán en un archivo en formato de valores separados por comas (*comma-separated values*, **CSV**) dos valores: el instante de tiempo actual y el nivel de carga de la batería. En los dos últimos casos, además, se toma una muestra y se decodifica también cada 5 s. El smartphone se deja durante todo el tiempo que dura el experimento dentro de una caja, a oscuras, para que la pantalla presente la misma intensidad en todos los casos, sin que influya el nivel de iluminación de la habitación en la que se encuentra, afectando de igual forma al consumo total de la batería. Para ello se ha desarrollado una aplicación que se describe en la sección 4.7. Los datos obtenidos se pueden consultar en la figura 3.31.

En estas pruebas, sin utilizar más que la aplicación para la obtención del valor de referencia, la duración de la batería es de 25,48 h. Con la aplicación de toma de muestras sin utilizar, la batería dura 6,97 h. Con la aplicación decodificando muestras la batería dura 6,53 h, mientras que con la aplicación decodificando ráfagas la duración de la misma es de 5,27 h. Si se toma el valor de referencia como el 100 % de la duración de la batería, la aplicación sin utilizar duraría en funcionamiento

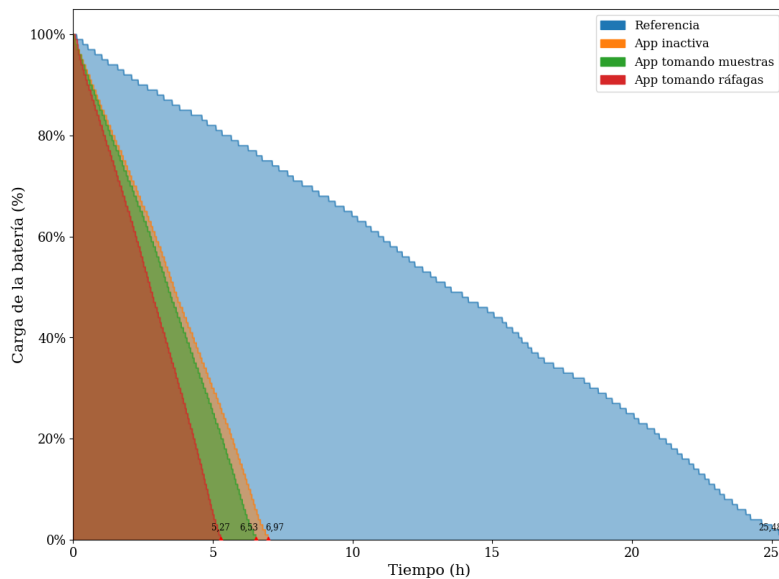


Figura 3.31. Rendimiento de la batería.

continuo el 27,35 % del total, decodificando duraría el 25,63 % del total, y tomando ráfagas el 20,68 % del total. Según estos valores, el proceso de toma de muestras y decodificación de las mismas cuando se dispone de una previsualización de lo que captura la cámara consume un 1,73 % del total, mientras que la toma de ráfagas, la decodificación de las mismas y la elección de la óptima consume un 6,67 % del total. Resulta interesante tener en cuenta que el algoritmo de ráfaga toma 10 fotografías que decodifica, mientras que el de toma de muestras sólo procesa 1. Sin embargo, el algoritmo de ráfaga no es 10 veces más costoso, sólo 3,86 veces.

3.8. Conclusiones

A lo largo de este capítulo se ha desarrollado un sistema de comunicación de datos basado en luz visible que utiliza un panel **LED** gestionado por un microcontrolador como emisor y la cámara de un dispositivo portable como receptor. El sistema resultante tiene un alcance en línea recta de 1,7 m con un 100 % de detecciones y decae paulatinamente hasta que no hay detecciones a los 2,4 m de distancia. Cuando ambos elementos están a 2,0 m de distancia el porcentaje de detecciones está en el 50 %. El tiempo medio de toma de muestras y decodificación de las mismas es de 0,730 s con la cámara trasera y 0,475 s con la frontal.

Resta ahora determinar si este alcance es suficiente para comenzar a desarrollar un sistema de posicionamiento basado en luz visible, donde las balizas estén montadas

en el techo. Como referencia, la altura estándar de los techos en las viviendas [128] es de 2,44 m. La altura media de una persona oscila entre los 1,65 y 1,78 m [129], dependiendo de si se trata de una mujer o un hombre, por lo que se considerará una altura media de 1,71 m. El dispositivo portable se suele sostener de manera casual en una posición situada a unos 2/3 de la altura de la persona, con lo que estaría, de media, a una altura de 1,14 m. Desde esa posición hasta el techo existe una distancia de 1,3 m, aproximadamente, como se puede ver en la figura 3.32. Teniendo en cuenta que el sistema descrito en este capítulo detecta las balizas con un 100 % de éxito hasta 1,7 m de distancia, se estaría trabajando dentro de una zona segura.

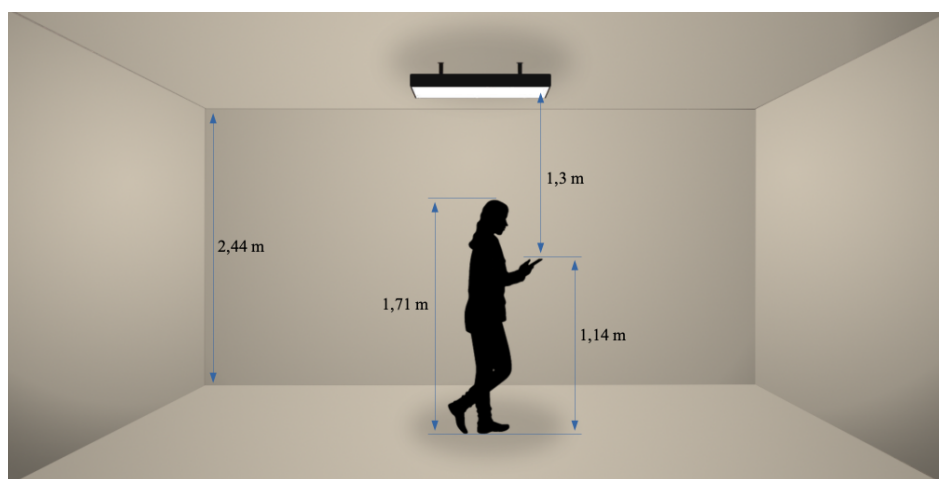


Figura 3.32. Estimación del alcance del sistema de comunicación en una habitación.

Con los datos obtenidos en las diferentes pruebas realizadas, el algoritmo de decodificación escogido es el de doble toma. Su alcance óptimo es de 1,3 m, lo que permitiría su funcionamiento en los márgenes descritos. En aquellos entornos en los que la distancia media sea mayor se podría utilizar el algoritmo de ráfaga. Aunque el algoritmo con rotación de bits tiene un alcance similar y menos tiempo de procesamiento que el de doble toma, la mejor detección del contorno de la baliza jugará un papel importante en el algoritmo de posicionamiento, como se verá en el capítulo 5.

Nos encontramos, por lo tanto, en disposición de avanzar en pos de conseguir un sistema de posicionamiento en interiores basado en luz visible, lo que constituye el objetivo final de este trabajo de tesis. Sin embargo, antes de avanzar en ese frente, sería interesante hablar sobre las diferentes herramientas software que se han desarrollado para alcanzar el objetivo marcado. El siguiente capítulo está dedicado por completo a hablar sobre ellas.

Herramientas software

4.1. Introducción

A lo largo de las siguientes páginas se describen las herramientas que se han creado para codificar los datos que las balizas emitirán y los receptores deberán interpretar, para desarrollar el proceso de decodificación de las señales de manera offline, así como la simulación utilizada para investigar las técnicas de posicionamiento y su precisión. Todas las herramientas presentadas en este capítulo están a disposición del lector en repositorios Git, indicados en sus apartados correspondientes.

4.2. Generador de códigos Manchester

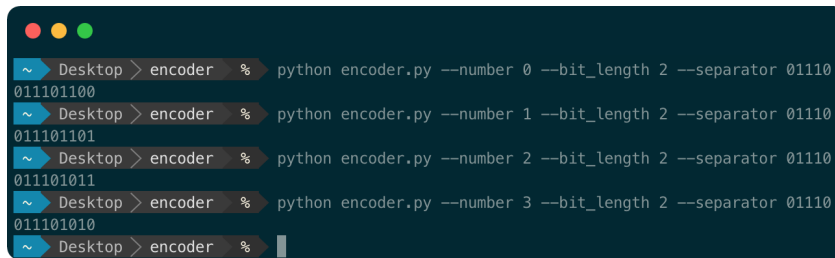
Las razones por las que se eligió esta codificación, ya explicadas en la sección 3.3, se pueden resumir en dos:

- debe ser lo suficientemente compacto como para aprovechar al máximo los recursos disponibles, pero lo suficientemente descriptivo como para evitar ambigüedades.
- todos los códigos creados con el mismo número de bits deben ser indistinguibles por el ojo humano, ofreciendo la misma intensidad lumínica.

El objetivo final es que las balizas desarrolladas dispongan de un mecanismo que permita cambiar el código emitido fácilmente. La intención es que la propia baliza ofrezca un panel de configuración (a través de una interfaz web) en el que sólo haya que introducir el código que la identifique de forma inequívoca. El software de la propia baliza lo almacenará y comenzará a emitirlo.

Sin embargo, antes de llegar a esta situación, se debe comprobar que la codificación elegida es adecuada para resolver el problema, es decir, encontrar un sistema de codificación adecuado para la comunicación de datos entre el emisor y el receptor. Para ello, utilizando el lenguaje de programación Python [130], se ha desarrollado una aplicación para el terminal que recibe una serie de parámetros y devuelve el

código Manchester correspondiente. En la figura 4.1 se puede ver un ejemplo de su funcionamiento.



```
~ Desktop > encoder % python encoder.py --number 0 --bit_length 2 --separator 01110
011101100
~ Desktop > encoder % python encoder.py --number 1 --bit_length 2 --separator 01110
011101101
~ Desktop > encoder % python encoder.py --number 2 --bit_length 2 --separator 01110
011101011
~ Desktop > encoder % python encoder.py --number 3 --bit_length 2 --separator 01110
011101010
~ Desktop > encoder %
```

Figura 4.1. Ejemplo de utilización de la herramienta para la generación de códigos Manchester.

En esta figura se han generado los cuatro códigos que pueden verse en la figura 4.2. Se trata de los valores del 0 al 3, codificados utilizando 2 bits y con la cabecera 01110. Como puede comprobarse, con esta configuración la emisión de 2 bits termina necesitando 9 bits.

El código fuente de esta aplicación puede consultarse en un repositorio de GitLab [131] dentro del grupo de proyectos dedicados a este trabajo [132]. Su funcionamiento se puede resumir en los siguientes pasos:

1. **Analizar los parámetros proporcionados.** Si no se proporciona ninguno, la aplicación utilizará unos valores predeterminados para devolver una solución correcta.
2. **Generar el código Manchester.** El algoritmo utilizado para ello puede verse en la figura 4.3.
3. **Devolver el código generado.** En ese momento se puede copiar y pegar, o reenviar a un archivo de texto o a otra aplicación que lo quiera procesar.

Con lo comentado en este apartado se dispone de una herramienta capaz de convertir el valor numérico que le proporcionemos utilizando la codificación Manchester con un número concreto de bits, y añadiendo al principio del resultado una cabecera que se utilizará como separador.

4.3. Sistema de control de la baliza

La parte software del sistema de control ha sido programada en C++ utilizando PlatformIO [133], un ecosistema de código abierto para el desarrollo de dispositivos

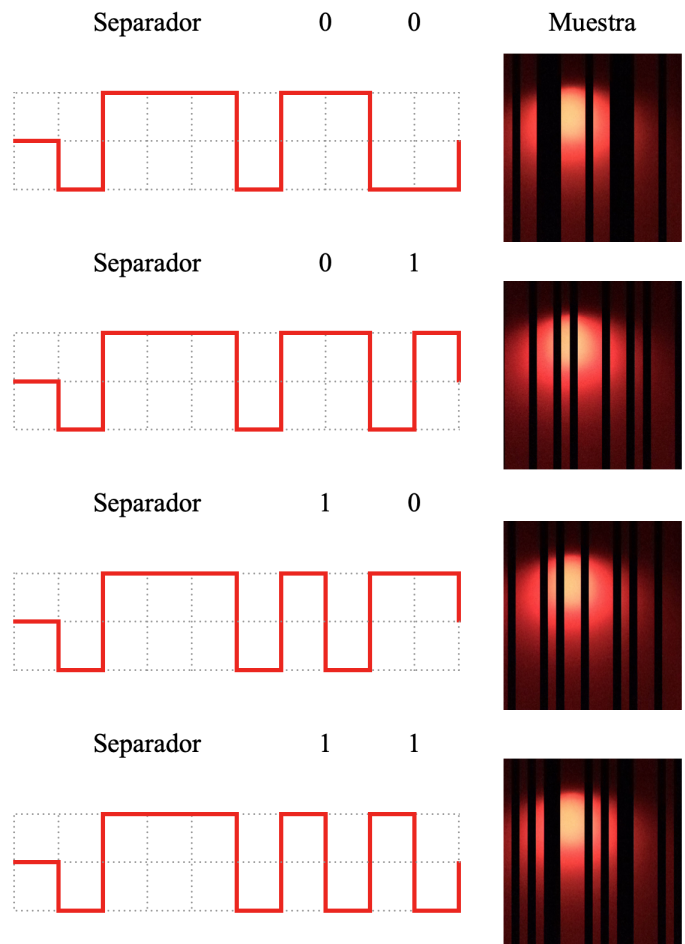


Figura 4.2. Valores del 0 al 3 codificados con dos bits usando Manchester, separador incluido. Se proporciona un ejemplo de cada secuencia [105].

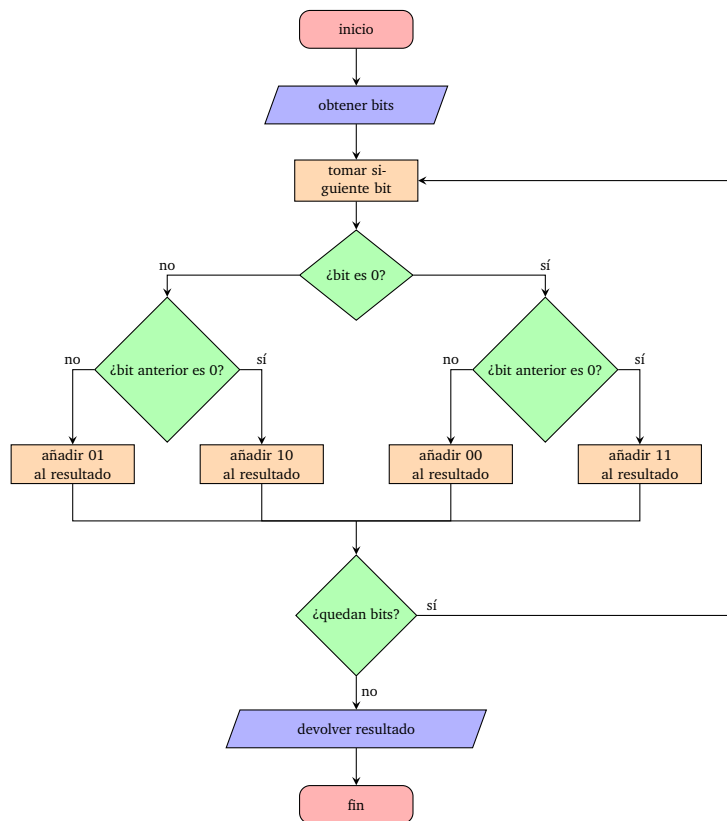


Figura 4.3. Diagrama de flujo del algoritmo propuesto de codificación Mánchester.

del internet de las cosas (*Internet of Things*, **IoT**). El código fuente de esta aplicación puede consultarse en el repositorio de GitLab correspondiente [134], dentro del grupo de proyectos dedicados a este trabajo [132]. Su funcionamiento se puede resumir en los siguientes pasos:

1. El emisor busca una red Wi-Fi a la que poder conectarse. Si no identifica ninguna de las presentes, es posible conectarse al propio emisor mediante una interfaz web para seleccionar una de las redes existentes e introducir las credenciales necesarias, como muestra la figura 4.4.
2. Una vez seleccionada la red Wi-Fi, el microcontrolador se intentará conectar a ella con las credenciales proporcionadas. Tras conectarse, el panel **LED** se encenderá. Desde ese momento, el microcontrolador ofrece un panel web desde el que se puede cambiar el comportamiento de la baliza.

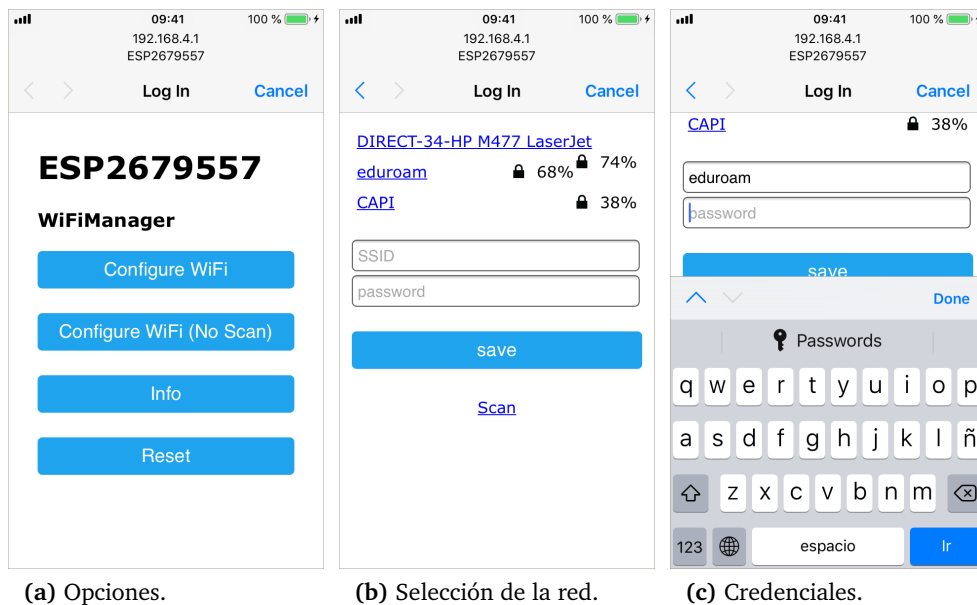


Figura 4.4. Selección de la red Wi-Fi.

Puesto que se está desarrollando un prototipo experimental, la interfaz web ofrecida es básica, proporcionando las funcionalidades mínimas necesarias para cumplir con la función esperada. Si el microcontrolador se ha conectado a la red Wi-Fi configurada obteniendo, por ejemplo, la IP 192.168.1.148, para configurar el panel de manera que emita a una frecuencia de 2,5 kHz tendría que visitarse el siguiente URL:

`http://192.168.1.148/frequency?2500`

Utilizando la herramienta descrita en la sección 4.2 se puede generar el código correspondiente al número 3 utilizando 4 dígitos con la cabecera 01110. El resultado sería 0111011001010. Para que la baliza emita dicho mensaje a los 2,5 kHz anteriormente configurados tendría que visitarse el siguiente URL:

```
http://192.168.1.148/message?0111011001010
```

La baliza permite asignar la cabecera en binario y un código en base 10 en lugar del mensaje en binario completo, simplificando la tarea del usuario. Para cambiar la cabecera predeterminada (01110) por cualquier otra secuencia en binario se puede visitar el siguiente URL:

```
http://192.168.1.148/header?10001
```

El código en base 10 se establecería así:

```
http://192.168.1.148/code?3
```

Adicionalmente, se puede visitar la IP asignada al microcontrolador, sin más, para consultar los parámetros con los que la baliza está configurada actualmente. En la figura 4.5 se pueden ver las peticiones web utilizadas para asignar la frecuencia y el mensaje enviado, y para obtener la configuración actual de la baliza. Además, también se muestra cómo realizar las mismas operaciones desde el terminal.

Aunque básico, este sistema tiene la ventaja de ser fácil de utilizar desde un terminal, lo que permite la creación de sencillos *scripts* de consola para programar el funcionamiento de la baliza.

El software del microcontrolador guarda en su memoria permanente el identificador de grupo de servicios (*Service Set Identifier*, **SSID**) de la red Wi-Fi y la contraseña de acceso a la misma. De esa manera, tras reiniciarlo, si la misma red está disponible intentará volver a conectarse a ella. El resto de parámetros también se guardarán en la memoria permanente del microcontrolador. Así sólo es necesario programar la baliza una vez, recordando los valores proporcionados para su próximo uso.

4.4. Análisis offline

El sistema de procesamiento descrito a continuación es offline y está programado utilizando Python [130] como lenguaje de programación y PyCharm [135] como entorno de desarrollo, lo que proporciona varias ventajas. En primer lugar, permite

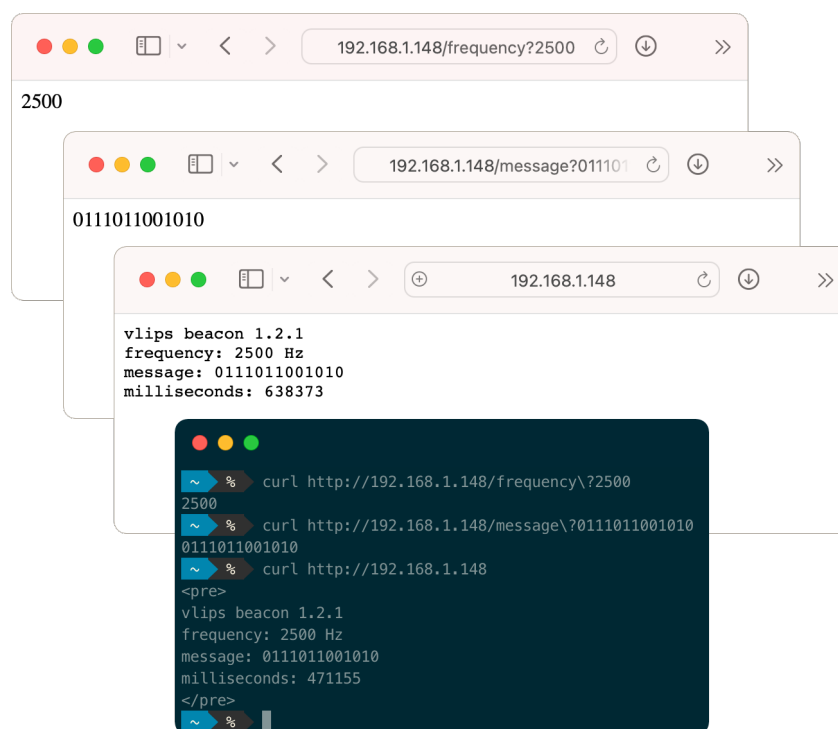


Figura 4.5. Panel de control de la baliza.

iterar con facilidad, es decir, resulta relativamente sencillo crear nuevas versiones tanto del algoritmo de decodificación como de los algoritmos de análisis de los resultados. Si se estuviese trabajando en el dispositivo portable, tras cada cambio sería necesario compilar la aplicación, instalarla en la plataforma de destino y, entonces, ejecutarla y obtener los resultados. Sin embargo, utilizando Python y un sistema offline, tras realizar los cambios en el código fuente basta con pulsar un botón para ejecutar el programa y obtener los resultados, gracias en parte a que Python es un lenguaje interpretado [136]. Poder realizar cambios de forma rápida y sencilla permite avanzar con agilidad en la toma de muestras, su procesamiento y posterior análisis. En segundo lugar, facilita la **repetición** de las pruebas, lo que permite disponer de un conjunto de muestras estable, sólido y controlado, pudiendo volver a ejecutar el proceso de decodificación y análisis tantas veces como se considere oportuno para comprobar si existen diferencias entre los resultados obtenidos. De esta manera es posible determinar si un cambio supone una mejora, incorporándolo entonces al sistema, o si por el contrario implica un retroceso, con lo que se podrán buscar alternativas. Por último, es **portable**: utilizando este sistema existe la posibilidad de llevarse la muestras a otro ordenador y poder realizar allí el proceso, e incluso resulta posible proporcionárselas a otros investigadores para que trabajen con ellas.

El decodificador de códigos Manchester ha sido desarrollado pensando en automatizar las tareas todo lo posible. Se ha desarrollado una herramienta para la línea de órdenes que ofrece dos modos de funcionamiento: la decodificación de imágenes individuales y la de carpetas que contengan imágenes.

Ambos modos necesitan como primer parámetro el código que se estaba emitiendo, para poder así compararlo con el resultado obtenido y determinar si la decodificación ha tenido éxito o no. Por ejemplo, si el código emitido fuese el 10, el parámetro sería `--code 10`.

Si se desea analizar una única imagen, basta con utilizar el parámetro `--image` seguido de la ruta en la que se encuentra la imagen a analizar. Dicha ruta puede ser relativa o absoluta a la ubicación de la herramienta. Por ejemplo, `--image_path samples/range/10/120/03.jpg` serviría para analizar una única imagen.

Si, por el contrario, se quieren analizar todas las imágenes contenidas dentro de una carpeta, el parámetro sería `--directory_path samples/range/10`. La aplicación analizará todas y cada una de las imágenes contenidas en dicha carpeta, tratando de decodificarlas.

El resultado del proceso se guardará en una carpeta llamada **results**. Si se trata de analizar una imagen, dicha carpeta estará al mismo nivel que la imagen analizada. Si se trata de analizar una carpeta, estará dentro de ella. En esta carpeta se guardará un archivo de **CSV** con todos los datos obtenidos, en crudo. El nombre de los archivos incluye una marca de tiempo, de manera que diferentes ejecuciones de la aplicación no borren resultados anteriores, pudiendo así compararlos.

El archivo de datos generado se utiliza como punto de partida para el análisis de los resultados, que se describen en la sección 3.6. Como ejemplo, el *script hits_and_misses.py* genera una serie de archivos **CSV** e imágenes para poder estudiar el rendimiento del sistema. Contienen información sobre los aciertos conseguidos, así como el nivel de confianza asociado. Además, se generarán dos gráficos representando tanto aciertos como nivel de confianza. En la figura 4.6 se pueden ver los archivos generados tras analizar los contenidos de una carpeta. Si se tratase del resultado de analizar un archivo, el nombre del mismo se hubiese incluido en el nombre de cada uno de los archivos generados.

Sin embargo, es posible que estos archivos no sean suficientes. En algunas ocasiones es necesario examinar el resultado de todos y cada uno de los pasos dados durante el proceso de decodificación para conocer más a fondo su funcionamiento, y así localizar las causas de los posibles problemas e intentar solucionarlos. Para ello, basta con utilizar el parámetro `--debug`. Como resultado, se generará una nueva

Name	Date Modified	Size	Kind
confidence_level_plot_2020-04-09-19-05-51.png	9 April 2020 at 19:06	19 KB	PNG image
hits_2020-04-09-19-05-51.csv	9 April 2020 at 19:06	375 bytes	CSV Document
hits_plot_2020-04-09-19-05-51.png	9 April 2020 at 19:06	20 KB	PNG image
raw_data_2020-04-09-19-05-51.csv	9 April 2020 at 19:06	5 KB	CSV Document

Figura 4.6. Archivos generados tras analizar el contenido de una carpeta.

carpeta llamada **debug**, al mismo nivel que la carpeta **result**. En su interior se encuentran varias carpetas, una por cada muestra analizada. Dentro, una imagen por cada paso dado durante en análisis de la muestra, así como un archivo de texto con datos importantes para el proceso de decodificación. En la figura 4.7 se puede ver el contenido de esta carpeta.

Name	Date Modified	Size	Kind
00-result.png	10 April 2020 at 00:51	252 KB	PNG image
01-source_image.png	10 April 2020 at 00:51	14,5 MB	PNG image
02-flipped_source_image.png	10 April 2020 at 00:51	14,5 MB	PNG image
03-grayscale_image.png	10 April 2020 at 00:51	5,2 MB	PNG image
04-blurred_image.jpg	10 April 2020 at 00:51	1,6 MB	JPEG image
05-thresholded_image.png	10 April 2020 at 00:51	278 KB	PNG image
06-thresholded_reduced_image.png	10 April 2020 at 00:51	222 bytes	PNG image
07-resized_thresholded_reduced_image.png	10 April 2020 at 00:51	551 KB	PNG image
08-barcode_image.png	10 April 2020 at 00:51	270 KB	PNG image
09-data.txt	10 April 2020 at 00:51	1 KB	text

Figura 4.7. Archivos de depuración generados tras analizar el contenido de una muestra.

El código fuente de esta aplicación puede consultarse en el repositorio de GitLab correspondiente [137], dentro del grupo de proyectos dedicados a este trabajo [132].

4.4.1. Otras herramientas

Además del proceso de decodificación realizado por el script **decoder.py**, descrito en la sección 3.4, el proyecto completo ofrece muchas otras herramientas, entre las que se pueden destacar:

- **hits_and_misses.py**: dada la ruta de un archivo **CSV**, genera una serie de archivos **CSV** y gráficos que pueden utilizarse para analizar el alcance del sistema de comunicación. Sólo necesita un parámetro, **data path** (`-d`, `--data_path`), la ruta al archivo **CSV** de entrada.

- **heatmap.py**: dada la ruta de un archivo **CSV**, genera una serie de archivos **CSV** y gráficos que pueden utilizarse para analizar la cobertura del sistema de comunicación. Sólo necesita un parámetro, **data path** (-d, --data_path), la ruta al archivo **CSV** de entrada.
- **interferences.py**: dada la ruta de un archivo **CSV**, genera una serie de archivos **CSV** y gráficos que pueden utilizarse para analizar la influencia de una segunda baliza en el campo de visión de la cámara durante el proceso de decodificación. Sólo necesita un parámetro, **data path** (-d, --data_path), la ruta al archivo **CSV** de entrada.
- **battery.py**: dada la ruta de una carpeta en la que hay una serie de archivos **CSV**, los trata como datos que definen áreas, y los muestra todos en el mismo gráfico.
- **compare scripts**: dada la ruta de una carpeta en la que hay una serie de archivos **CSV**, los muestra todos en el mismo gráfico. El tipo de gráfico es diferente dependiendo del tipo de gráfico a comparar.

Es posible encontrar más información sobre el proyecto, todas las herramientas que ofrece y cómo utilizarlas, en el archivo **README.md** [138] del proyecto.

4.5. Análisis online

En la sección 4.4 se describen las ventajas de crear un sistema de análisis offline utilizando Python. Estas ventajas permiten que el sistema pueda evolucionar con rapidez. Pero, por desgracia, utilizar un sistema offline tiene un gran inconveniente: no se ajusta a los objetivos marcados. El sistema que se está investigando debe ser capaz de funcionar por completo desde el dispositivo portable. Se busca desarrollar una aplicación capaz de tomar las muestras, decodificar los datos emitidos por las balizas y estimar la posición en la que dichas muestras fueron tomadas desde un dispositivo móvil. Los requisitos mínimos de dicha aplicación serían los siguientes:

- debe mostrar lo que vé la cámara, para poder apuntarla correctamente.
- debe poder tomarse una fotografía.
- el archivo resultante debe poder compartirse.

Para cumplir dichos objetivos se ha desarrollado una aplicación Android utilizando Android Studio como entorno de desarrollo y Kotlin como lenguaje de programación. En el momento de escribir estas líneas, los dos lenguajes principalmente utilizados para esta tarea son Java y Kotlin [139]. Dado que este último está en alza y

tiene todos los visos de convertirse en el lenguaje preferido para el desarrollo de aplicaciones Android [140], es el que se ha escogido.

El funcionamiento básico de la aplicación se describe a continuación. Al ponerse en marcha pide permiso al usuario para utilizar la cámara, como se aprecia en la figura 4.8a. En la figura 4.8b se pueden observar las opciones de configuración, a las que se accede tocando en los tres puntos que aparecen en la esquina superior derecha de la pantalla principal. Dicha pantalla se puede ver en la figura 4.8c. En ella se pueden apreciar las guías y un botón llamado **Take Samples**, que cumple una función doble. Al tocarlo comienza el proceso de toma de muestras y, además, cambia su texto para informar al usuario sobre el progreso del proceso de toma de muestras, como se ilustra en la figura 4.8d.

Además, en la figura 4.8c se puede apreciar un detalle interesante: se ha incluido un nivel en las guías para facilitar la correcta alineación del dispositivo portable, que debe estar vertical y perpendicular al plano. Dicho nivel permite ajustar los ángulos de elevación (*pitch*) y alabeo (*roll*). Se trata del segmento que aparece inclinado en el centro de la figura 4.8c, abandonando su posición como parte inferior del rectángulo central de la guía, donde sí está en la figura 4.8d con el dispositivo portable está correctamente alineado.

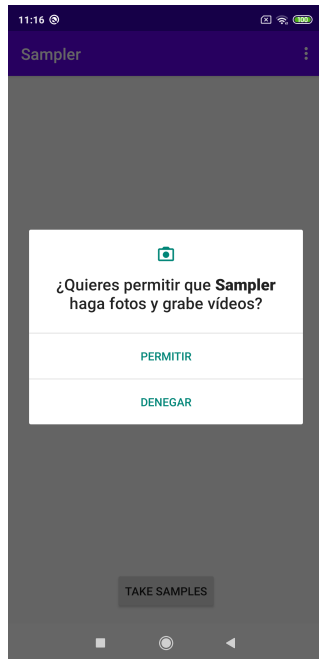
Por último, la aplicación se puede configurar a través de sus preferencias para que suene una alerta cuando el proceso de toma de muestras concluya. De esa manera, cuando el número de muestras a tomar es alto, el usuario puede dejar la aplicación desatendida sabiendo que, al terminar, será avisado. Sin embargo, dicha opción está desactivada por defecto, como puede verse en la figura 4.8b.

El código fuente de esta aplicación puede consultarse en el repositorio de GitLab correspondiente [141], dentro del grupo de proyectos dedicados a este trabajo [132].

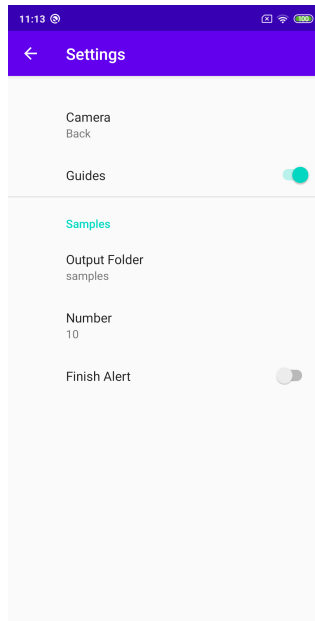
4.5.1. Decodificación

La misma aplicación para la toma de muestras descrita en la sección 4.5 se ha modificado para realizar decodificación en tiempo real. Su aspecto ha sufrido algunos cambios que pueden verse en la figura 4.9.

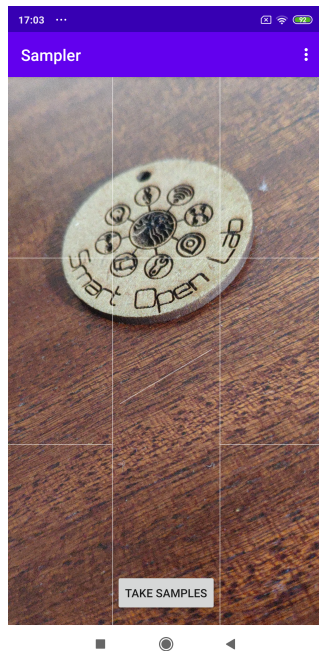
El primer cambio que se puede apreciar está en la pantalla principal de la aplicación, que muestra la figura 4.9a. En ella hay dos botones nuevos en su parte superior. Ambos botones activan nuevos modos de funcionamiento que sirven para explorar el entorno, sin guardar ningún dato.



(a) Permisos.



(b) Opciones.

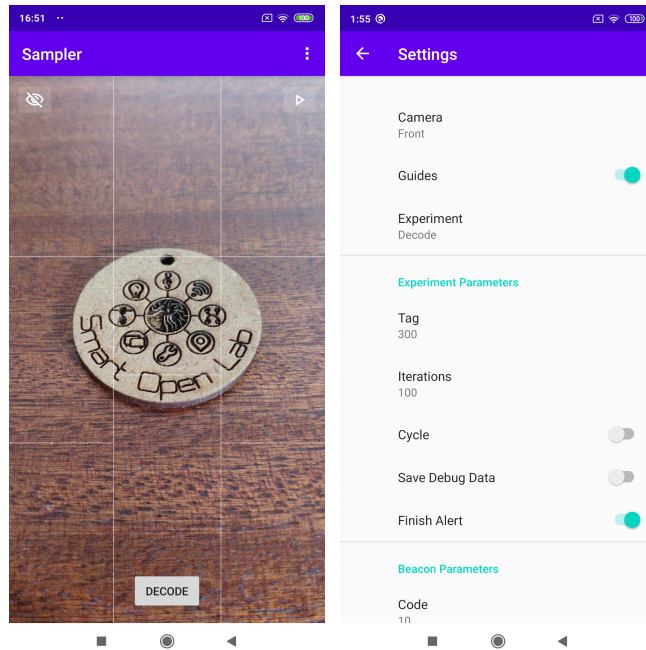


(c) Nivel.



(d) Toma de muestras.

Figura 4.8. Aplicación para la toma de muestras.



(a) Pantalla principal.

(b) Opciones.

Figura 4.9. Aplicación para la toma de muestras con decodificación nativa.

El botón de la izquierda permite activar o desactivar la previsualización. Cuando está desactivada, la aplicación presenta lo que la cámara ve sin ninguna modificación de su configuración. Sin embargo, cuando está activada, la aplicación presenta lo que se verá en las muestras, dado que tanto sensibilidad como tiempo de exposición se han ajustado. Ambos modos de funcionamiento pueden compararse en la figura 4.10. Por una parte, la figura 4.10a muestra la aplicación con la previsualización desactivada, mientras que en la figura 4.10b está activada.

El segundo botón nuevo, en la esquina superior derecha, sirve para activar la decodificación en tiempo real. En este modo de funcionamiento no se guardan datos. Cuando está activado, la aplicación toma muestras y las analiza en busca de un código emitido. El resultado se muestra en el centro de la parte superior, entre los dos botones nuevos. Por ejemplo, en la figura 4.10c la baliza está emitiendo el código 10. La aplicación muestra $\{10=2\}$, lo que significa que ha detectado el código 10 dos veces en la muestra tomada.

Cuando el modo de decodificación en tiempo real está activado el resto de botones se desactivan, para que la única función que la aplicación puede llevar a cabo sea dicha decodificación. Si se desea utilizar alguna de las otras funciones se deberá desactivar la decodificación en tiempo real.



Figura 4.10. Efecto de utilizar los botones de la parte superior.

El segundo cambio realizado consiste en reorganizar las opciones de configuración de la aplicación, añadiendo de paso algunas nuevas, como muestra la figura 4.9b. Existen tres categorías de opciones:

- **General:** permite configurar las siguientes opciones de la aplicación:
 - **Camera:** qué cámara utilizar (frontal o trasera).
 - **Guides:** determina si las guías y el nivel se mostrarán en la pantalla principal o no.
 - **Experiment:** para seleccionar el experimento que se realizará con la aplicación. Existen tres opciones:
 - **Take Samples:** toma de muestras, con el mismo comportamiento descrito en la sección 4.5.
 - **Decode:** nuevo experimento para determinar el rendimiento de la decodificación en el dispositivo móvil.
 - **Battery:** nuevo experimento para determinar el consumo de batería del proceso de decodificación.
 - **Battery Baseline:** nuevo experimento para determinar el consumo de batería de la aplicación en funcionamiento, sin realizar más operación que la previsualización de lo que la cámara ve.

- **Experiment Parameters:** parámetros de los experimentos. Cuatro opciones que permiten modificar el comportamiento de los experimentos:
 - **Tag:** una etiqueta que se utilizará para marcar los datos obtenidos.
 - **Iterations:** número de veces que el experimento se realizará.
 - **Cycle:** activar o desactivar la rotación de bits en el proceso de decodificación.
 - **Save Debug Data:** si se activa, además de los datos del experimento se guardarán otros datos adicionales que pueden utilizarse para examinar el funcionamiento de la aplicación.
 - **Finish Alert:** si se activa, al terminar el experimento se hará sonar una alarma.

- **Beacon Parameters:** parámetros de las balizas. Contiene opciones para describir la baliza con la que se trabajará:
 - **Code:** código emitido por la baliza.
 - **Message Header:** cabecera del mensaje emitido.
 - **Message Length:** longitud en bits del mensaje emitido.

4.5.2. Procesamiento

Independientemente del experimento seleccionado, su nombre será el que aparezca en el texto del botón de la parte inferior de la pantalla principal de la aplicación. Al poner en marcha el experimento, todos los botones se desactivarán hasta que termine. El experimento se repetirá tantas veces como se haya indicado en la opción de **Iterations**. El texto del botón irá cambiando a medida que el experimento avance, mostrando la iteración actual y el número de iteraciones a cumplir.

Al poner en marcha el experimento **Decode**, se tomarán muestras con la cámara seleccionada en las opciones de configuración. El resultado será una carpeta con el nombre indicado en **Tag**, dentro de la que habrá tantas fotos como iteraciones se indicasen en las preferencias de la aplicación. Las fotos serán tomadas con la cámara seleccionada, el nombre de las mismas será el de la iteración actual. Dicho nombre estará relleno con ceros por la izquierda para que todos los nombres tengan la misma longitud y sea sencillo ordenarlos. Así, la vigésimoprimer fotografía tomada, si el número de fotos a tomar es 100, se llamará **021.jpg**.

Al poner en marcha el experimento **Decode**, se intentará decodificar el mensaje enviado por la baliza que esté en el campo visual de la cámara seleccionada en las opciones de configuración. En cada iteración se guardará el resultado obtenido en

una línea de un archivo **CSV**, así como una marca de tiempo del instante en el que se ha tomado la muestra. Este valor se utilizará para determinar el tiempo medio que lleva la toma de una muestra y su decodificación. El nombre de este archivo será **data_YYYY-MM-DD_HH_MM_SS.csv**, donde la parte final del nombre es la marca de tiempo del instante en el que el experimento comenzó. Estará dentro de la carpeta **results**, en el interior de una carpeta que se creará con el nombre indicado en **Tag**.

Por ejemplo, para probar el rango de alcance del sistema en ambos experimentos se ha configurado la aplicación para que use la cámara trasera y muestre las guías. Se tomarán 100 muestras, no se guardarán datos de depuración, y se hará sonar la alerta. El código emitido es el **10**, la cabecera **01110** y la longitud del mensaje será de 8 bits. El dispositivo portable se coloca a 0,3 m de distancia en línea recta de la baliza, las superficies de ambas paralelas entre sí. El valor de **Tag** en ese momento será **030**. Se toca en el botón para realizar el experimento. Cuando este concluya, se cambia la posición del dispositivo portable, ahora a 0,4 m de distancia, y el valor de **Tag** por **040**. Los pasos se repiten hasta llegar a los 3 m de distancia.

4.5.3. Tiempo de procesamiento

Con los datos obtenidos de los experimentos realizados con la primera versión de esta aplicación para Android se sabe que el tiempo medio dedicado a la toma de una muestra y su decodificación utilizando la cámara trasera es 0,730 s, mientras que si se utiliza la cámara frontal el tiempo medio es 0,475 s, debido a la diferencia en las dimensiones de las muestras obtenidas.

Para obtener estos tiempos se situó el dispositivo portable a 1,2 m de distancia de la baliza, en línea recta y paralelas las superficies emisora y receptora. Se tomaron 100 muestras utilizando el experimento **Decode**, que lleva a cabo el proceso de decodificación en el propio dispositivo. Una de las columnas del archivo resultante contiene la marca de tiempo, expresada en milisegundos, del instante en el que se llevo a cabo cada paso del experimento.

Con estos datos, resulta posible tomar una muestra cada segundo, utilizando el tiempo restante para las operaciones necesarias para actualizar la interfaz de usuario. Es más, la cámara frontal permite realizar este proceso hasta dos veces por segundo.

4.5.4. Posicionamiento

Además de para tomar muestras con las que evaluar los diferentes sistemas de comunicación de datos desarrollados en este trabajo, de los que se habla en la sección 3, Sampler también permite tomar muestras con las que estudiar el rendimiento del sistema de posicionamiento. Para ello la aplicación ofrece dos herramientas: 1) un sistema de calibración para ubicar el dispositivo portable lo más centrado posible dentro de un banco de pruebas en el que la baliza está sobre el plano en el que se coloca el dispositivo portable para determinar su posición, y 2) una serie de procesamientos para poder tomar muestras que posteriormente analizar para realizar el posicionamiento. El sistema de calibración también realiza un posicionamiento en tiempo real.

Antes de realizar procesamiento alguno relacionado con el posicionamiento es necesario realizar una calibración. En la figura 4.11 se puede ver el banco de pruebas utilizado para desarrollar el sistema de posicionamiento.

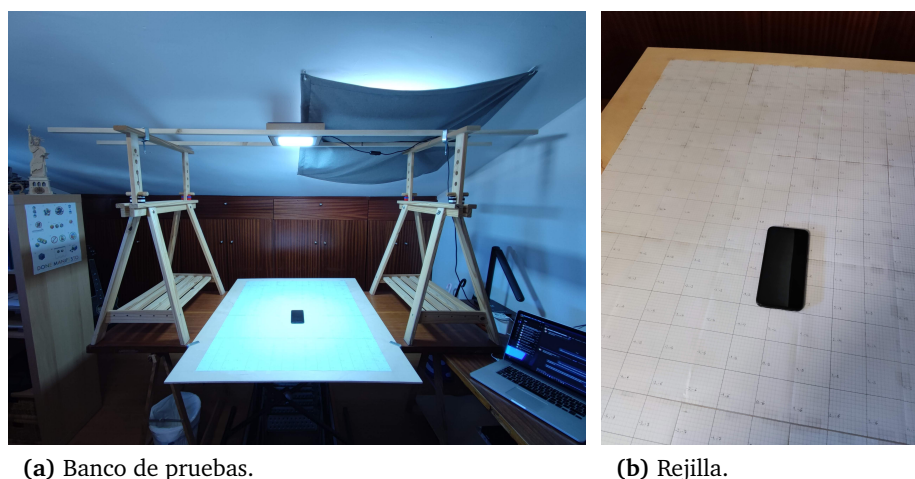


Figura 4.11. Banco de pruebas para el posicionamiento.

La baliza se encuentra sobre una superficie marcada con casillas de 5 cm de lado, con sus superficies paralelas y centradas. Para conseguir alinear el centro de ambos elementos, la aplicación detecta la baliza y localiza su centro y sus esquinas, mostrándolos en la interfaz de usuario tal y como se puede ver en la figura 4.12. Además, bajo la previsualización, se incluyen datos para ayudar a la calibración. En primer lugar, y de forma textual, los pasos que habría que dar para centrar mejor la baliza. En este caso habría que desplazarla hacia la izquierda y arriba, y girarla a la izquierda. En segundo lugar, se muestran los valores utilizados para crear esas instrucciones: el desplazamiento del centro y la inclinación de la baliza. Por último,

se muestra una estimación de la ubicación del dispositivo portable dentro de la rejilla.

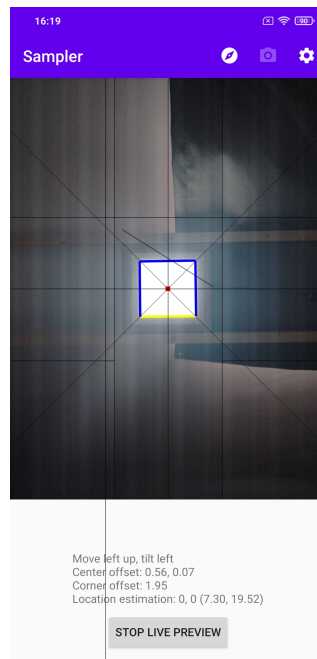


Figura 4.12. Calibración del banco de pruebas.

El funcionamiento del procesamiento incluido en la aplicación es muy similar a los utilizados para probar el sistema de comunicación de datos: se asignan una serie de valores en las preferencias de la aplicación y, posteriormente, se toman muestras en las diferentes posiciones a probar de la rejilla. Se puede encontrar más información sobre el sistema de posicionamiento en la sección 5.

4.6. GetSensorData

En los apartados anteriores de esta sección hemos descrito el funcionamiento de la aplicación desarrollada para la toma de muestras y la decodificación online. Dicha aplicación cumple con su función y ha permitido el desarrollo de toda una serie de experimentos descritos en la sección 3.6 y la sección ??.

Por otra parte, dentro del Grupo LOPSI [142] perteneciente al Consejo Superior de Investigaciones Científicas (CSIC), se ha desarrollado una aplicación para Android llamada GetSensorData [22], capaz de capturar los datos de los sensores presentes en el dispositivo: el acelerómetro, el giroscopio, el magnetómetro, el barómetro, el sensor de luz, el de proximidad, el de temperatura, el giroscopio, etcétera. También

utiliza como fuente de datos sensoriales el **GNSS**, la conectividad Wi-Fi, la conectividad Bluetooth o el micrófono del dispositivo portable. Además, existe la posibilidad de utilizar sensores externos conectados al dispositivo como, por ejemplo, lectores de identificación por radiofrecuencia (*Radio Frequency Identification*, **RFID**) o sensores inerciales. En la figura 4.13a se puede ver el aspecto de `GetSensorData` mostrando la lista de sensores, mientras que en la figura 4.13b se la puede ver mostrando tanto las características de dichos sensores como los datos proporcionados por ellos. Es posible configurar ciertos aspectos del funcionamiento de la aplicación a través de sus preferencias, como frecuencias de actualización, colores o si se buscan balizas iBeacon o no. La pantalla de preferencias puede verse la figura 4.13c, mientras que en la figura 4.13d puede observarse cómo se cambia el valor de una de ellas.

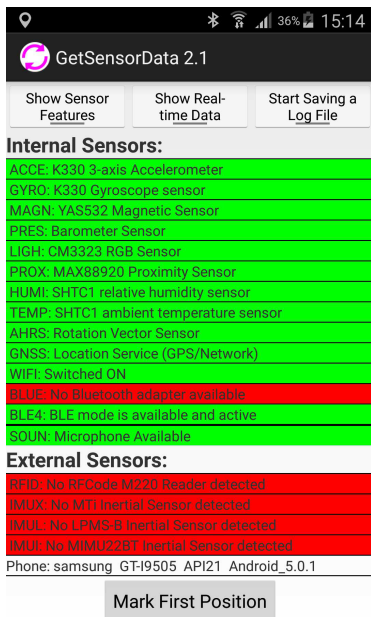
Además de mostrar las características y los datos de los sensores, la aplicación es capaz de guardar un registro en archivo de los datos de los sensores, juntos con las marcas que el usuario considere oportunas al hacer clic en un botón de la interfaz de usuario. Esta aplicación se utiliza en las competiciones organizadas en las diferentes ediciones del **IPIN** [143].

Durante este trabajo se ha incluido en la aplicación `GetSensorData` la posibilidad de adquirir información a través de las cámaras del dispositivo portable. La aplicación `Sampler` podría aprovechar los datos de los sensores con los que `GetSensorData` trabaja si decidiésemos utilizar algún tipo de fusión sensorial para el posicionamiento. Incluir la funcionalidad de cámara de `Sampler` en `GetSensorData` sólo puede tener consecuencias positivas y garantiza que el trabajo realizado para desarrollar `Sampler` pueda ser aprovechado por más personas en trabajos futuros, además del que suscribe.

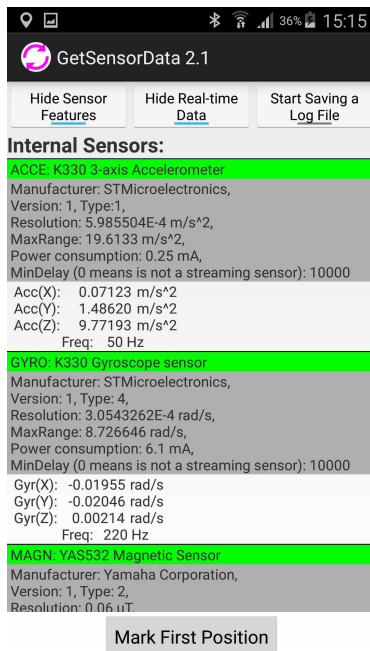
4.6.1. Cambios realizados

El proyecto, en resumen, es el fruto del trabajo de muchas personas distintas, con diferentes niveles de conocimiento sobre programación, con objetivos variados y no todos con la misma atención al detalle. Por tanto, el código es muy heterogéneo.

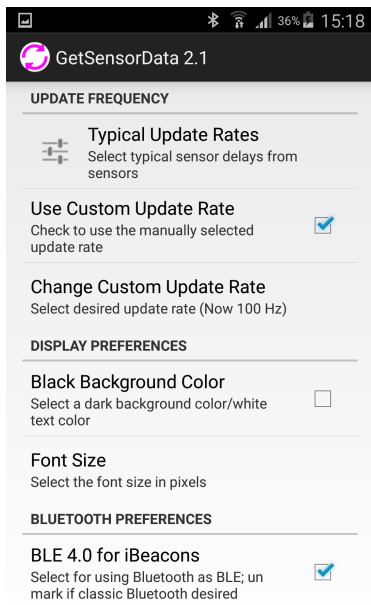
A lo largo de los años 2020 y 2021 se han estado realizando diferentes cambios en `GetSensorData` con el único objetivo de optimizarla. Estos cambios pueden clasificarse dentro de dos categorías: los relacionados con la administración del proyecto y aquellos que afectan a la forma en la que está programado.



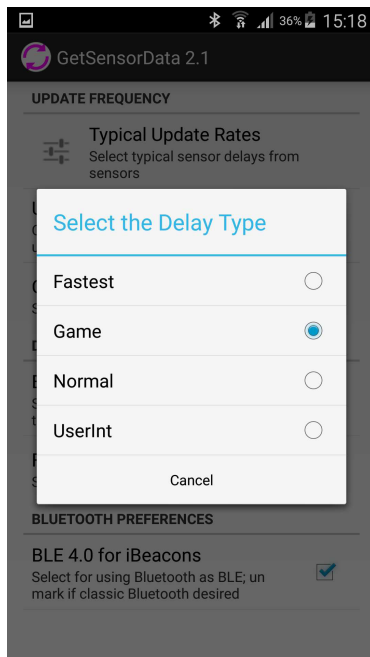
(a) Lista de sensores.



(b) Características y datos en tiempo real.



(c) Lista.



(d) Cambio.

Figura 4.13. Aspecto de la aplicación GetSensorData [22].

Administración

Los cambios relacionados con la administración del proyecto determinan la forma en la que los encargados del mismo realizarán las diferentes operaciones de mantenimiento necesarias para que pueda mantenerse vigente tanto tiempo como sea posible:

- **control de versiones:** para facilitar el trabajo de múltiples desarrolladores distribuidos en ubicaciones distantes se ha publicado el código fuente del proyecto en GitLab [144], un gestor web de repositorios Git con muchas otras capacidades. Utilizar un sistema de control de versiones permite visitar la historia del código, con lo cual no es necesario dejar fragmentos comentados cada vez que se quiera cambiar algo. Cada cambio queda asociado al usuario que lo ha realizado, proporcionando la merecida autoría al trabajo de cada colaborador.
- **organización en repositorios:** el proyecto se ha dividido en varios repositorios, como puede verse en la figura 4.14.

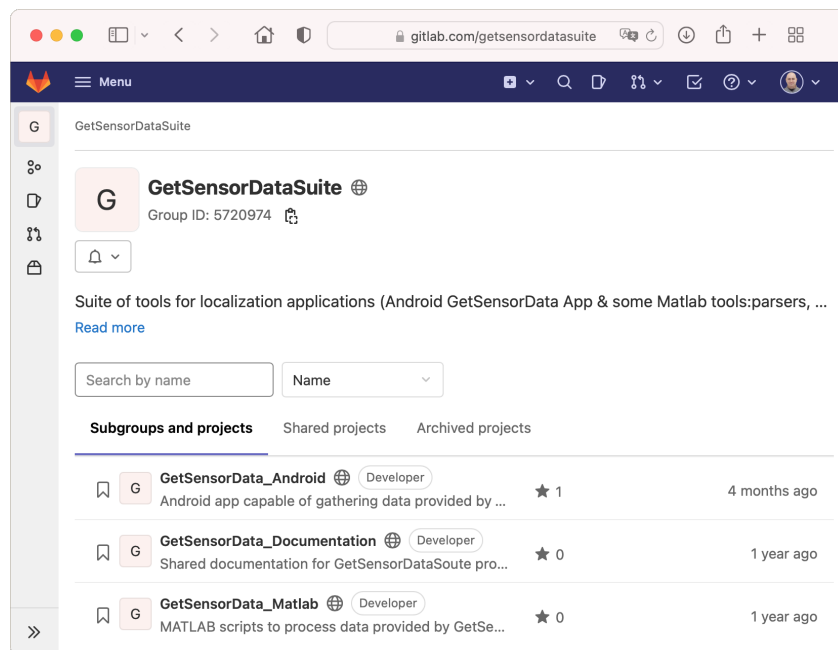


Figura 4.14. Repositorios de GetSensorData.

El primero de ellos contiene el código fuente. Su primera entrada, **GetSensorData_Android** se corresponde con la versión inicial del proyecto, antes de empezar a trabajar con ella. **GetSensorData_Documentation** es un repositorio dedicado única y exclusivamente a la documentación externa del proyecto,

del que se habla en el siguiente punto. Por último, **GetSensorData_Matlab** contiene herramientas de MATLAB [145] para trabajar con el proyecto.

Esta organización cumple dos funciones. Por una parte, divide el proyecto en diferentes elementos que es más sencillo comprender y gestionar. Por otra, abre la puerta a futuros proyectos relacionados como, por ejemplo, una versión de la aplicación para iOS que replique, en la medida de lo posible, el funcionamiento de la versión para Android.

- **documentación:** en un mismo repositorio se encuentra toda la documentación necesaria para comprender el proyecto: guías de usuario, de programador, normas a seguir para utilizar Git, etcétera. Además, se incluye una lista de tareas pendientes que sirve para controlar qué tareas quedan por hacer, o son deseables; además, proporciona sugerencias para aquellos que quieran colaborar voluntariamente con el proyecto. Gracias a la utilización de Git, no es necesario que estos colaboradores formen parte del repositorio, pudiendo enviar su aportación en forma de *pull requests* [146].

Código

Los cambios relacionados con el código del proyecto son difíciles de cuantificar en todos y cada uno de los aspectos que se mostrarán a continuación. Cuando sea posible se proporcionarán cifras, como número de líneas o de archivos. Sin embargo, existen casos en los que el cambio se ha realizado a lo largo de varios meses, de forma paulatina, y se aplica a casi todo el código fuente del proyecto como, por ejemplo, la utilización de nombres significativos. Sirva como medida de la magnitud de estos cambios las advertencias que el entorno de desarrollo integrado (*Integrated Development Environment, IDE*) de Android Studio proporciona sobre cada archivo abierto. Inicialmente, sólo sobre el archivo **Mainactivity.java** se ha realizado mejoras significativas, como puede observarse en la figura 4.15.

En la esquina superior derecha de cada una de las dos ventanas de Android Studio en la figura 4.15 se puede ver que el archivo original presentaba 183 advertencias importantes, 12 menos serias y más 1000 errores tipográficos.

Los cambios realizados en el código son los siguientes:

- **nombres significativos:** los nombres de todas las variables se han revisado para que sigan el mismo formato, *camel case* [147]. Además, cada método tiene un nombre que describe su tarea claramente, facilitando la lectura del código.

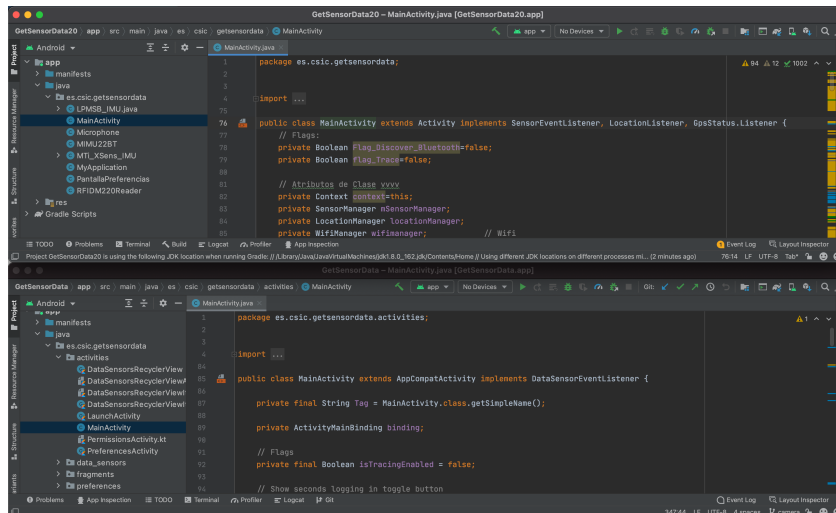


Figura 4.15. Consejos de Android Studio.

Todos los nombres están escritos en inglés, favoreciendo así la participación de más investigadores en el desarrollo del proyecto.

- **funciones pequeñas y concretas:** todas y cada una de las funciones del proyecto se han dividido en funciones más pequeñas que realizan una tarea definida y concreta, con lo que resulta más sencillo saber dónde es necesario realizar las modificaciones del proyecto.
- **comentarios:** se han eliminado todos los comentarios superfluos, a la vez que se ha añadido documentación interna a clases y métodos para facilitar la comprensión del proyecto. Un ejemplo de este tipo de comentarios puede verse en la figura 4.16.
- **formato:** ahora, todos los archivos presentes en el proyecto siguen las mismas normas de estilo.
- **límites:** las fronteras entre las diferentes partes del proyecto se han definido claramente, dividiendo clases demasiado cargadas en clases más pequeñas. El archivo **MainActivity.java**, que superaba las 3000 líneas, no llega ahora a las 1200. Aunque aspectos de su funcionalidad se han trasladado a otras partes de la clase, una parte importante de la reducción se debe a la reutilización de código, eliminando repeticiones.
- **lógica y presentación:** se ha separado la lógica de la aplicación de la representación de la misma. Por ejemplo, los sensores no contienen código relacionado con la manera en la que sus datos se presentan en la pantalla del dispositivo

```

/**
 * Register the listener of this class so its can be notified when something of interest
 * happens. Do any other actions needed to connect to the underlying data sensor.
 *
 * Best suited to be called in an activity's onCreate() method.
 *
 * @param listener; reference to the listener of the events generated by this class, not by
 * `SensorManager`.
 */
open fun connect(listener: DataSensorEventListener) {
    this.listener = listener
}

/**
 * Unregister the listener of this class. Do any other actions needed to connect to the
 * underlying data sensor.
 *
 * Best suited to be called in an activity's onDestroy() method.
 */
open fun disconnect() {
    this.listener = null
}

```

Figura 4.16. Documentación interna.

móvil. A su vez, las actividades sólo se comunican con sus sensores a través de interfaces.

- **organización:** el contenido del proyecto se ha reorganizado de manera que sea más sencillo encontrar cualquier elemento del mismo, como puede verse en la figura 4.17.

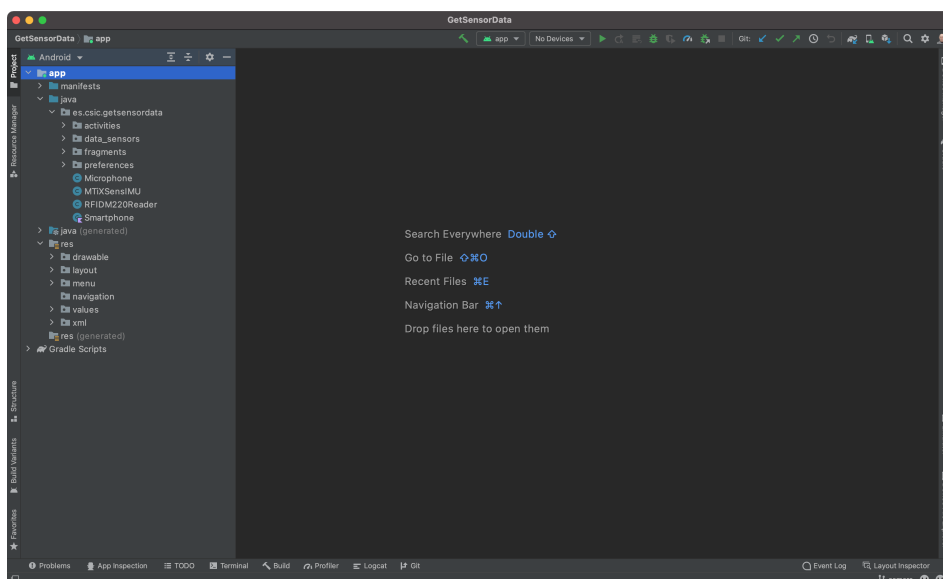


Figura 4.17. Organización en Android Studio.

Esta división es mucho más importante en la nueva versión del proyecto, puesto que ahora el número de archivos ha pasado de los 8 que incluía la versión inicial del proyecto a unos 40. Entre ellos, los archivos **README** y **CHANGELOG**,

que proporcionan una introducción a `GetSensorData` y describen los cambios que incorpora cada una de las versiones. Por sí solo, el número de archivos del proyecto no es un indicativo de calidad, pero si comparamos esta versión con la anterior, sí que sirve como demostración de que se ha realizado una división de la funcionalidad en partes más pequeñas, más fáciles de comprender.

- **clases:** aunque todos los cambios descritos hasta el momento son importantes, posiblemente el más significativo sea la creación de una nueva jerarquía de clases, que se presenta aquí de forma simplificada. Dicha jerarquía pretende abstraer el funcionamiento de los sensores. La idea principal es proporcionar una interfaz común aplicable a todos los sensores, internos y externos, que facilite el trabajo con ellos. Gracias a esta interfaz se puede compartir código y simplificar el proceso de añadir nuevos sensores al proyecto.

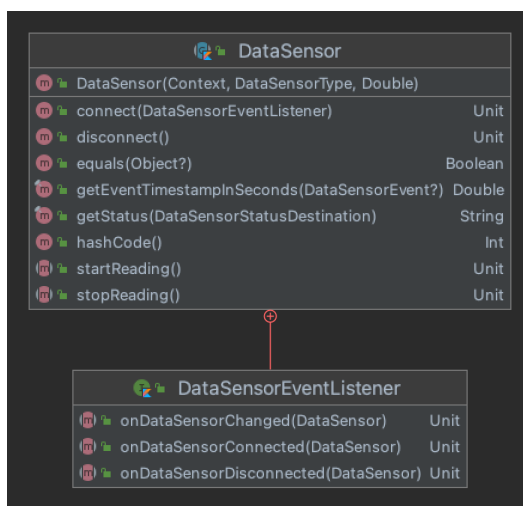


Figura 4.18. Clase `DataSensor`.

El funcionamiento de esta clase es muy simple. El primer paso consiste en crear una clase que herede de `DataSensor` y rellenar los métodos con el código apropiado. En la figura 4.19 se puede ver el comienzo de la clase creada para el acelerómetro del dispositivo portable.

Una vez creada la subclase, el funcionamiento es siempre el mismo, independiente del sensor. Al comenzar la aplicación es necesario llamar al método `connect()`, que conecta el sensor. Al terminar la aplicación, hay que llamar al método `disconnect()`, que libera los recursos asignados al ejemplar de la clase. Mientras la aplicación esté en funcionamiento, el método `startReading()` permite empezar a leer datos del sensor, mientras que el método `stopReading()` detiene la lectura.

```

package es.csic.getsensordata.data_sensors

import ...

class AccelerometerDataSensor(context: Context, updateInterval: Double) :
    ManagedDataSensor(context, DataSensorType.Accelerometer, updateInterval) {

    override fun getName(): String =
        if (sensor != null) {
            sensor.name
        } else {
            "No Accelerometer detected"
        }

    override fun getFeatures(): String =
        if (sensor != null) {
            """
            | {"Manufacturer": ${sensor.vendor},
            | {"Version": ${sensor.version}, Type: ${sensor.type},
            | {"Resolution": ${sensor.resolution} m/s^2,
            | {"MaxRange": ${sensor.maximumRange} m/s^2,
            | {"Power consumption": ${sensor.power} mA,
            | {"MinDelay (0 means is not a streaming sensor)": ${sensor.minDelay}
            """
            .trimMargin()
        } else {
            "No Features"
        }
    }
}

```

Figura 4.19. Clase AccelerometerDataSensor.

La clase `DataSensor` está acompañada por una interfaz llamada `DataSensorEventListener`. Cualquier clase que siga esta interfaz será capaz de recibir los eventos del sensor, disparados cuando el sensor se conecta (`onDataSensorConnected()`), se desconecta (`onDataSensorDisconnected()`) o tiene datos que mostrar (`onDataSensorChanged()`). Al crear el sensor, éste recibe como parámetro una referencia a un ejemplar de una clase que sigue dicha interfaz, que será la encargada de atender a los eventos disparados por el sensor. De esta manera, cada vez que el sensor tenga datos que mostrar avisará de ello, y la clase correspondiente podrá pedirle los datos para mostrarlos utilizando el método `getStatus()`, que recibe como parámetro el destino de los datos: pantalla o archivo de registro. Dependiendo del destino, el formato de los datos será diferente.

Gracias a este diseño, es posible tratar a todos los sensores por igual, en lugar de necesitar gestionar los diferentes eventos por separado, ahorrando así mucho esfuerzo de programación.

La principal ventaja de la utilización de esta jerarquía de clases es que, mientras se sigan unas sencillas normas, añadir un nuevo sensor a la aplicación, interno o externo, no supondrá ninguna dificultad.

- **presentación:** otro cambio de envergadura realizado en el código ha consistido en sustituir la utilización de `TextViews` con un `RecyclerView`. Este componente no es más que una lista de elementos. Su principal ventaja es que para añadir un nuevo sensor sólo es necesario proporcionarle más datos que mostrar, en lugar de tener que crear una nueva colección de elementos visuales. Además,

permite mantener separada lógica y presentación, algo cuya importancia se mencionó anteriormente.

4.6.2. Descripción de resultados

En la figura 4.20 se puede comparar la nueva versión de GetSensorData (la figura 4.20a) con la original (la figura 4.20b). Como se aprecia, su aspecto es prácticamente idéntico, a pesar de las numerosas y profundas modificaciones por las que su código fuente ha pasado. Que no hubiese cambios estéticos apreciables era uno de los objetivos que se tenían al comenzar a trabajar en la aplicación.

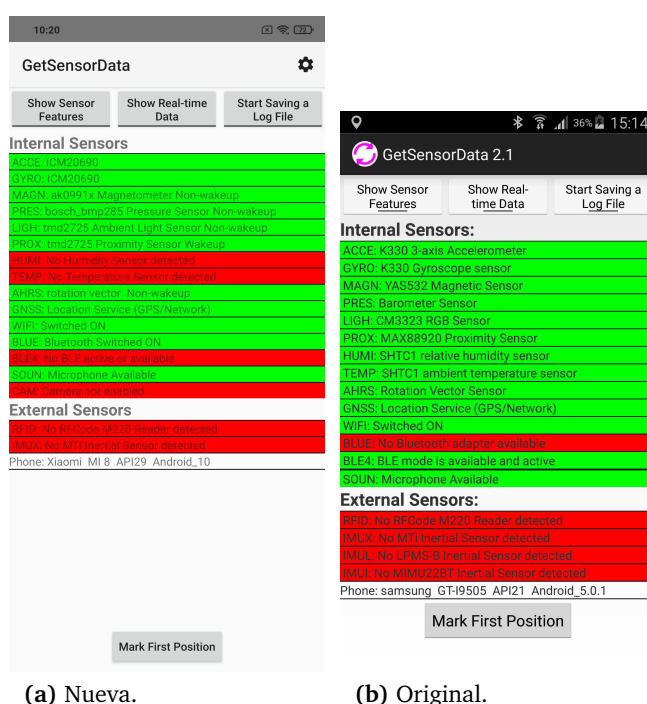
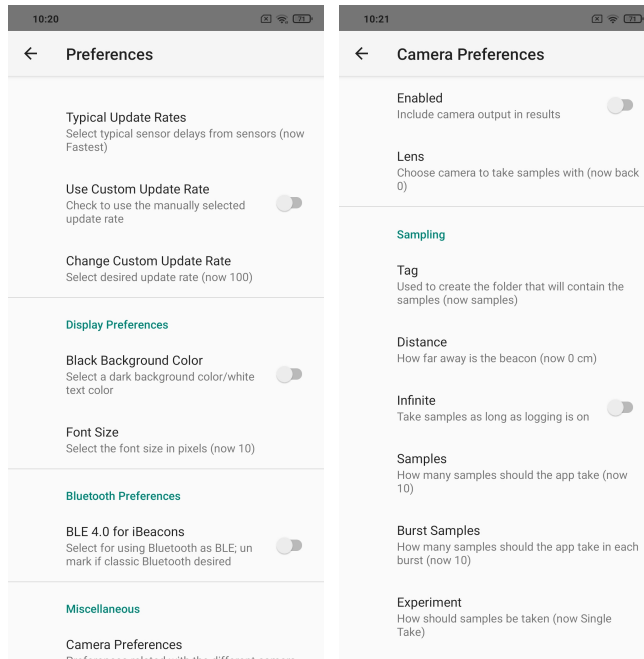


Figura 4.20. Aspecto de las diferentes versiones de la aplicación GetSensorData.

Un par de sensores externos, ahora obsoletos, han desaparecido, mientras que uno nuevo, la cámara, se ha añadido. El mayor cambio estético se encuentra en la esquina superior derecha de la aplicación, donde se ha añadido un icono con forma de engranaje (⚙️) para permitir acceder directamente a las preferencias de la aplicación. Si se hace clic en este botón se podrán ver pantallas como las que muestra la figura 4.21.

La nueva pantalla principal de preferencias (ver la figura 4.21a) muestra las mismas opciones que la versión original salvo por la final, llamada **Camera Preferences** que, a su vez, permite acceder a las preferencias de la cámara. Desde estas preferencias



(a) Principal.

(b) Cámara.

Figura 4.21. Nuevas preferencias de GetSensorData.

se pueden configurar las mismas opciones descritas en la sección 4.5 sobre la aplicación Sampler, cuyas primeras opciones se pueden ver en la figura 4.21b, ya que GetSensorData incorpora ahora toda la funcionalidad de dicha aplicación. Si se activa la cámara y la previsualización, la pantalla principal mostrará el aspecto que puede verse en la figura 4.22. En este caso, se está mostrando en la esquina inferior derecha de la interfaz de usuario.

4.7. Consumo de batería

Con el fin de poder caracterizar el impacto que la decodificación de muestras tiene en la vida de la batería del dispositivo portable se ha desarrollado una aplicación específica y modificado la aplicación de toma de muestras que registrará la evolución de la carga de la misma. Se puede encontrar más información al respecto del consumo de batería en la sección 3.7.6.

La apariencia de la aplicación para la obtención de la referencia del nivel de batería puede verse en la figura 4.23. Dicha aplicación comenzará a registrar el nivel de carga de la batería en mismo momento en el que se ponga en marcha. Estos datos se guardarán en un archivo que luego se podrá copiar para su posterior análisis. El

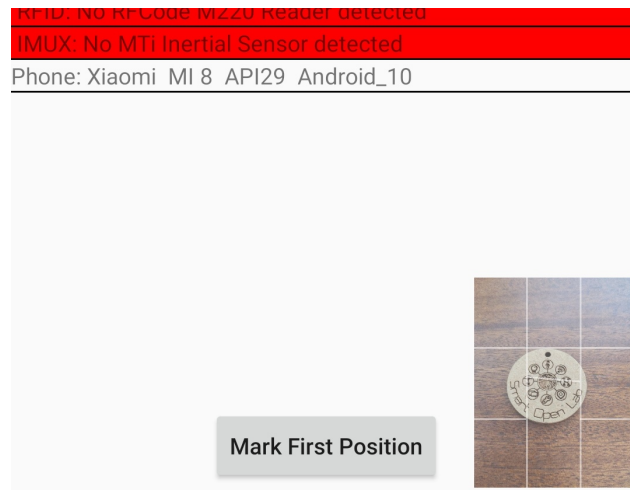


Figura 4.22. Detalle de la previsualización de cámara en GetSensorData.

nombre del archivo incluye la fecha y la hora en la que la toma datos comenzó, de forma que los resultados de varias ejecuciones de la aplicación puedan mantenerse.

Como puede verse en la figura 4.23a, la aplicación no tiene más elementos en su interfaz que el botón para acceder a la configuración de la misma. Esta configuración es muy simple, permitiendo únicamente seleccionar el intervalo entre lecturas del nivel de batería.

La aplicación empezará a registrar en un nuevo archivo el nivel de batería sí:

- se modifica el intervalo entre lecturas en la configuración.
- se cambia a otra aplicación.
- se muestran las aplicaciones abiertas.

El código fuente de esta aplicación puede consultarse en el repositorio de GitLab correspondiente [148], dentro del grupo de proyectos dedicados a este trabajo [132].

4.8. Simulación

Idealmente, para este trabajo se dispondría de un entorno experimental preparado para poder situar emisor y receptor a distancias concretas. Entonces, se realizarían las pruebas necesarias para determinar si es posible estimar la distancia entre ellos a partir de los datos obtenidos. Lamentablemente, no se trata de algo práctico debido a los costes (tanto económicos como temporales) derivados de intentar todas las

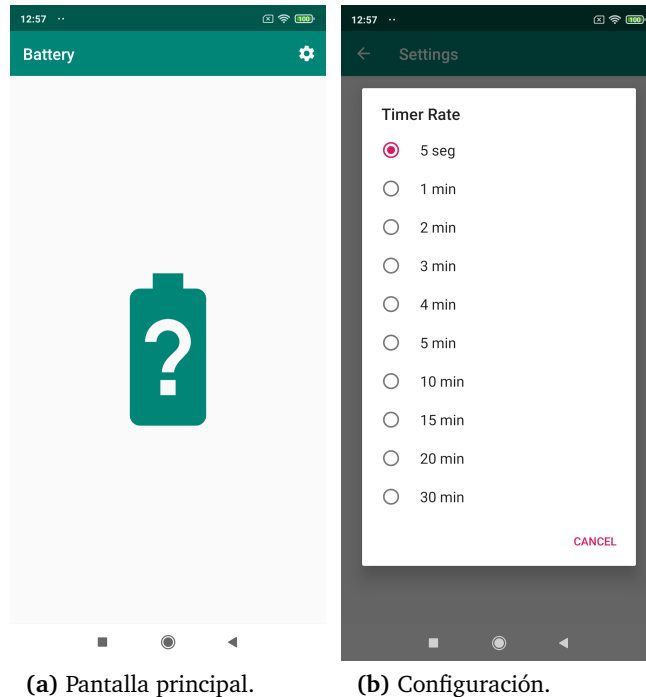


Figura 4.23. Aplicación para tomar los valores de referencia de consumo de la batería.

combinaciones necesarias. La mejor manera de enfrentarse a este tipo de problemas consiste en comenzar con una simulación y, posteriormente, aplicar lo aprendido con garantías de éxito al mundo real.

Blender [149] es un software de creación 3D multiplataforma, abierto y gratuito. Se utiliza en multitud de proyectos de diversa índole (publicidad, películas, juegos, etcétera) y resulta especialmente adecuado para ayudar a solucionar este problema por tres de sus características:

1. **Blender permite simular la realidad.** Con Blender se pueden crear objetos y cámaras, determinando detalles como su tamaño, orientación, posición o distancia focal con precisión (por sólo mencionar algunas de las características que es posible controlar).
2. **Blender es programable.** Es posible utilizar Python [150] para crear y colocar objetos en un entorno tridimensional, así como para cambiar sus propiedades, lo que lo convierte en la herramienta ideal para intentar solucionar este problema.
3. **Blender es gratuito.** Esta característica es perfecta dado que no supone un coste adicional.

En la figura 4.24 se puede ver Blender en funcionamiento.

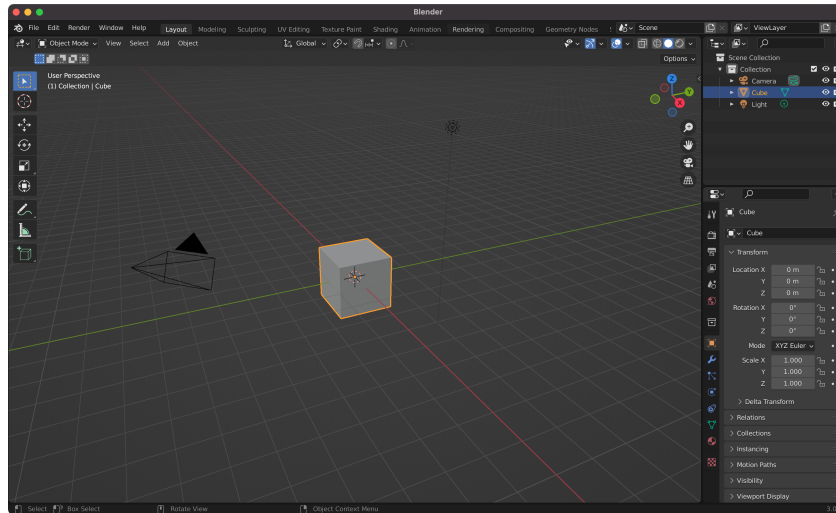


Figura 4.24. Blender en funcionamiento.

El entorno que se desea simular tiene las siguientes características:

- la cámara está situada en el origen de coordenadas.
- las características de la cámara en Blender son las mismas que las de la cámara que se desee simular.
- la cámara está apoyada en el plano, mirando hacia arriba, simulando lo que vería la cámara del dispositivo portable si estuviese apoyada en el suelo, con la pantalla mirando al techo.
- el panel LED es un cuadrado situado sobre la cámara con sus superficies paralelas.

En este trabajo se está utilizando un smartphone desarrollado por Xiaomi, el Mi 8 [120]. Dicho smartphone dispone de dos cámaras, una frontal y una trasera. Por una parte, la cámara frontal se ha construido a partir del sensor S5K3T1 [151] desarrollado por Samsung [152]. Por la forma en la que el integrador ha utilizado este sensor, sus características son:

- ancho: 5184 px.
- alto: 3880 px.
- tamaño del píxel: $0,9\mu\text{m}$.
- distancia focal: 3,52 mm.

Por otra parte, la cámara trasera se ha construido a partir del sensor IMX363 Exmor RS [123] desarrollado por Sony [153]. En este caso, el integrador lo ha incluido con las siguientes características:

- ancho: 4032 px.
- alto: 3024 px.
- tamaño del píxel: 1,4 μm .
- distancia focal: 4,216 mm.

La distancia focal es propia de cada integrador, ya que depende de la separación que haya decidido dejar entre el sensor óptico y las lentes con las que decidan equipar el dispositivo portable.

La toma de muestras permite renderizar lo que la cámara ve desde su posición y guardar el resultado en un archivo. El nombre del archivo se corresponde con la secuencia de toma de muestras, con ceros añadidos a la izquierda para que la longitud de todos los nombres sea la misma. De esta forma se podrán examinar y procesar en el mismo orden en el que las muestras fueron tomadas. Por otra parte, se utilizan los metadatos de cada imagen para almacenar los detalles de cada muestra. En concreto, las muestras están en formato JPEG [125] y los metadatos en formato Exif [124]. Uno de los campos de Exif, llamado **UserComment**, permite guardar datos de texto. En este caso en concreto se guarda una estructura de datos en formato JSON [154] con todo lo que pueda ser de utilidad al procesar la imagen. Por ejemplo:

```
{
  "simulation": {
    "tile_side": 0.05,
    "led_panel_size": 0.173,
    "led_panel_height": 1
  },
  "camera": {
    "location": "front",
    "resolution_width": 3880,
    "resolution_height": 5184,
    "sensor_width": 3.492,
    "sensor_height": 4.6656,
    "focal_length": 3.52,
    "pixel_size": 0.9
  },
  "location": {
    "tile_x": -5,
    "tile_y": -8,
    "x": -0.25,
```

```

    "y": -0.4
  }
}

```

A partir de esta estructura de datos se puede saber todo lo necesario sobre la manera en la que la muestra fue generada. Guardar estos datos en formato JSON dentro de la estructura Exif de un archivo JPEG es eficaz pero resulta poco conveniente para saber de un vistazo cómo se generó la muestra. Para solucionar este problema se utilizan otros campos Exif. En la figura 4.25 se pueden ver algunos detalles sobre la manera en la que una muestra se generó sin necesidad de utilizar ningún lenguaje de programación.

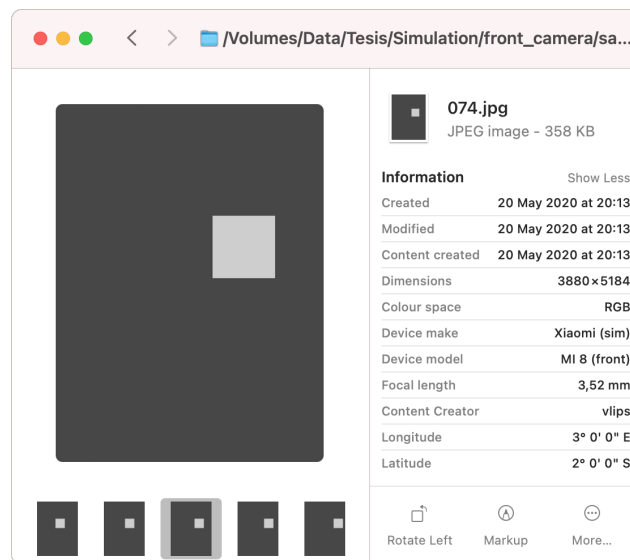


Figura 4.25. Metadatos contenidos en cada muestra.

Los campos generados por el script de Python de toma de muestras son:

- **Device make:** nombre del fabricante del dispositivo portable cuyas especificaciones se utilizaron en el proceso de simulación. En este caso se trata de un Xiaomi. El añadido “(sim)” es para diferenciarlo de una fotografía tomada, en realidad, por un dispositivo portable de esa marca.
- **Device model:** nombre del modelo del dispositivo portable cuyas especificaciones se utilizaron en el proceso de simulación. En este caso se trata de un Mi 8. El añadido “(front)” indica que fue tomado con la cámara frontal. Si se hubiese tomado con la trasera se habría añadido “(back)”.
- **Focal length:** distancia focal de la cámara simulada para tomar las muestras.
- **Content Creator:** nombre del software utilizado para generar las fotos. En este caso es *vlips*, por *Visible Light Indoor Positioning System*.

- **Longitude:** coordenada X desde la que se tomo la muestra. Los grados indican el valor absoluto de la coordenada X. Si su valor es positivo le acompañará **E**, si es negativo **W**.
- **Latitude:** coordenada Y desde la que se tomo la muestra. Los grados indican el valor absoluto de la coordenada Y. Si su valor es positivo le acompañará **N**, si es negativo **S**.

La cámara se moverá a intervalos regulares para tomar las muestras. El sistema utilizado para obtener el rango de posiciones por las que la cámara pasará está detallado en la sección 5.4. Por ejemplo, utilizando la cámara frontal se obtendrán 187 muestras, con la cámara moviéndose desde los $-0,25$ m hasta los $0,25$ m en el eje X, y desde los $-0,40$ m hasta los $0,40$ m en el eje Y. De esta manera, el panel **LED** siempre está completamente visible en todas las muestras que se tomen, tanto con la cámara frontal como con la trasera, obteniéndose un total de 187 muestras. Para conseguir estos objetivos se ha desarrollado el add-on que se describe a continuación.

4.8.1. Add-on de Blender

Los *add-ons* (o extensiones) de Blender [155] permiten gestionar *scripts* secundarios que extienden su funcionalidad. Estos add-ons se pueden instalar, activar y desactivar desde la interfaz de usuario de Blender. Y, lo más interesante en lo que a este trabajo respecta, se pueden desarrollar utilizando Python. Un add-on puede estar compuesto por tantos *scripts* como sea necesario, siempre que se sigan ciertas reglas. En la figura 4.26 se puede ver parte de la jerarquía de archivos contenidos dentro del add-on desarrollado como simulador.

4.8.2. Instalación

Una vez desarrollada la funcionalidad del add-on (de la que se habla en la sección 4.8.3), resulta muy sencillo instalarlo en cualquier copia de Blender, independientemente del sistema operativo utilizado. El primer paso consiste en ejecutar `package.sh` en la misma carpeta en la que estén los *scripts* de Python, para generar el archivo ZIP del add-on. Para instalarlo, sólo es necesario abrir Blender, seleccionar **Edit > Preferences > Add-ons**, hacer clic en el botón **Install**, y localizar el archivo ZIP creado anteriormente. Marcando la casilla de verificación que hay delante del nombre del add-on éste se activa para poder utilizarlo a través de la interfaz de usuario de Blender.

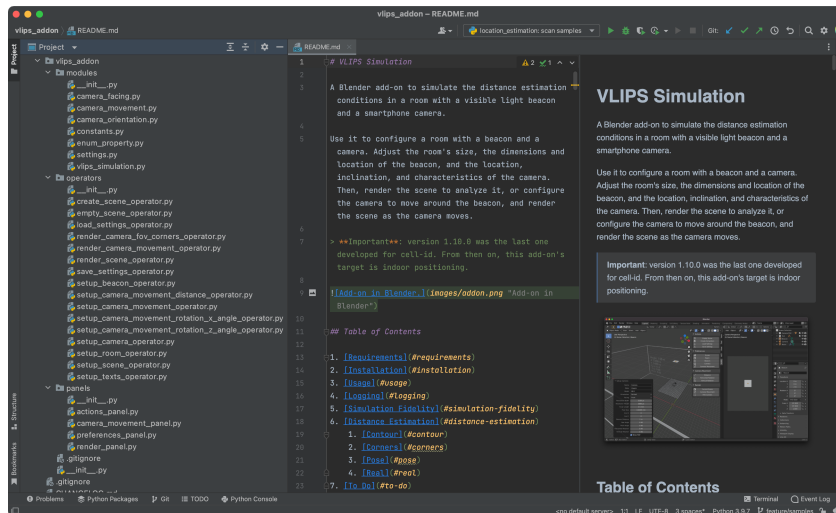


Figura 4.26. Add-on de Blender en PyCharm.

4.8.3. Utilización

Este add-on se ofrece al usuario de dos maneras: mediante operadores y mediante paneles. Así, mediante los paneles que se describen a continuación, el usuario puede cambiar las dimensiones de la habitación en la que la baliza y la cámara se encuentran, las características físicas de la cámara y la baliza, o la posición de ambos elementos. Una vez situados ambos elementos en la ubicación deseada, es posible tomar una muestra que posteriormente se podrá analizar. También se pueden definir una serie de movimientos que la cámara describirá con relación a la baliza, y tomar muestras en las diferentes posiciones en las que la cámara va parando, lo que facilita el análisis sistemático de los diferentes algoritmos de posicionamiento que se vayan desarrollando.

Las medidas de longitud se exponen al usuario en metros a través de los operadores y los paneles, mientras que los ángulos lo hacen en grados. Internamente, sin embargo, el add-on utiliza milímetros para que todas las operaciones se realicen utilizando valores enteros en la medida de lo posible.

Para acceder a los operadores se pulsa **F3** y se escribe `vlips_` en el cuadro de texto. Blender mostrará todos los operadores disponibles a través del add-on, como puede verse en la figura 4.27.

La lista de operadores que este add-on ofrece son:

- **Create Scene:** añade una habitación, una baliza y una cámara a la escena.

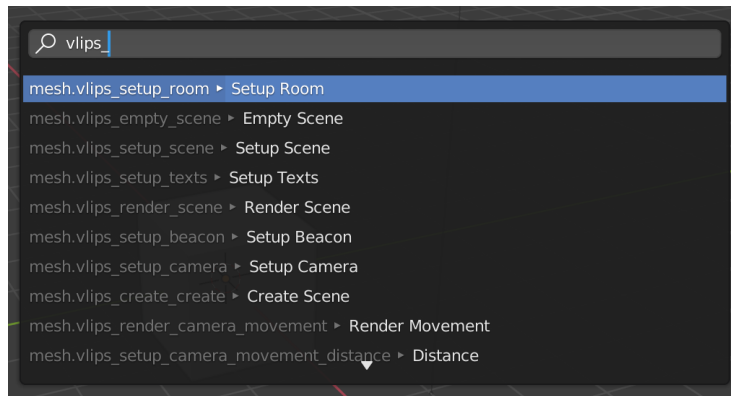


Figura 4.27. Operadores de Blender.

- **Empty Scene:** elimina todos los elementos de la escena para poder empezar desde cero.
- **Render Movement:** captura una serie de muestras en las que la ubicación de la cámara cambia con el tiempo. Puede variar la distancia relativa a la baliza, o su ángulo.
- **Render Scene:** captura la escena actual desde el punto de vista de la cámara, guardando el resultado en un archivo cuya ubicación selecciona el usuario.
- **Render FOV Corners:** para cada FOV calculado, mueve la cámara a cada una de sus esquinas y captura la escena actual desde el punto de vista de la cámara, guardando todas las imágenes resultantes para su posterior consulta.
- **Setup Beacon:** añade una baliza a la habitación.
- **Setup Camera:** añade una cámara a la habitación.
- **Setup Camera Movement:** define los valores necesarios para que la cámara describa un movimiento y dónde se deben guardar las muestras tomadas.
- **Distance:** define un movimiento de cámara en el que su distancia con relación a la baliza cambia.
- **Horizontal Rotation:** define un movimiento de cámara en el que su ángulo de rotación horizontal con relación a la baliza cambia.
- **Vertical Rotation:** define un movimiento de cámara en el que su ángulo de rotación vertical con relación a la baliza cambia.
- **Setup Room:** añade una habitación a la escena.
- **Setup Scene:** cambia la configuración de la escena con valores predeterminados adecuados para la simulación.
- **Setup Texts:** añade textos a la pared de la habitación que muestran detalles sobre la simulación.

Al activar cada uno de estos operadores se abre un cuadro de diálogo en la esquina inferior izquierda de la ventana de Blender, mostrando diferentes parámetros

configurables. Dejando el puntero del ratón sobre cualquiera de ellos aparece una descripción del mismo. Si se hace clic y se arrastra sobre su valor numérico éste cambia y se ven resultados en tiempo real.

Para acceder a los paneles se pulsa **n** mientras el puerto de visualización está activo. Se verán una serie de fichas en el borde derecho del puerto de visualización. Por defecto, estas fichas son: **Item**, **Tool** y **View**. Si el add-on está activo, se verá una nueva ficha al final llamada **VLIPS**. Si se hace clic sobre ella se verá algo parecido a lo que se muestra en la figura 4.28.

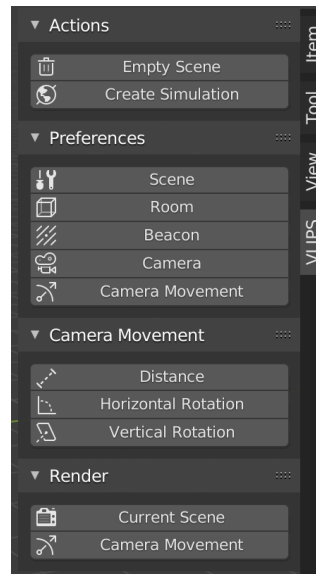


Figura 4.28. Paneles de Blender.

La primera sección, llamada **Actions**, contiene dos botones:

- **Empty Scene**: ejecuta el operador **Empty Scene**.
- **Create Simulation**: ejecuta el operador **Create Simulation**.

Estos dos botones son útiles accesos directos a los operadores más frecuentemente utilizados. Normalmente, lo primero que se hace al utilizar este add-on es vaciar la escena para entonces añadir los elementos de la simulación a la misma.

La segunda sección, llamada **Preferences**, contiene cinco botones:

- **Scene**: ejecuta el operador **Setup Scene**.
- **Room**: ejecuta el operador **Setup Room**.
- **Beacon**: ejecuta el operador **Setup Beacon**.
- **Camera**: ejecuta el operador **Setup Camera**.
- **Camera Movement**: ejecuta el operador **Setup Camera Movement**.

De nuevo, estos botones son accesos directos a los operadores. Por lo general, sólo se necesitará hacer clic en **Camera Movement** para seleccionar la carpeta en la que se guardarán las muestras que se tomen.

La tercera sección, llamada **Camera Movement**, contiene tres botones:

- **Distance:** ejecuta el operador **Beacon Distance**.
- **Horizontal Rotation:** ejecuta el operador **Horizontal Rotation**.
- **Vertical Rotation:** ejecuta el operador **Vertical Rotation**.

Estos botones se utilizan para configurar el movimiento que la cámara describirá con relación a la baliza. En cada caso, la baliza siempre permanecerá centrada con respecto al punto de vista de la cámara.

La cuarta sección, llamada **Render**, contiene tres botones:

- **Current Scene:** ejecuta el operador **Render Scene**.
- **Camera Movement:** ejecuta el operador **Render Camera Movement**.
- **FOV Corners:** ejecuta el operador **Render FOV Corners**.

Las imágenes obtenidas al tomar una muestra de la escena contiene todos los detalles necesarios para recrearla, guardados como datos Exif. Dichos datos pueden extraerse utilizando multitud de herramientas como, por ejemplo, [exiftool \[156\]](#).

El simulador desarrollado permite guardar la configuración en cualquier momento, para poder reutilizarla posteriormente sin necesidad de acceder a cada uno de los paneles, asignando los valores necesarios de uno en uno. Esto no sólo acelera la configuración del sistema y evita posibles errores en la introducción de los valores necesarios, sino que también permite disponer de una trazabilidad completa del proceso, algo muy importante para garantizar la replicabilidad de los experimentos. A continuación puede verse la parte del archivo de configuración que describe, en formato YAML no es un lenguaje de marcado (*YAML Ain't Markup Language*, [YAML](#)) [\[157\]](#), los elementos de una simulación:

```
version: 2.2.0
date: '2022-03-16 10:59:09'
scene:
  tile_side: 50.0
  floor_side_tiles: 32
room:
  name: Room
  width: 6000.0
```

```
depth: 4000.0
height: 2000.0
thickness: 50.0
beacon:
  name: Beacon
  width: 174.0
  height: 174.0
camera:
  name: Camera
  make: Xiaomi
  model: MI 8
  orientation: portrait
  facing: front
  resolution_width: 5184
  resolution_height: 3880
  focal_length: 3.52
  pixel_size: 0.0009
  grid_x: 0
  grid_y: 0
  beacon_distance: 1000.0
  rotation_x_angle: 0.0
  rotation_z_angle: 0.0
  show_fov: true
```

Los ajustes del movimiento de cámara y de las rotaciones también se guardan en el archivo de configuración en formato YAML. Dicho archivo se puede guardar manualmente utilizando el botón **Save Settings** del panel de control de la simulación. Sin embargo, para facilitar el proceso, cada vez que se toma un conjunto de muestras el simulador guarda con ellas el archivo de configuración correspondiente y una captura de pantalla en la que se puede ver el aspecto del simulador en el momento de tomarlas.

4.8.4. Fidelidad de la simulación

Antes de dar por bueno el entorno de simulación, es necesario comprobar si se ajusta a la realidad. Para ello compararemos imágenes reales con sus contrapartidas dentro de la simulación. En este caso, las imágenes reales son muestras tomadas de una baliza cuadrada con esquinas redondeadas. El contorno del cuadrado que contiene

la parte luminosa de la misma tiene 17,3 cm de lado. Tanto en el mundo real como en la simulación, cámara y baliza estaban ambas situadas a la misma altura, con sus superficies paralelas entre sí. En el mundo real, se utilizó la cámara trasera de un smartphone Xiaomi Mi 8 [122], en orientación vertical, cuyas características ya se han descrito en la página 139.

A simple vista, las muestras tomadas en el mundo real pueden compararse con las generadas en la simulación a diferentes distancias. En la figura 4.29a ambas estaban a 0,3 m, en la figura 4.29b a 1 m, en la figura 4.29c a 2 m y, por último, en la figura 4.29d estaban a 3 m.

Sin embargo, la mera comparación visual de las parejas de muestras tomadas en el mundo real y las generadas en la simulación no suponen una medida adecuada de la fidelidad de la simulación. Para cuantificarla se ha creado un script de Python llamado **fidelity_evaluation.py** que toma como entradas dos carpetas con las parejas de imágenes y devuelve como resultado un archivo CSV. El proceso de evaluación consiste en:

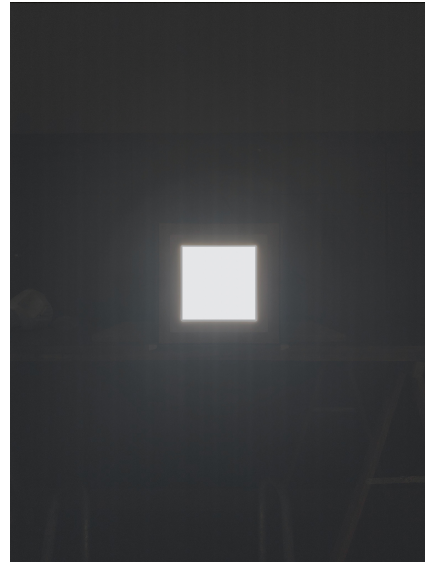
1. tomar la imagen del mundo real.
2. convertirla a blanco y negro con una operación de umbralización para aislar la parte luminosa de la baliza.
3. localizar su contorno.
4. tomar el segmento de la parte superior del contorno.
5. medir su longitud en píxeles.
6. repetir las mismas operaciones para la imagen generada en la simulación.
7. comparar ambos valores para obtener la diferencia y, a partir de ella, determinar el error cometido en la simulación con respecto al mundo real.

Las entradas de este script deben cumplir las siguientes condiciones:

- el argumento obligatorio `--samples` (o `-s`) debe ser una carpeta en la que estén las muestras tomadas en el mundo real.
- el argumento obligatorio `--renders` (o `-r`) debe ser una carpeta en la que estén las imágenes generadas en la simulación.
- el número de imágenes reales y simuladas debe coincidir.
- el ancho y el alto de cada una de las imágenes debe ser el mismo.
- los nombres de las muestras deben seguir el patrón `sample_XXX.jpg`, siendo XXX la distancia en centímetros entre la baliza y la cámara cuando la muestra fue tomada, rellenada con ceros para que el tamaño de esa parte del nombre del archivo sea el mismo en cada caso.



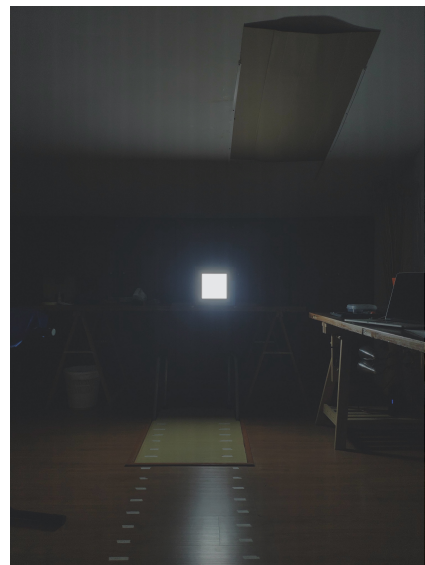
(a) 30 cm.



(b) 1 m.



(c) 2 m.



(d) 3 m.

Figura 4.29. Mundo real y simulación.

- los nombres de las imágenes generadas en la simulación deben seguir el patrón `render_XXX.jpg`, siendo `XXX` la distancia en centímetros entre la baliza y la cámara cuando la imagen se generó, de nuevo rellena con ceros para igualar el tamaño de esa parte del nombre del archivo.
- la distancia entre cada imagen debe ser la misma en cada caso.
- para cada muestra del mundo real debe existir un equivalente de la simulación, de manera que la distancias coincidan.
- es aconsejable colocar ambas carpetas de imágenes en la misma ubicación, dentro de la misma carpeta, ya que el archivo CSV resultante se guardará allí.

Para evaluar la fidelidad de la simulación dados ambos conjuntos de imágenes se debe ejecutar esta orden:

```
python3 fidelity_evaluation.py --samples 01-samples --renders 02-renders
```

También existe la opción de utilizar el perfil de ejecución **fidelity_evaluation** dentro de PyCharm con los mismos argumentos.

El argumento opcional `--debug` (o `-d`) hace que se guarden datos extra en la carpeta **debug**, dentro de cada carpeta de muestras e imágenes de la simulación. Para cada muestra se generarán dos imágenes. La primera es una representación en blanco y negro de la imagen en la que únicamente se observa la parte luminosa de la misma, que coincide con la superficie de la baliza. La segunda muestra el contorno de la baliza, con un círculo azul marcando el inicio del segmento que se utilizará para medir su lado y un círculo rojo marcando el final del mismo, tanto en la muestra tomada en el mundo real como en la generada en la simulación. En la figura 4.30 se muestra un ejemplo de esta última imagen.

El archivo CSV resultante se guardará con el nombre:

```
fidelity_evaluation_result_YYYY-MM-DD_hh-mm-ss.csv
```

Estará dentro de la carpeta que contenga las dos series de imágenes a comparar, donde `YYYY-MM-DD_hh-mm-ss` es la marca de tiempo del momento en el que se ejecutó el script de Python que produjo ese archivo. Las columnas contenidas en el CSV son las siguientes:

- **distance**: distancia a la que se tomó la muestra y se generó la imagen correspondiente dentro de la simulación.
- **sample_side**: longitud en píxeles del segmento a comparar de la muestra.

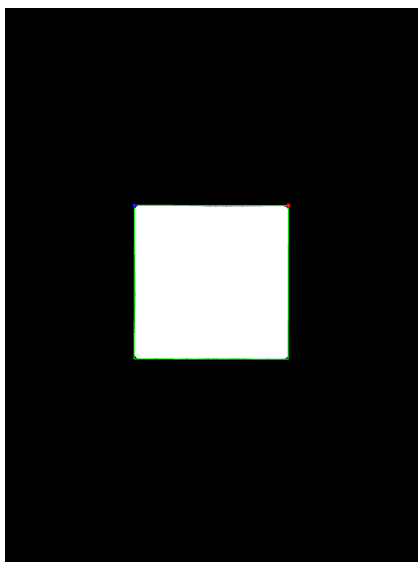


Figura 4.30. Imagen de depuración.

- **render_side**: longitud en píxeles del segmento a comparar de la imagen generada en la simulación.
- **side_difference**: diferencia entre ambos segmentos.
- **error**: porcentaje de error cometido.
- **normalized_error**: error dividido entre la distancia, normalizado de manera que el mayor de los errores representa el límite superior del mismo (100 %).

En la figura 4.31 se puede ver un gráfico generado a partir de los valores del CSV anterior.

La media de la proporción entre las muestras y la simulación es 0,9638. El máximo de la proporción, 0,9733, ocurre a 2,4 m, mientras que el mínimo, 0,9492, tiene lugar a 1 m de distancia. La desviación estándar de la proporción es 0,0047.

El 85 % de las proporciones están dentro de dos desviaciones estándar de la media. Teniendo en cuenta que las balizas van a estar situadas en el techo a una distancia mínima de 1 m y que todas las proporciones de fidelidad están por encima de 0,95, se puede concluir que los resultados obtenidos al evaluar la fidelidad de la simulación son aceptables.

4.8.5. Otras características

Para facilitar el desarrollo del add-on se ha creado otro add-on que acelera todo el proceso, gracias al que se ahorra mucho tiempo y esfuerzo en una tarea altamente

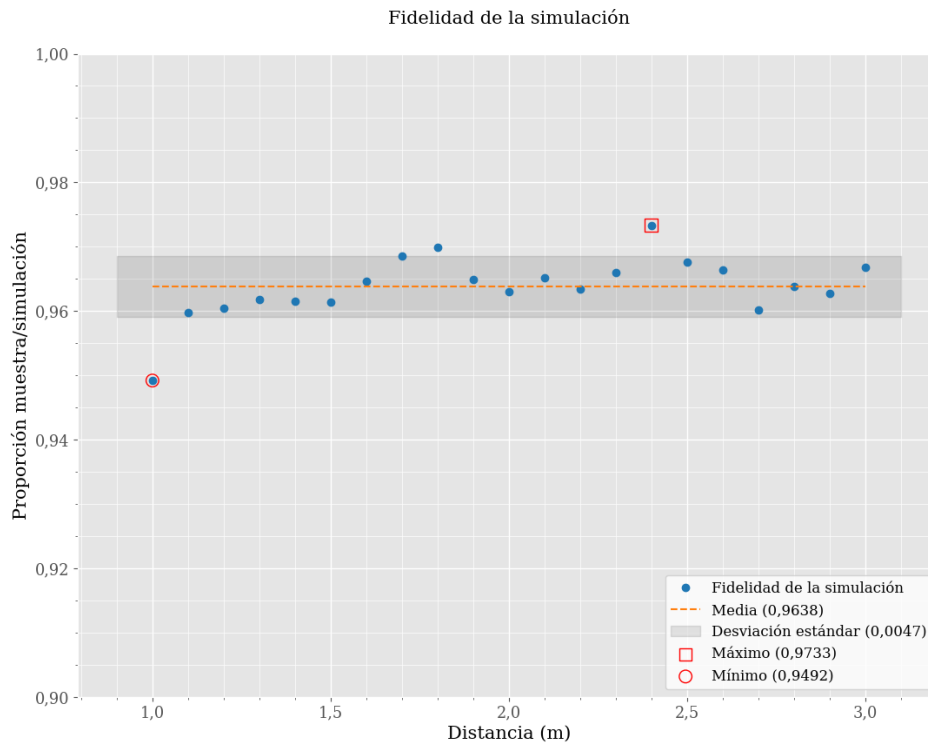


Figura 4.31. Fidelidad de la simulación.

repetitiva. Su nombre es ACP, de *Add-on Control Panel* (panel de control de add-on). Dicho add-on incluye dos botones en la interfaz de usuario:

- **Install Add-on:** empaqueta el código fuente del add-on en un archivo ZIP, instala el add-on y lo activa.
- **Uninstall Add-on:** desinstala el add-on.

Al activar este add-on ofrece unas preferencias que permiten indicar el nombre del módulo y su ubicación en el sistema de archivos, algo que añade flexibilidad y permite trabajar con cualquier otro add-on que se pueda necesitar en el futuro. Tanto las preferencias como los botones que añade a la interfaz de usuario pueden verse en la figura 4.32. El código fuente de este add-on puede consultarse en el repositorio de GitLab correspondiente [158], dentro del grupo de proyectos dedicados a este trabajo [132].

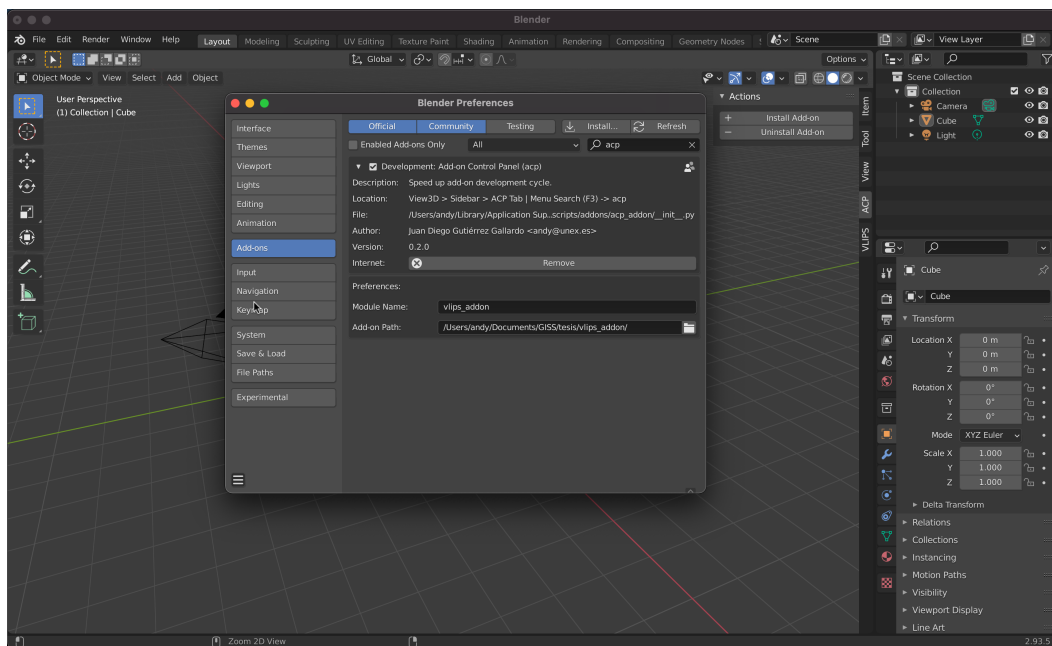


Figura 4.32. Add-on Control Panel.

Posicionamiento

5.1. Introducción

A continuación se describe un sistema de posicionamiento basado en luz visible en el que el emisor es un panel **LED** comercial y el receptor es la cámara de un dispositivo portable sobre el que se realizará todo el procesamiento. El funcionamiento del sistema se basa en el modelo óptico de la cámara estenopeica y en el análisis de los datos obtenidos tras realizar un procesamiento de las muestras obtenidas. Ambos elementos serán presentados a continuación en este capítulo, antes de describir el sistema completo.

Además, se describe el sistema de posicionamiento propiamente dicho, así como algunos aspectos relevantes del simulador utilizado para probar su funcionamiento.

5.2. Conceptos previos

5.2.1. Modelo de cámara estenopeica

Las cámaras de los dispositivos portables actuales se basan el modelo de cámara estenopeica [159]. Una versión simplificada en dos dimensiones de este modelo puede verse en la figura 5.1, donde a es el ancho (o el alto) del sensor de la cámara, f su distancia focal, D la distancia de la cámara al objeto que se está fotografiando y A el ancho (o el alto) del objeto que está siendo fotografiado. La diferencia más significativa consiste en la presencia de una lente situada entre el sensor óptico y la luz que incide sobre él, haciéndola converger justo en el punto en el que debería estar el orificio o *pinhole*, optimizando así la calidad de la señal adquirida [160].

En base a este modelo representado en la figura 5.1, \mathbf{a} es una matriz 2×1 que representa las dimensiones del sensor de la cámara (ancho y alto). Así mismo, \mathbf{A} es una matriz 2×1 que representa las dimensiones del **FOV** de la cámara. Por último, D representa la distancia entre la cámara y la baliza en el sistema de posicionamiento

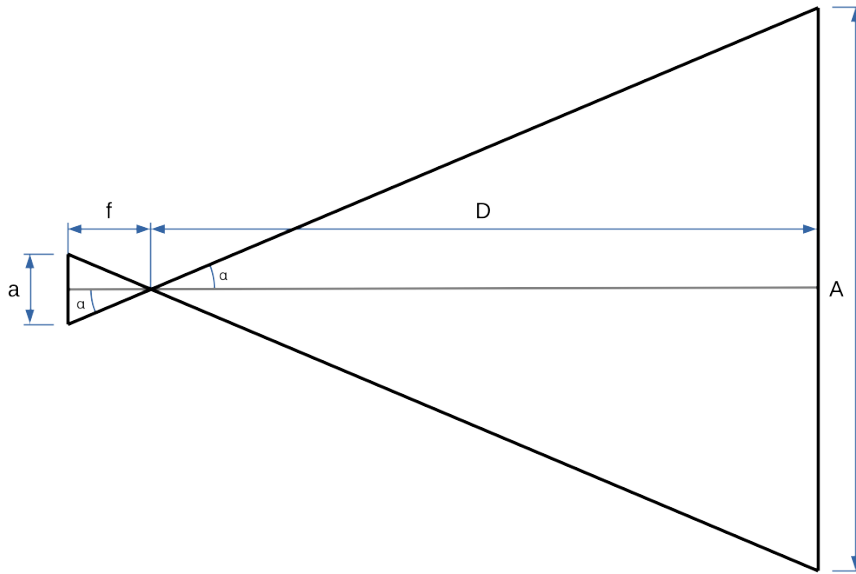


Figura 5.1. Modelo de cámara estenopeica.

en estudio y f es la distancia entre la lente y el sensor de la cámara, es decir, la distancia focal.

5.2.2. Geometría de la formación de imágenes

En esta sección se describe cómo se forman las imágenes en el sensor de la cámara según el modelo estenopeico y en qué base matemática se apoya [161], lo que será de utilidad para el desarrollo del sistema de posicionamiento.

Es necesario describir la situación en la que se encuentra cada elemento de interés. En este caso, se trata de una sala (con su propio sistema de coordenadas), una cámara (también con su propio sistema de coordenadas, diferente al de la sala) y uno de los puntos de una baliza presente en la sala que la cámara quiere capturar. Esta situación puede verse en la figura 5.2.

El sistema de coordenadas de la sala tiene su origen $(0, 0, 0)$ en la esquina donde confluyen los tres ejes X_s , Y_s y Z_s . Los ejes X_s e Y_s definen el plano formado por el suelo de la habitación, mientras que el eje Z_s se encuentra a lo largo de la esquina conformada por las paredes donde se ubica el origen de coordenadas. Utilizando este sistema de referencia resulta posible situar en tres dimensiones cualquier punto dentro de la sala, midiendo su distancia desde el origen a lo largo de los tres ejes.

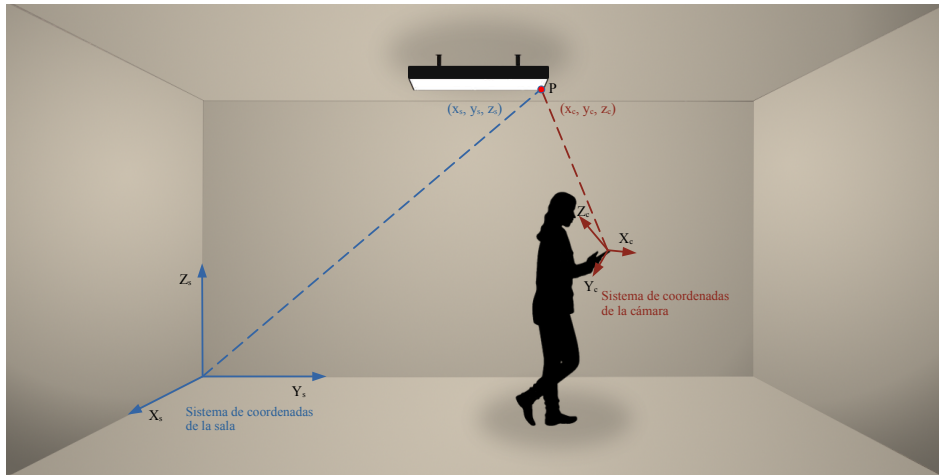


Figura 5.2. Sistemas de coordenadas presentes al tomar una muestra.

En este sistema de coordenadas, el punto P de la baliza tendría las coordenadas (x_s, y_s, z_s) .

Por otra parte, es posible crear otro sistema de coordenadas para el dispositivo portable en el que el origen esté en la cámara frontal del mismo. En este caso, los ejes X_c e Y_c podrían estar en el plano del dispositivo: el primero tomaría valores positivos hacia la derecha, mientras que el segundo lo haría hacia abajo. El eje Z_c , por otro lado, sería perpendicular al plano del dispositivo, tomando valores positivos hacia el rostro del usuario. De igual forma que en el sistema de coordenadas anterior, es posible situar en tres dimensiones cualquier punto dentro del campo de visión de la cámara, midiendo su distancia a lo largo de los tres ejes definidos. En el sistema de coordenadas de la cámara, el mismo punto P de la baliza tendría las coordenadas (x_c, y_c, z_c) .

El punto P puede definirse de diferentes formas, dependiendo del sistema de coordenadas utilizado. Sin embargo, es posible cambiar de sistema de coordenadas. Para ello, basta con rotar el punto de partida y, a continuación, realizar una operación de traslación:

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \mathbf{R} \begin{bmatrix} x_s \\ y_s \\ z_s \end{bmatrix} + \mathbf{t} \quad (5.1)$$

donde \mathbf{R} es una matriz de rotación de 3×3 elementos y \mathbf{t} es un vector de traslación de 3 elementos. Es posible simplificar la ecuación (5.1):

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = [\mathbf{R}|\mathbf{t}] \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix} \quad (5.2)$$

donde la matriz de rotación \mathbf{R} y el vector de traslación \mathbf{t} se han concatenado para formar una matriz de 3×4 y la posición del punto P en la sala se ha expresado utilizando coordenadas homogéneas [89]. En este tipo de coordenadas, se añade una cuarta coordenada, W , que permite indicar la distancia entre la cámara y el objeto. En este caso sólo se utiliza para simplificar la notación y su valor será 1. La concatenación entre la matriz de rotación y el vector de traslación se conoce como matriz extrínseca:

$$\mathbf{E} = [\mathbf{R}|\mathbf{t}] \quad (5.3)$$

Combinando la ecuación (5.2) y la ecuación (5.3) se obtiene:

$$\begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = \mathbf{E} \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix} \quad (5.4)$$

Utilizando la ecuación (5.4) es posible transformar las coordenadas de un punto de la baliza descrito según el sistema de referencia de la sala a otro descrito utilizando el sistema de referencia de la cámara. Sin embargo, en las muestras tomadas por la cámara, los puntos utilizan otro sistema de referencia diferente de los anteriores: el de las imágenes. En la figura 5.3 se puede ver cómo se forma una imagen al proyectarse la luz sobre el sensor de la cámara de un dispositivo portable, junto con los diferentes sistemas de referencia presentados anteriormente.

Mientras que en el sistema de referencia de la cámara el eje X toma valores positivos hacia la derecha del usuario (que está de espaldas al punto P) y el eje Y hacia abajo, en la proyección ambos ejes están reflejados debido a la manera en la que las imágenes se forman en el modelo de cámara estenopeica [162]. El centro óptico O_c se encuentra en el origen de coordenadas del sistema de referencia de la cámara. La distancia entre el plano en el que se forma la imagen y el centro óptico es la distancia focal f . Mediante relaciones trigonométricas es posible expresar las coordenadas

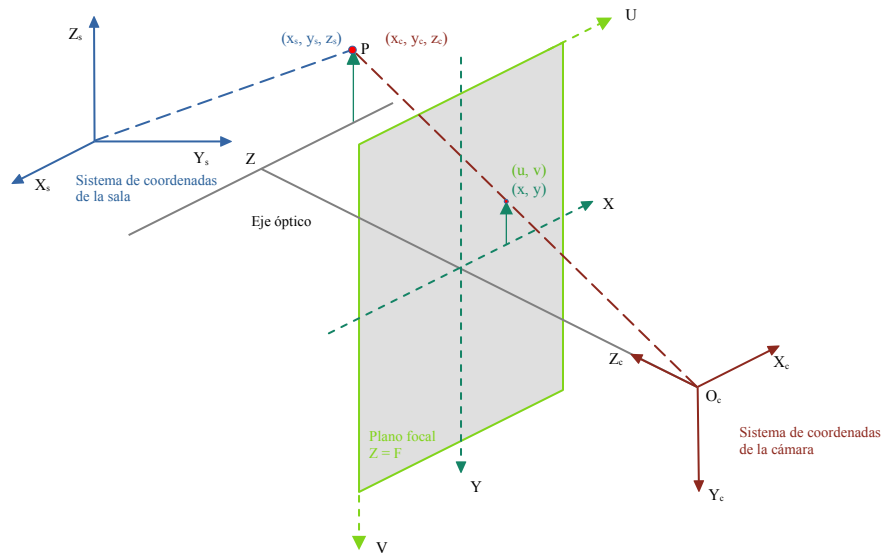


Figura 5.3. Proyección del punto P en el sensor de la cámara y sistemas de referencia [161].

x e y de la proyección del punto P sobre el plano de la imagen en función de las coordenadas de la cámara y la distancia focal:

$$x = f \frac{x_c}{z_c} \quad ; \quad y = f \frac{y_c}{z_c} \quad (5.5)$$

Incluyendo z en la ecuación (5.5) y expresando ambas igualdades en notación matricial se obtiene el siguiente resultado:

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \quad (5.6)$$

donde $x = x'/z'$ e $y = y'/z'$.

El primer término a la derecha de la igualdad se conoce como la matriz intrínseca de la cámara I . Sin embargo, para que esté completa, es necesario modificar sus elementos. En primer lugar, si los píxeles de la imagen no fuesen cuadrados, la distancia focal tendría diferentes valores para la x y la y , con lo que se podría desglosar en dos componentes: f_x y f_y . No se trata de un hecho frecuente. Por otra parte, el centro óptico no tiene por qué coincidir con el origen de coordenadas de la imagen. De hecho, en la figura 5.3 el origen de coordenadas de la muestra obtenida está en la esquina superior izquierda de la misma (u, v) . Con estos datos, es posible reescribir la matriz intrínseca de la cámara así:

$$\mathbf{I} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (5.7)$$

De los resultados anteriores se deduce que:

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \mathbf{I} \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \quad (5.8)$$

donde $u = u'/w'$ y $v = v'/w'$. Las coordenadas (u, v) , cuyo origen se encuentra en la esquina superior izquierda de la imagen, se miden en píxeles. Sin embargo, para mantener las mismas unidades en todas las ecuaciones es necesario utilizar el sistema métrico decimal. Cada sensor utilizado en la cámara de un dispositivo portable tiene una característica, el tamaño del píxel, que suele estar expresado en μm . El tamaño del píxel es un valor proporcionado por el fabricante del sensor óptico integrado en el dispositivo portable. La distancia focal es un valor que depende de cómo combine el fabricante el sensor óptico y las lentes que lo complementan.

La combinación de la ecuación (5.4) y la ecuación (5.8) permite transformar las coordenadas de un punto según el sistema de referencia de una sala a las coordenadas de del sistema de referencia de la cámara de un dispositivo portable y, de ahí, a las coordenadas de la imagen capturada:

$$\begin{bmatrix} u' \\ v' \\ w' \end{bmatrix} = \mathbf{I} \times \mathbf{E} \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix} \quad (5.9)$$

5.2.3. Estimación de la pose

Según [163], la estimación de la pose es una tarea presente en el campo de la visión por ordenador desde sus comienzos, crucial para múltiples tareas como el reconocimiento de objetos o la organización de enjambres de robots, por citar dos ejemplos. Una definición formal de la estimación de la pose extraída de [164] sería:

Por pose nos referimos a la transformación necesaria para mapear un modelo de objeto a partir de su sistema de coordenadas inherente para que coincida con los datos sensoriales.

Una definición de la estimación de la pose más cercana a las intenciones de este trabajo se puede encontrar en [165]. Se llama estimación de la pose en el campo de la visión por ordenador a la estimación de la posición relativa de un objeto con relación a la cámara. Dicha posición relativa se describe mediante un vector de traslación con el que mover la cámara de su posición inicial en tres dimensiones a una nueva ubicación, y una matriz de rotación, para cambiar la orientación de la cámara. El vector de traslación tiene tres elementos. Aunque la matriz de rotación sea de 3×3 elementos, representa el movimiento de la cámara alrededor de los tres ejes del espacio tridimensional. Este movimiento se corresponde con los tres ángulos de Euler: alabeo, elevación y dirección (*roll*, *pitch* y *yaw* en inglés), que describen rotaciones alrededor de los ejes X, Y y Z. En consecuencia, estimar la pose consiste en averiguar seis números: tres para la traslación y tres para la rotación de la cámara.

Como se explica en la sección 5.2.2, hay tres sistemas de coordenadas en juego. En primer lugar, el sistema de coordenadas de la sala, en el que se encuentra la baliza que ayudará en el sistema de posicionamiento (en este caso, una luminaria ubicada en el techo). En segundo lugar, el sistema de coordenadas de la cámara, desde el que se toma la muestra utilizada para estimar la posición del dispositivo portable. Por último, el sistema de coordenadas de la imagen, en el que se proyecta el objeto de la sala utilizado como baliza. Conocida la ubicación del punto $P = (x_s, y_s, z_s)$ de la baliza dentro del sistema de coordenadas de la sala, y utilizando la ecuación (5.4):

$$\begin{bmatrix} x_c \\ y_c \\ z_c \\ 1 \end{bmatrix} = \mathbf{E} \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix} = \begin{bmatrix} r_{00} & r_{01} & r_{02} & tx \\ r_{10} & r_{11} & r_{12} & ty \\ r_{20} & r_{21} & r_{22} & tz \end{bmatrix} \begin{bmatrix} x_s \\ y_s \\ z_s \\ 1 \end{bmatrix} \quad (5.10)$$

Las coordenadas del punto P en la proyección también son conocidas gracias a la ecuación (5.8):

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = e\mathbf{I} \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} = e \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_c \\ y_c \\ z_c \end{bmatrix} \quad (5.11)$$

donde el factor e es un factor de escala que representa a la distancia a la que el objeto se encuentra del centro óptico de la cámara. Existen infinitos puntos conectados al centro óptico que, proyectados sobre el sensor, producirían el mismo resultado.

Combinando la ecuación (5.10) y la ecuación (5.11) no resulta posible determinar la posición de la cámara al tomar la muestra, ya que existen más incógnitas que valores conocidos. Por fortuna, el punto P no existe por sí mismo, aislado. Por una parte, en la muestra tomada por la cámara se pueden encontrar tanto ese punto como otros puntos de la baliza. Por otra parte, se conoce la geometría de la baliza a la que pertenece. Combinando todos los datos conocidos, es posible utilizar una transformación lineal directa (*Direct Linear Transformation, DLT*) para obtener la posición de la cámara [166]. Sin embargo, existe una mejora de esta solución conocida como el algoritmo Levenberg–Marquardt (*Levenberg–Marquardt Algorithm, ALM*), que permite obtener mejores resultados [167]. Este algoritmo (también conocido como el método de mínimos cuadrados amortiguados), se utiliza para resolver problemas de mínimos cuadrados no lineales. Se puede encontrar más información al respecto en [168] y [169]. El algoritmo itera buscando posibles soluciones, cambiando ligeramente los valores de la matriz de rotación y el vector traslación hasta encontrar un mínimo local que satisface los condiciones preestablecidas. La complejidad de los diferentes métodos conocidos para la resolución del problema de la estimación de la pose puede llegar a $O(n^5)$, incluso hasta $O(n^8)$ [170]. Sin embargo, existe un método presentado en [171] que permite rebajar esa complejidad hasta $O(n)$, manteniendo resultados consistentes a la vez que rebaja el tiempo necesario para obtenerlos: el algoritmo SQPnP.

5.2.4. El algoritmo SQPnP

Tal y como se describe en [171], el problema de estimación de la pose, también conocido como problema de perspectiva desde n puntos (*Perspective N-Points* o PnP para abreviar), consiste en la recuperación de los seis grados de libertad de la pose de la cámara, dados cierto número de puntos n en un espacio tridimensional y la proyección en el plano de la cámara. Para poder resolver este problema es necesario que $n \geq 3$, pues que de otra manera existirían ambigüedades que no permitirían obtener soluciones únicas.

Así, los puntos conocidos en el espacio tridimensional serían $M_i \in \mathbb{R}^3, i \in \{1, \dots, n\}$, mientras que sus proyecciones en el plano euclídeo $Z = 1$ serían m_i , con relación a un sistema de coordenadas, el de la cámara, desconocido. La rotación de la cámara dentro del espacio tridimensional está descrita en la matriz R , mientras que su

traslación lo está en el vector \mathbf{t} . El algoritmo que se describe en este apartado intenta minimizar el error al cuadrado acumulado de la proyección de los puntos:

$$\sum_{i=1}^n \left\| \mathbf{m}_i - \frac{\mathbf{R}\mathbf{M}_i + \mathbf{t}}{\mathbf{1}_z^T (\mathbf{R}\mathbf{M}_i + \mathbf{t})} \right\|^2 \quad (5.12)$$

donde $\mathbf{1}_z = [0 \ 0 \ 1]^T$ y $(\mathbf{R}\mathbf{M}_i + \mathbf{t}) / (\mathbf{1}_z^T (\mathbf{R}\mathbf{M}_i + \mathbf{t}))$ es la proyección de \mathbf{M}_i en el plano euclídeo $Z = 1$, según el sistema de referencia de la cámara.

A partir de la ecuación (5.12) se puede obtener la siguiente función de coste:

$$\mathcal{E}^2 = \sum_{i=1}^n \left\| \mathbf{1}_z^T (\mathbf{R}\mathbf{M}_i + \mathbf{t}) \mathbf{m}_i - (\mathbf{R}\mathbf{M}_i + \mathbf{t}) \right\|^2 \quad (5.13)$$

Sea $\mathbf{r} \in \mathbb{R}^9$ el vector formado al apilar las filas de \mathbf{R} . La inversa de esta operación se denota por $\text{mat}(\mathbf{r}) = \mathbf{R}$. Dado un punto \mathbf{M}_i , se define $\mathbf{A}_i \in \mathbb{R}^{3 \times 9}$ como la matriz:

$$\mathbf{A}_i = \begin{bmatrix} \mathbf{M}_i^T & \mathbf{0}_3^T & \mathbf{0}_3^T \\ \mathbf{0}_3^T & \mathbf{M}_i^T & \mathbf{0}_3^T \\ \mathbf{0}_3^T & \mathbf{0}_3^T & \mathbf{M}_i^T \end{bmatrix} \quad (5.14)$$

Así, $\mathbf{R}\mathbf{M}_i = \mathbf{A}_i \mathbf{r}$. Realizando esta sustitución en la ecuación (5.13) y aprovechando el hecho de que $\mathbf{1}_z^T (\mathbf{R}\mathbf{M}_i + \mathbf{t}) \mathbf{m}_i = \mathbf{m}_i \mathbf{1}_z^T (\mathbf{R}\mathbf{M}_i + \mathbf{t})$, la función de coste se puede transformar así:

$$\mathcal{E}^2 = \sum_{i=1}^n (\mathbf{A}_i \mathbf{r} + \mathbf{t})^T \mathbf{Q}_i (\mathbf{A}_i \mathbf{r} + \mathbf{t}) \quad (5.15)$$

donde \mathbf{Q}_i es una matriz semidefinida positiva asociada a la proyección normalizada euclidiana \mathbf{m}_i mediante la equivalencia $\mathbf{Q}_i = (\mathbf{m}_i \mathbf{1}_z^T - \mathbf{I}_3)^T (\mathbf{m}_i \mathbf{1}_z^T - \mathbf{I}_3)$.

Si se tienen en cuenta las condiciones necesarias de optimización de primer orden, es posible eliminar \mathbf{t} de la función de coste transformada la ecuación (5.15). Igualando a la cero la derivada de \mathcal{E}^2 con respecto a \mathbf{t} se obtiene que:

$$\sum_{i=1}^n \mathbf{Q}_i (\mathbf{A}_i \mathbf{r} + \mathbf{t}) = 0 \Leftrightarrow \left(\sum_{i=1}^n \mathbf{Q}_i \right) \mathbf{t} = - \left(\sum_{i=1}^n \mathbf{Q}_i \mathbf{A}_i \right) \mathbf{r} \quad (5.16)$$

Dado que la matriz $\sum_{i=1}^n \mathbf{Q}_i$ es invertible, el vector de traslación puede expresarse en función de la matriz de rotación:

$$\mathbf{t} = \mathbf{P}\mathbf{r} \quad (5.17)$$

donde:

$$\mathbf{P} = - \left(\sum_{i=1}^n \mathbf{Q}_i \right)^{-1} \left(\sum_{i=1}^n \mathbf{Q}_i \mathbf{A}_i \right) \quad (5.18)$$

Si se sustituye la ecuación (5.17) en la ecuación (5.15) se obtiene una expresión del error cuadrático al cuadrado únicamente en función del vector de rotación \mathbf{r} :

$$\mathcal{E}^2 = \mathbf{r}^T \boldsymbol{\Omega} \mathbf{r} \quad (5.19)$$

donde $\boldsymbol{\Omega}$ es la matriz semidefinida positiva de 9×9 :

$$\boldsymbol{\Omega} = \sum_{i=1}^n (\mathbf{A}_i + \mathbf{P})^T \mathbf{Q}_i (\mathbf{A}_i + \mathbf{P}) \quad (5.20)$$

Es ahora cuando se puede enunciar el problema PnP como un programa cuadrático no lineal (*Non-Linear Quadratic Program*, **NLQP**) sobre un vector desconocido $\mathbf{x} \in \mathbb{R}^9$, partiendo de la base de que existen valores de \mathbf{x} que no se corresponden con matrices de rotación válidas. Dicho **NLQP** es:

$$\underset{\mathbf{x} \in \mathbb{R}^9}{\text{minimizar}} \mathbf{x}^T \boldsymbol{\Omega} \mathbf{x}, \mathbf{h}(\mathbf{x}) = \mathbf{0}_6 \quad (5.21)$$

donde $\boldsymbol{\Omega}$ se define en la ecuación (5.20) y $\mathbf{h}(\mathbf{x}) \in \mathbb{R}$ es un vector de restricción para asegurar que cuando $\mathbf{h}(\mathbf{x}) = \mathbf{0}_6$ el vector \mathbf{x} representa una matriz de rotación:

$$\mathbf{h}(\mathbf{x}) = \begin{bmatrix} \mathbf{x}_{1:3}^T \mathbf{x}_{1:3} - 1 \\ \mathbf{x}_{4:6}^T \mathbf{x}_{4:6} - 1 \\ \mathbf{x}_{1:3}^T \mathbf{x}_{4:6} \\ \mathbf{x}_{1:3}^T \mathbf{x}_{7:9} \\ \mathbf{x}_{4:6}^T \mathbf{x}_{7:9} \\ \det(\text{mat}(\mathbf{x})) - 1 \end{bmatrix} \quad (5.22)$$

La solución a este problema se encuentra en una hipersfera de radio $\sqrt{3}$ con centro en el origen de \mathbb{R}^9 , llamada \mathbb{S}^8 de ahora en adelante. Con estas premisas, el problema se puede redefinir así:

$$f(\mathbf{x}) = \mathbf{x}^T \Omega \mathbf{x}, \mathbf{x} \in \mathbb{S}^8 \quad (5.23)$$

Se sabe que los puntos estacionarios de una función cuadrática restringida por norma unitaria son vectores propios de Ω [172], $\mathbf{e}_1, \dots, \mathbf{e}_9$ en adelante. Por lo tanto, la función f de la ecuación (5.23) tiene 18 puntos estacionarios diferentes $\mathbf{x}_1, \dots, \mathbf{x}_{18} \in \mathbb{S}^8$ que se corresponden con los 9 vectores propios de Ω multiplicados por el factor $\pm\sqrt{3}$:

$$\mathbf{x}_1 = +\sqrt{3}\mathbf{e}_1, \dots, \mathbf{x}_9 = +\sqrt{3}\mathbf{e}_9, \mathbf{x}_{10} = -\sqrt{3}\mathbf{e}_1, \dots, \mathbf{x}_{18} = -\sqrt{3}\mathbf{e}_9 \quad (5.24)$$

Como la función de coste la ecuación (5.21) es cuadrática, se puede expresar en términos de la diferencia $\delta = \mathbf{x} - \mathbf{r}$ utilizando su primera aproximación de Taylor:

$$\underset{\delta \in \mathbb{R}^9}{\text{minimizar}} \delta^T \Omega \delta + 2\mathbf{r}^T \Omega \delta, \mathbf{H}_r \delta = -\mathbf{h}(\mathbf{r}) \quad (5.25)$$

Las condiciones de primer orden de la función lagrangiana junto con las restricciones lineales de la ecuación ?? conforman un sistema lineal cuya solución es una dirección descendente que converge o se mantiene en la trayectoria de las posibles soluciones. En la figura 5.4 se puede ver un ejemplo gráfico de la obtención de estos mínimos locales que constituyen soluciones al problema.

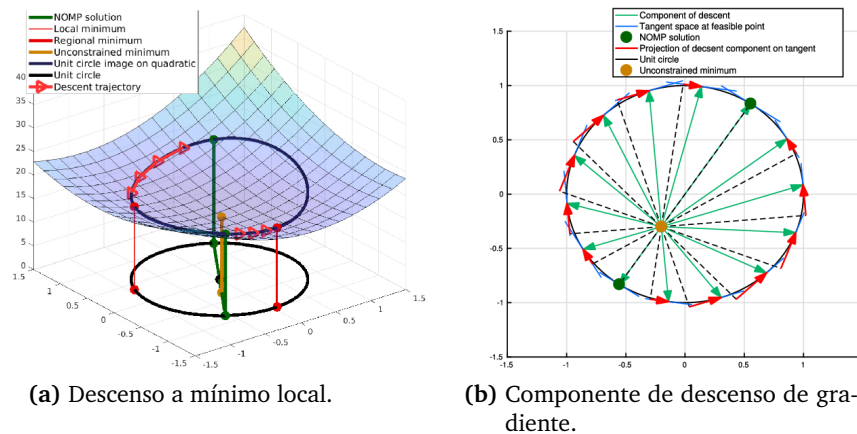


Figura 5.4. Búsqueda de un mínimo local a partir de las soluciones del problema de la matriz ortogonal más próxima para el caso en el que $\mathbb{R} \in \mathcal{SO}(2)$ [171].

Existen dos mínimos locales, cada uno de ellos asociado a un vector propio multiplicado por $\pm\sqrt{3}$:

$$\mathbf{r}_1 = \underset{\text{mat}(\mathbf{x}) \in \mathcal{SO}(3)}{\text{argmin}} \left\| \mathbf{x} - \sqrt{3}\mathbf{e} \right\|^2, \mathbf{r}_1 = \underset{\text{mat}(\mathbf{x}) \in \mathcal{SO}(3)}{\text{argmin}} \left\| \mathbf{x} + \sqrt{3}\mathbf{e} \right\|^2 \quad (5.26)$$

Generalizando el caso anterior, se obtiene:

$$\mathbf{r}_i = \underset{\text{mat}(\mathbf{x}) \in \mathcal{SO}(3)}{\text{argmin}} \left\| \mathbf{x} - (-1)^{\lfloor (i-1)/k \rfloor} \sqrt{3}\mathbf{e}_{9-k+i-\lfloor i/k \rfloor k} \right\|^2 \quad (5.27)$$

El proceso necesario para aplicar la ecuación (5.27) conforma el algoritmo SQPnP. Este algoritmo, como se explica en [171], permite obtener los mínimos globales del problema PnP y está dividido de dos partes. La primera parte, en la tabla 5.1, resuelve el problema PnP devolviendo una lista de minimizadores que contiene el mínimo global. El algoritmo calcula la matriz Ω , la matriz P que necesita la ecuación (5.17) y los puntos de comienzo de la búsqueda de las soluciones.

Tabla 5.1. Primera parte del algoritmo SQPnP [171].

Algoritmo 1. SQPnP: SolvePnP.

Requisitos:

- Número de puntos: $n \geq 3$
- Puntos del mundo: $\mathbf{M}_i \in \mathbb{R}^3, 1 \leq i \leq n$
- Proyecciones: $\mathbf{m}_i = [x_i \ y_i \ 1]^T, 1 \leq i \leq n$
- Tolerancia a las perturbaciones: $\epsilon \leq 10^{-5}$
- Número máximo de iteraciones: $T \geq 15$

$\{\Omega, P\} \leftarrow$ de las ecuaciones (5.18), (5.20)

$\{\mathbf{e}_1, \dots, \mathbf{e}_9, s_1, \dots, s_9\} \leftarrow \text{SVD}(\Omega)$

$\{\mathbf{e}_{10-k}, \dots, \mathbf{e}_9\} \leftarrow \underset{\mathbf{x} \in \mathbb{S}^8}{\text{argmin}} \mathbf{x}^T \Omega \mathbf{x}$

desde $i \leftarrow 1$ hasta $2k$ hacer

$\mu \leftarrow \lfloor (i-1)/k \rfloor$

$\nu \leftarrow 9 - k + i - \lfloor i/k \rfloor k$

$\mathbf{r}_i \leftarrow \underset{\text{mat}(\mathbf{x}) \in \mathcal{SO}(3)}{\text{argmin}} \left\| \mathbf{x} - (-1)^\mu \sqrt{3}\mathbf{e}_\nu \right\|^2$

$\hat{\mathbf{r}}_i \leftarrow \text{SolveSQP}(\mathbf{r}_i, \Omega, \epsilon, T)$

$\mathcal{E}_i^2 \leftarrow \hat{\mathbf{r}}_i^T \Omega \hat{\mathbf{r}}_i$

fin desde

Algoritmo 1. SQPnP: SolvePnP.

mientras $\min \{\mathcal{E}_1^2, \dots, \mathcal{E}_{2k}^2\} \geq s_{9-k}$ hacer
 desde $i \leftarrow 1$ hasta 2 hacer
 $\mathbf{r}_{2k+i} \leftarrow \operatorname{argmin}_{\operatorname{mat}(\mathbf{x}) \in \mathcal{SO}(3)} \left\| \mathbf{x} - (-1)^i \sqrt{3} \mathbf{e}_{9-k} \right\|^2$
 $\hat{\mathbf{r}}_{2k+i} \leftarrow \operatorname{SolveSQP}(\mathbf{r}_{2k+i}, \mathbf{\Omega}, \epsilon, T)$
 $\mathcal{E}_{2k+i}^2 \leftarrow \hat{\mathbf{r}}_{2k+i}^T \mathbf{\Omega} \hat{\mathbf{r}}_{2k+i}$
 fin desde
 $k \leftarrow k + 1$
fin mientras
devolver $\hat{\mathbf{r}}_1, \dots, \hat{\mathbf{r}}_{2k}, \mathcal{E}_1^2, \dots, \mathcal{E}_{2k}^2$

La segunda parte del algoritmo, llamada SolveSQP, puede verse en la tabla 5.2. Se encarga de realizar la búsqueda de los mínimos, alcanzando una solución, por lo general, en unas 10 iteraciones:

Tabla 5.2. Segunda parte del algoritmo SQPnP [171].

Algoritmo 2. SQPnP: SolveSQP.

Requisitos:

- Punto de partida: $\mathbf{r} \in \mathbb{R}^9, \operatorname{mat}(\mathbf{x}) \in \mathcal{SO}(3)$
- Matriz de datos: $\mathbf{\Omega} \in \mathbb{R}^{9 \times 9}, \mathbf{\Omega} \succeq \mathbf{0}$
- Tolerancia a las perturbaciones: $\epsilon \leq 10^{-5}$
- Número máximo de iteraciones: $T \geq 15$

iteración $\leftarrow 0$

$\hat{\mathbf{r}} \leftarrow \mathbf{r}$

repetir

$$\mathbf{H}_{\hat{\mathbf{r}}} \leftarrow \left. \frac{\partial \mathbf{h}(\mathbf{x})}{\partial \mathbf{x}} \right|_{\mathbf{x}=\hat{\mathbf{r}}}$$
$$\begin{bmatrix} \hat{\boldsymbol{\delta}} \\ \hat{\boldsymbol{\lambda}} \end{bmatrix} \leftarrow \begin{bmatrix} \mathbf{\Omega} \mathbf{H}_{\hat{\mathbf{r}}}^T \\ \mathbf{H}_{\hat{\mathbf{r}}} \mathbf{0}_{6 \times 6} \end{bmatrix}^{-1} \begin{bmatrix} -\mathbf{\Omega} \hat{\mathbf{r}} \\ -\mathbf{h}(\hat{\mathbf{r}}) \end{bmatrix}$$
$$\hat{\mathbf{r}} \leftarrow \hat{\mathbf{r}} + \hat{\boldsymbol{\delta}}$$

iteración \leftarrow iteración + 1

hasta $\|\hat{\boldsymbol{\delta}}\| < \epsilon$ o iteración $> T$

devolver $\hat{\mathbf{r}}$

El lector interesado en una explicación más detallada tanto del algoritmo SQPnP como de su base matemática encontrará interesante el artículo original en el que fue publicado [171].

Con los conceptos básicos presentados en esta sección se está en disposición de describir el sistema de posicionamiento utilizado en este trabajo.

5.3. Estimación de la posición

En esta sección se describe el sistema utilizado para estimar la posición de una persona dentro de una sala utilizando para ello una baliza construida sobre una luminaria LED y una fotografía tomada con la cámara del dispositivo portable que dicha persona lleva en su mano. Este sistema se utiliza tanto en la simulación como en el mundo real.

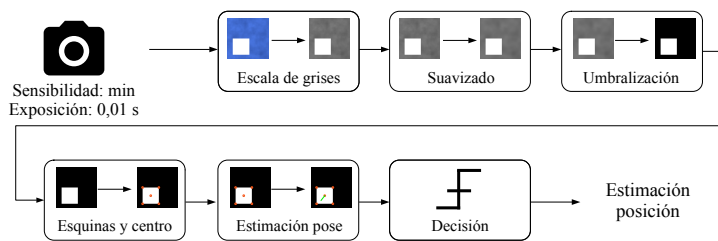


Figura 5.5. Diagrama de funcionamiento del sistema de estimación de la posición.

Antes de entrar en detalle sobre cada uno de los pasos del sistema de estimación de la posición, se puede resumir el proceso diciendo que, en primer lugar, la cámara del dispositivo portable se configura para tomar las muestras con cierta configuración que permite destacar el contorno de la baliza. Dicha muestra se procesa para destacar aún más el contorno. En ese momento, se analiza la imagen utilizando técnicas de visión por ordenador (como la localización de contornos o la aproximación de polígonos) para localizar las esquinas de la baliza y encontrar el centro de la misma. Con los datos obtenidos se estima la pose y la posición relativa de la cámara con relación a la baliza. Entonces, se combina la estimación de la posición con el código emitido por la baliza. Esto permite tomar una decisión y estimar en qué posición dentro de la sala estaba la cámara cuando se tomó la muestra. A continuación se describe cada paso del proceso en detalle.

5.3.1. Toma de muestras

Atendiendo a los motivos expuestos en la sección 3.8 se ha elegido el sistema de comunicación de doble toma. Este sistema tiene la peculiaridad de tomar dos muestras: una para determinar con precisión el contorno de la baliza y otra para obtener el código emitido por la misma. La primera de las dos muestras, por sus características, es ideal para realizar una tarea adicional: estimar la posición de la baliza en el momento en el que fue tomada.

Aunque en la sección dedicada a la doble toma se describa el sistema en detalle, es interesante recordar que la cámara se ajusta de manera que su sensibilidad tome el mínimo valor posible. Además, el tiempo de exposición se ajusta para que sea de 0,01 s, puesto que se ha comprobado experimentalmente que con ese valor se consigue que el parpadeo a alta frecuencia de la luminaria sea prácticamente inapreciable. En la figura 5.6 se puede ver un ejemplo de las dos muestras que se obtienen. Ambas fotos fueron tomadas a una distancia de 1,2 m del emisor. En la figura 5.6a se puede ver la muestra utilizada en el sistema de posicionamiento, mientras que en la figura 5.6b se puede ver la muestra utilizada en el sistema de comunicación de datos.

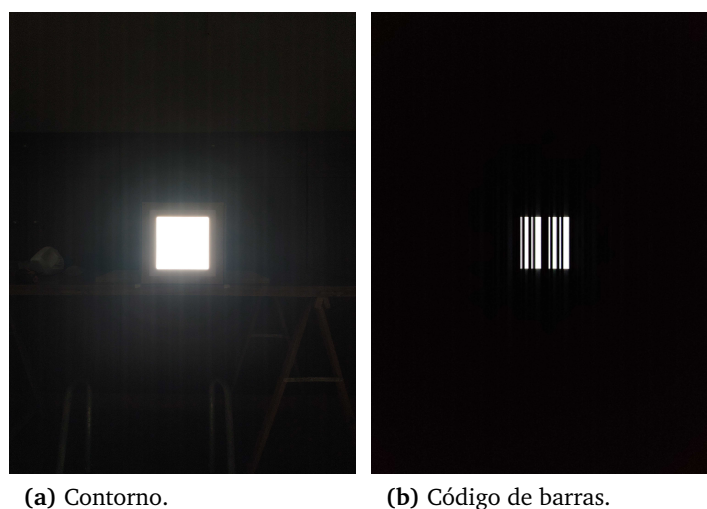


Figura 5.6. Ejemplo de toma de dos muestras.

5.3.2. Procesamiento de muestras

Antes de poder trabajar con las muestras obtenidas en el paso anterior es necesario realizar algunas operaciones sobre ellas para dejarlas en un estado óptimo. Dicho procesamiento consta de los siguientes pasos:

- **conversión de la imagen a escala de grises:** las muestras adquiridas están en formato YUV [126], que permite acceder directamente a los datos de intensidad luminosa sin necesidad de realizar ninguna operación sobre los datos originales. Como los componentes cromáticos de la imagen no aportan nada al proceso de decodificación se pueden descartar, quedando una imagen en escala de grises.
- **suavizado de la imagen:** un leve suavizado eliminará ruidos en la imagen, optimizando el resultado de los siguientes pasos.
- **umbralización de la imagen:** para poder decodificar la imagen es necesario trabajar con ceros y unos, en lugar de con valores entre 0 y 255. Utilizando una umbralización binaria que sólo se quede con los valores más intensos de la imagen.

Al terminar este procesamiento de las muestras, todo lo que no sea la baliza se corresponderá con el valor 0 (negro en la imagen), mientras que la baliza se corresponderá con el valor 1 (blanco en la imagen). En la figura 5.7 se pueden ver los pasos dados en este proceso, partiendo de la muestra original, hasta tenerla lista para su análisis. Es interesante observar como, en la figura 5.7a, se puede apreciar una perturbación en forma de líneas verticales en la iluminación de la sala debido al parpadeo de alta frecuencia emitido por la baliza. Dicha perturbación no es apreciable a simple vista por el ojo humano.

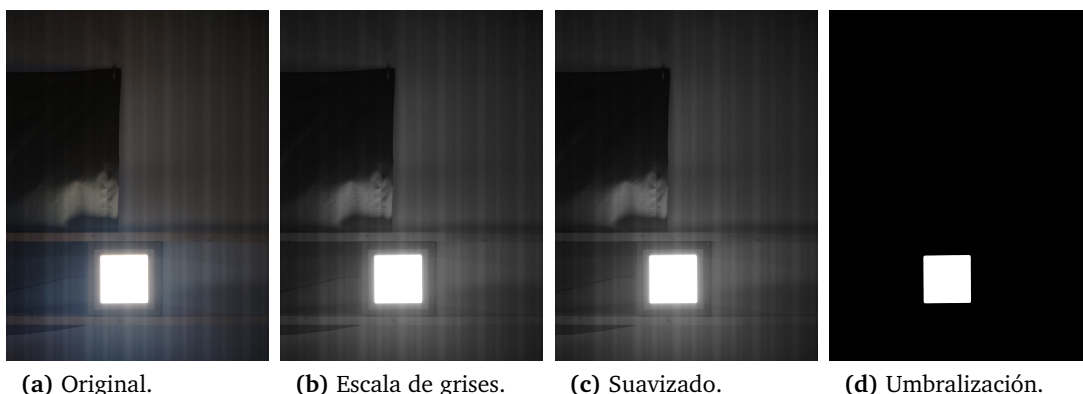


Figura 5.7. Procesamiento de muestras.

5.3.3. Búsqueda de elementos en la muestra

OpenCV es un paquete multiplataforma y de código abierto que proporciona múltiples herramientas para realizar tareas de visión por ordenador [173]. Como se describe en la sección ??, se necesitan un mínimo de 4 puntos para estimar la pose de la cámara con la que se tomó la muestra. La geometría de las balizas, cuya superficie luminosa es un cuadrado, permite obtener esos cuatro puntos localizando las esquinas. Para mayor seguridad, es posible obtener un punto más de las balizas, su centro, contando así con 5 puntos.

Localizar las esquinas de la baliza utilizando las muestras de la simulación es una tarea relativamente sencilla, gracias a las herramientas de visión por ordenador disponibles en OpenCV. La baliza del simulador es un cuadrado perfecto, como puede verse en la figura 5.8a. Sin embargo, la baliza utilizada para las pruebas en el mundo real, que puede verse, en la figura 5.8b, tiene sus esquinas redondeadas. El algoritmo necesario para localizar las esquinas debe tener en cuenta esta particularidad.

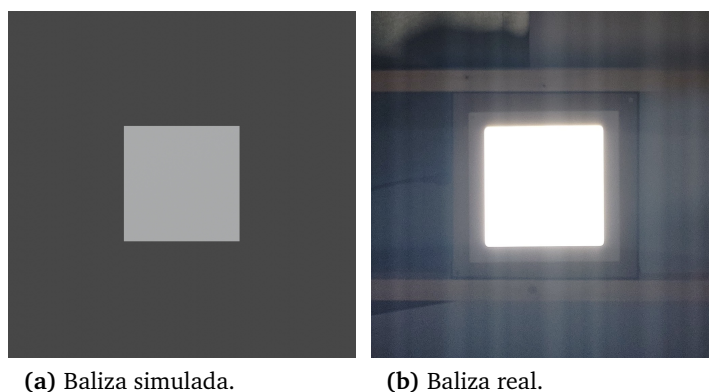


Figura 5.8. Detalle de las esquinas de las balizas utilizadas en la simulación y en el mundo real.

El proceso de detección de esquinas de la baliza comienza con la detección de los contornos presentes en la imagen resultante tras la operación de umbralización, en la que la imagen está en blanco y negro. Debido a la naturaleza de los datos presentes en estas muestras, las zonas en blanco se corresponden con las balizas y el resto estará en negro. Los contornos detectados en estas muestras envolverán por completo a la baliza, delimitando la zona en la que se encuentre.

Aunque las balizas sean perfectamente cuadradas, este contorno no tiene por qué estar compuesto por cuatro segmentos rectilíneos, de esquina a esquina. Debido a las imperfecciones de la imagen adquirida tanto en la simulación como en el mundo real, dicho contorno podría estar compuesto por varios segmentos de línea, uno a

continuación de otro. En este momento comienzan las diferencias entre el proceso de localización de esquinas en las muestras tomadas en la simulación y las tomadas en el mundo real. El proceso, en ambos casos, comienza con el contorno de la baliza presente en la muestra.

Detección de esquinas en la simulación

Para disminuir el contorno a su mínima expresión en la simulación basta con aproximar el contorno a un polígono con menos vértices. Para ello se utiliza el algoritmo Ramer–Douglas–Peucker [174], capaz de reducir el número de segmentos que definen una curva. Como criterio se utiliza un valor que determina la máxima distancia entre la curva original y su aproximación. En este caso en particular, se trata de la centésima parte del perímetro del contorno. El resultado es un contorno con cuatro esquinas en todos de los casos que se han probado. Sin embargo, si el contorno resultante no tuviese cuatro esquinas, el proceso de detección se daría por incorrecto. En la figura 5.9a puede verse el resultado de detectar las esquinas siguiendo este algoritmo.

Detección de esquinas en el mundo real

Debido a que las balizas utilizadas en el mundo real tienen las esquinas redondeadas, el contorno de partida está compuesto por dos zonas bien diferenciadas. Por una parte están las zonas rectas en cada lado de la baliza, por el otro las zonas curvas en cada esquina. Por esta razón, el resultado de aproximar el contorno utilizando el algoritmo Ramer–Douglas–Peucker no es óptimo, como puede comprobarse en la figura 5.9b. El contorno final está próximo al contorno esperado, pero idealmente debería contener por completo a la baliza, dejando un hueco en las esquinas redondeadas. El objetivo es trabajar con un contorno que contenga en su interior el total de la baliza. Para ello, el primer paso consiste en convertir el contorno encontrado en un casco convexo utilizando el algoritmo de Sklansky [175] y, a continuación, utilizar el algoritmo Ramer–Douglas–Peucker, pero esta vez manteniendo el tamaño del perímetro idéntico. Como resultado, el contorno se mantiene alrededor de la baliza, pero el número de vértices es menor, lo que facilita el trabajo posterior.

En este punto se dispone la lista de puntos que trazan el contorno de la baliza. Dicha lista de puntos se transforma en una lista de segmentos, el final de uno unido al comienzo del siguiente. El último segmento enlaza con el primero para terminar el contorno. Junto con la lista de segmentos se guarda una lista con las longitudes de

los mismos. Los segmentos con menor longitud son aquellos que definen la curva de las esquinas. Para eliminarlos, se calcula la media la longitud de los segmentos, descartando aquellos segmentos que estén por debajo de ella.

Los segmentos de mayor longitud, que forman los lados de la baliza, se concatenan para formar un único segmento, se toman los cuatro más largos y se calcula su intersección, que se corresponde con las esquinas de la baliza. El resultado puede verse en la figura 5.9c.

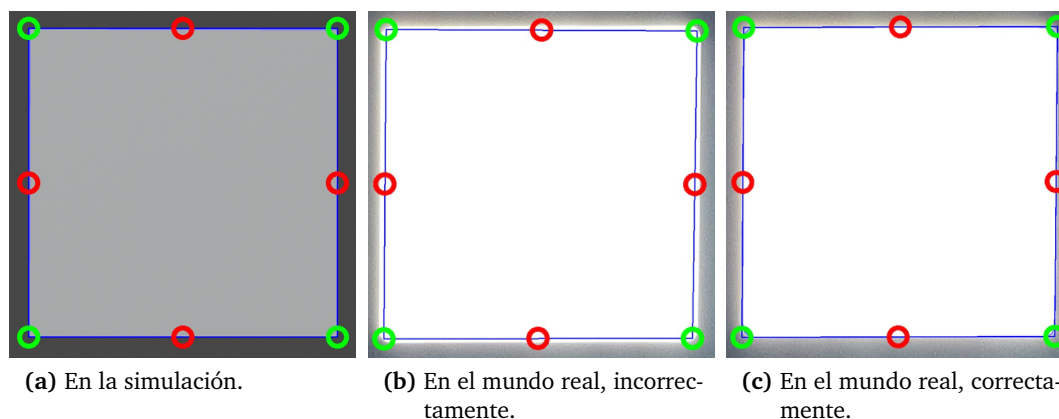


Figura 5.9. Comparación de la detección de esquinas.

En ambos casos (simulación y mundo real), tras localizar las esquinas estas se ordenan en sentido contrario a las agujas de reloj, para que siempre se trabaje en las mismas condiciones.

Las esquinas están marcadas en verde en la figura 5.9. Los puntos rojos en dicha figura marcan el punto medio de cada lado. Uniendo los puntos medios opuestos se trazan dos segmentos cuya intersección determina el centro de la baliza. Con las cuatro esquinas y el centro se está disposición de estimar la pose de la cámara cuando la muestra fue tomada.

5.3.4. Obtención de la traslación y la rotación de la cámara

Como se comenta en la sección 5.2.3, el algoritmo SQPnP soluciona el problema PnP de manera óptima, buscando los mínimos que satisfacen las condiciones del problema. OpenCV ofrece una implementación de este algoritmo a través del método `solvePnP`. Este método ofrece una implementación del algoritmo SQPnP si se le proporciona el valor `SOLVEPNP_SQPMP` siempre que se disponga de los siguientes datos:

- los puntos del modelo del objeto cuya pose se quiere estimar.
- los puntos de dicho objeto en una imagen.
- la matriz intrínseca de la cámara utilizada para capturar la imagen.
- los coeficientes de distorsión de dicha cámara.

Puntos del modelo del objeto

Para los objetivos de este trabajo, el modelo del objeto describe sus características si estuviese exactamente en el plano de proyección de la cámara. Es decir, se trata de las dimensiones de la baliza si no hubiese distancia entre su superficie y la cámara del dispositivo portable. Para poder crear este modelo es necesario conocer el lado de la baliza cuadrada y el tamaño del píxel de la cámara. El primer valor es conocido, puesto que se conocen las propiedades geométricas de la baliza. El segundo valor lo proporciona el fabricante del sensor óptico integrado por el fabricante del dispositivo portable.

El centro del modelo de la baliza está en las coordenadas $(0, 0)$. Cada una de las esquinas de la baliza, puesto que la utilizada en este trabajo es cuadrada, está a la mitad del lado de distancia del centro en ambos ejes, cubriendo los cuatro cuadrantes del plano. El lado del modelo del objeto se calcula multiplicando la longitud del lado por el tamaño del píxel de la cámara. El lado de la baliza está medido utilizando el sistema métrico decimal. Las unidades del tamaño del píxel están en esas mismas unidades por píxel. El resultado, por tanto, está en píxeles. Así, si el lado de la baliza midiese 174 mm y el tamaño del píxel de la cámara fuese $0,9 \mu\text{m}/\text{px}$, el lado del modelo de la baliza, en píxeles, sería:

$$\begin{aligned} \text{lado modelo baliza} &= \frac{\text{lado baliza}}{\text{tamaño píxel}} = \frac{174 \text{ mm}}{0,9 \mu\text{m}/\text{px}} = \frac{174 \cdot 10^{-3} \text{ m}}{0,9 \cdot 10^{-6} \text{ m}/\text{px}} \\ &= \frac{174}{0,9} \cdot 10^3 \text{ px} = 193\,333,33 \text{ px} \end{aligned} \quad (5.28)$$

Por lo tanto, si el plano de la baliza estuviese pegado a la cámara y su sensor fuese capaz de capturarlo por completo, el lado de la baliza ocuparía 193 333,33 px. Con este valor, con el centro de la baliza en el origen de coordenadas, cada esquina del modelo de la baliza empezando por la superior izquierda y avanzando en sentido horario estarían en:

- $(-96\,666,66, +96\,666,66)$.
- $(+96\,666,66, +96\,666,66)$.

- $(+96\,666,66, -96\,666,66)$.
- $(-96\,666,66, -96\,666,66)$.

Si la baliza no fuese cuadrada, el modelo del objeto tendría que crearse de una manera equivalente, y la obtención de los puntos de la imagen del objeto debería modificarse de acuerdo al modelo. Puesto que el algoritmo **ALM** necesita, al menos, 4 puntos para estimar la pose de un objeto plano, la geometría de la baliza podría ser, como mínimo, triangular, utilizando sus tres vértices y el centro para estimar la pose. En el otro extremo estarían balizas con formas circulares, de las que habría que tomar tantos puntos de su contorno como se considerase oportuno para aproximar la forma circular mediante un polígono, manteniendo el error por debajo del margen de tolerancia.

Puntos de la imagen del objeto

La manera en la que los puntos del objeto se localizan en la muestra se describe en la sección 5.3.3. Aquí, esta tarea aparece en segundo lugar dentro de la lista de requisitos para la estimación de la pose por ser el segundo parámetro que el método `solvePnP` necesita. En la práctica se trata de la condición sin la cual no se puede proceder a la estimación de la pose. Por lo tanto, si no resulta posible encontrar las cuatro esquinas y el centro de la baliza en la muestra, la estimación de la pose no podrá llevarse a cabo.

Matriz intrínseca de la cámara

En la sección 5.2.2 se puede ver la definición matemática de la matriz intrínseca de la cámara, que forma parte del sistema de proyección de los objetos del mundo real en el sensor de la cámara. La distancia focal $f = (f_x, f_y)$ es un parámetro que depende del fabricante del dispositivo portable, y determina el espacio que ha dejado entre la lente y el sensor de la cámara. Por lo general, suele tener el mismo valor en ambas coordenadas, por que lo $f_x = f_y$. Este valor está en las especificaciones del dispositivo portable y suele aparecer entre los campos Exif incluidos en las imágenes capturadas con el sensor[124]. Por otra parte, el centro óptico $c = (c_x, c_y)$ de la cámara coincide con el centro de la imagen capturada.

En el caso de la cámara frontal del dispositivo portable utilizado durante el desarrollo del sistema de posicionamiento, un Xiaomi Mi 8 cuyas características ya se han descrito en la página 139, el tamaño del píxel es $0,9\ \mu\text{m}/\text{px}$, la distancia focal es

3,52 mm y las muestras, en modo retrato, tienen un ancho de 3880 px y un alto de 5184 px. Por lo tanto, en este caso particular, la matriz intrínseca de la cámara es:

$$\begin{aligned}
 f_x = f_y &= \frac{3,52 \text{ mm}}{0,9 \text{ } \mu\text{m/px}} = \frac{3,52 \cdot 10^{-3} \text{ m}}{0,9 \cdot 10^{-6} \text{ m/px}} = \frac{3,52}{0,9} 10^3 \text{ px} = 3911,11 \text{ px} \\
 c_x &= \frac{3880 \text{ px}}{2} = 1940 \text{ px} \quad ; \quad c_y = \frac{5184 \text{ px}}{2} = 2592 \text{ px} \\
 \mathbf{I} &= \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 3911,11 & 0 & 1940 \\ 0 & 3911,11 & 2592 \\ 0 & 0 & 1 \end{bmatrix}
 \end{aligned} \tag{5.29}$$

Coeficientes de distorsión de la cámara

Al explicar el modelo de cámara estenopeica (ver la sección 5.2.1) se comentó que entre el objeto y el sensor óptico se coloca una lente para concentrar la luz en el sensor. Estas lentes no son perfectas y pueden presentar diferentes tipos de distorsión que afectan a la formación de la imagen. El sistema más utilizado para estimar dichos coeficientes consiste en realizar un proceso de calibración durante el que la cámara toma un determinado número de fotografías de un patrón conocido que son posteriormente analizadas en busca de aberraciones. Tras dicho proceso, los coeficientes de distorsión se incorporan al proceso de tratamiento de las imágenes para obtener un resultado más preciso [87].

Aunque añadir un paso de calibración al sistema desarrollado aquí podría mejorar los resultados obtenidos, se ha descartado porque resultaría molesto para el usuario. Antes de utilizar cualquier aplicación móvil derivada de lo aprendido en este trabajo, el usuario debería leer unas instrucciones en las que se le explicaría cómo tomar ciertas fotografías a una distancia específica, pidiéndolo que repitiese el proceso en caso de que algún paso no se hubiese realizado correctamente, retrasando el uso de la app o incluso haciéndolo imposible porque el usuario no fuese capaz de seguir las instrucciones o porque su paciencia se agotase.

Esto no quiere decir que se descarte por completo, y prueba de ello es la presencia de esta sección en el capítulo. Entre las futuras mejoras que podrían incorporarse al sistema no sólo está la estimación de los coeficientes de distorsión sino el desarrollo de un proceso de calibración lo menos molesto posible. Sin embargo, en su estado actual, el sistema asume que no existe distorsión alguna en la lente.

En este punto, todos los datos necesarios para realizar la estimación de la pose están disponibles. El resultado principal del proceso es una bandera booleana que determina si fue posible o no reconstruir la pose. Si no lo fue, se toma otra muestra para repetir el proceso completo desde el principio. Si lo fue, `solvePnP` proporciona un vector de rotación y otro de traslación que permiten estimar la posición relativa de la cámara con relación a la baliza al tomar la muestra.

5.4. Cálculo del FOV de la cámara

Se llama **FOV** a la porción de realidad que un sensor óptico es capaz de percibir. En el caso de una persona, su **FOV** se corresponde con lo que puede alcanzar a ver de izquierda a derecha y de arriba a abajo cuando está totalmente quieto. En el caso particular de este trabajo, se habla del **FOV** de las cámaras utilizadas para el posicionamiento.

Existen dos razones por las que es necesario calcular el **FOV** con precisión:

1. **para evaluar el sistema de posicionamiento.** Conociendo el **FOV** es posible realizar una división del mismo en una malla y mover la cámara a cada una de las celdas resultantes, tomar una muestra, estimar la posición y calcular la desviación entre la posición estimada y la real.
2. **para determinar cuántas balizas se necesitan en un espacio concreto.** Conocido el **FOV** de una determinada baliza, es posible calcular a qué distancia entre sí se deben colocar varias de ellas para cubrir una determinada superficie.

Las cámaras de los dispositivos portables actuales utilizan la estructura de la cámara estenopeica que puede verse en la figura 5.1, presentada en la sección 5.2.1. El valor de la distancia focal f es configurable, pudiendo utilizarse el de la cámara cuyas características se quieran utilizar en la estimación del rendimiento del sistema de posicionamiento. Las dimensiones del sensor de la cámara (ancho y alto) se corresponderían con los componentes de la matriz 2×1 de \mathbf{a} , y la distancia a la que se encuentra la baliza con D . Dados estos valores y utilizando trigonometría es posible conocer el ancho y el alto del **FOV**, que se correspondería con los componentes de la matriz 2×1 de \mathbf{A} .

Sobre el eje central de la figura 5.1 existen dos triángulos rectángulos enfrentados. En el triángulo de la izquierda, la longitud del lado opuesto al ángulo α es $\frac{a}{2}$, es decir, la mitad de cada dimensión del sensor de la cámara. La longitud del lado contiguo es la distancia focal con la que el fabricante ha decidido montar el sensor

óptico en el dispositivo portable. En el triángulo de la derecha, la longitud del lado opuesto al ángulo α es $\frac{A}{2}$, es decir, la mitad de cada dimensión del **FOV**. La longitud del lado contiguo es la distancia entre la cámara y la baliza. Así, se cumple que:

$$\tan \alpha = \frac{a}{f} \quad ; \quad \tan \alpha = \frac{\frac{A}{2}}{D} \quad (5.30)$$

De la ecuación (5.30) se puede deducir que:

$$\frac{a}{f} = \frac{\frac{A}{2}}{D} \Rightarrow \mathbf{A} = \mathbf{a} \cdot \frac{D}{f} \quad (5.31)$$

Si P es el tamaño del píxel en mm/px y \mathbf{R} es una matrix de 2×1 que contiene como componentes el ancho y el alto de la muestras tomadas con la cámara del dispositivo portable, se deduce que:

$$\mathbf{a} = \mathbf{R} \cdot P \quad (5.32)$$

Por lo tanto, siguiendo la ecuación (5.31) y la ecuación (5.32) se tiene que:

$$\mathbf{FOV} = \mathbf{A} = \mathbf{R} \cdot P \cdot \frac{D}{f} \quad (5.33)$$

donde \mathbf{A} es una matriz de 2×1 cuyos componentes son el ancho y el alto del **FOV**. Esto significa que es posible calcular las dimensiones del **FOV** conocidos el tamaño del pixel, la distancia focal y las dimensiones de cada muestra que toma la cámara, así como la distancia a la que se encuentra la cámara de la baliza. Utilizando estos valores es posible representar el **FOV** en la simulación y ver qué observa la cámara si se sitúa en una de las esquinas del mismo. Si los cálculos son correctos, debería poder verse la baliza en su totalidad desde cualquiera de sus esquinas, justo en el borde de la muestra, ya que está en el extremo del campo visual. En la parte izquierda de la figura 5.10 se puede ver, seleccionado, un plano que representa al **FOV** que se acaba de calcular¹. La cámara está situada en la esquina superior derecha del mismo. En la parte derecha de la figura 5.10 se puede observar lo que ve la cámara. Como se puede comprobar, no toda la baliza está visible.

Sin embargo, los cálculos son correctos y la figura 5.10 lo confirma, puesto que en el extremo del **FOV** se encuentra justo el centro de la baliza. Sólo es necesario reducir

¹En la sección 4.8 se puede encontrar más información sobre la simulación utilizada para desarrollar el sistema de posicionamiento, creada utilizando Blender y Python.

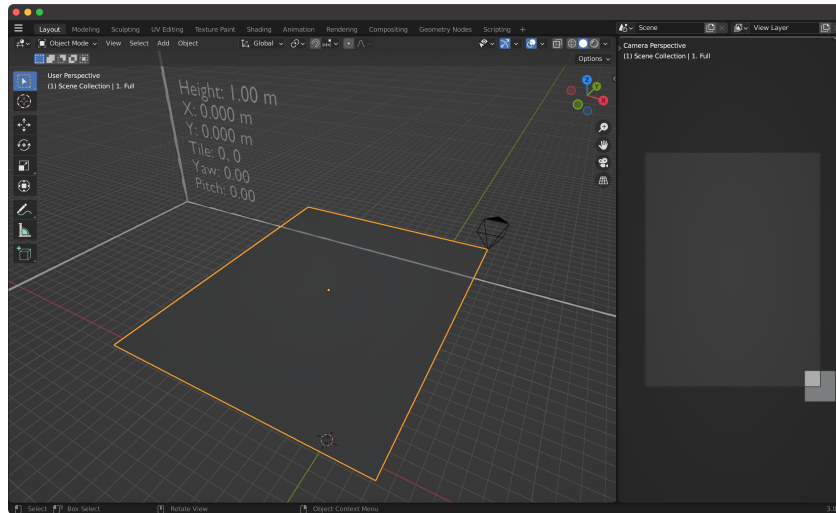


Figura 5.10. Cámara en una esquina del FOV completo.

la superficie del FOV restando media baliza en cada dimensión para conseguir que la baliza completa esté dentro del FOV en cada punto de la misma:

$$\text{FOV} = a \cdot \frac{D}{f} - B \quad (5.34)$$

donde B es una matriz de 2×1 cuyos componentes son el ancho y alto de la baliza, respectivamente. En la figura 5.11 se puede ver la misma situación que mostraba la figura 5.10, pero usando el FOV calculado en la ecuación (5.34). En este caso, toda la baliza es visible en los extremos, confirmando que con la ecuación (5.34) resulta posible obtener las dimensiones de un FOV tal que la baliza completa siempre sea visible en toda su superficie.

El FOV de una cámara digital depende de dos factores: el tamaño del sensor y la distancia focal [89]. Sin embargo, en este trabajo entra en juego un tercer factor: el tamaño de la baliza. Ahora bien, para evaluar el rendimiento del sistema de posicionamiento desarrollado no resulta factible tomar muestras en todos y cada uno de los puntos del FOV. Es necesario cuantificar ese espacio, dividiéndolo en celdas. En este caso, será dividido utilizando celdas de igual ancho que largo. La cámara se situará en las esquinas de cada celda y tomará una muestra desde esa posición. El origen de coordenadas está en el centro del FOV. Por ejemplo, en la figura 5.12 la cámara está en la celda (1, 3).

Para cubrir el FOV calculado en la ecuación (5.34) con una malla con celdas cuadradas de lado conocido bastaría con dividir cada lado del FOV entre el lado de la celda, desechando el resto:

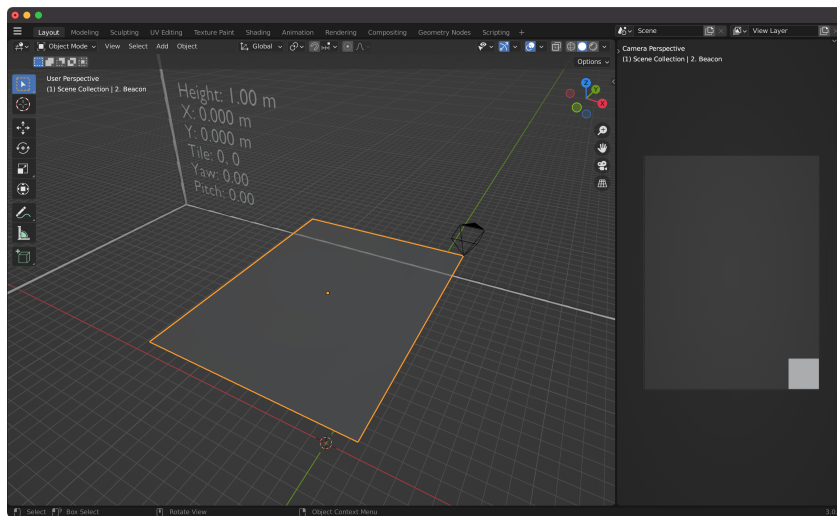


Figura 5.11. Cámara en una esquina del FOV ajustado a la baliza.

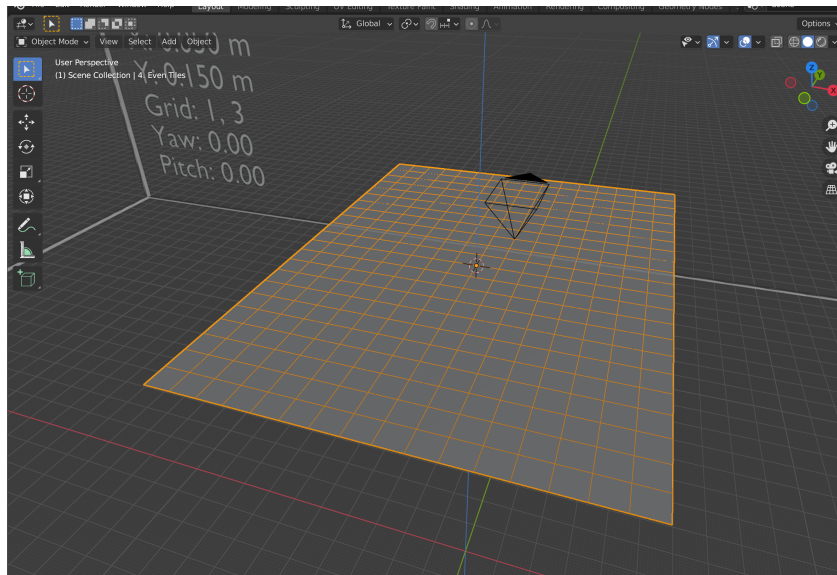


Figura 5.12. División del FOV en celdas.

$$\Delta = \left\lfloor \frac{\text{FOV}}{\delta} \right\rfloor \quad (5.35)$$

donde δ es el la longitud del lado de la celda cuadrada y Δ es una matriz de 2×1 cuyos componentes son el ancho y el alto del **FOV** medido en celdas. Por último, podría darse el caso de que la división en celdas del **FOV** resultase en una malla con celdas impares, con lo que el origen de coordenadas no coincidiría con las esquinas de cuatro celdas. Para cumplir esta condición basta con calcular $\Delta \bmod 2$. Si es igual a cero el resultado se mantiene, pero si es diferente se debe restar una unidad a dicha dimensión del **FOV** para hacerla par:

$$\Delta = \begin{cases} \Delta & \text{si } \Delta \bmod 2 = 0 \\ \Delta + 1 & \text{si } \Delta \bmod 2 \neq 0 \end{cases} \quad (5.36)$$

En conclusión, es posible obtener el **FOV** como sigue:

$$\text{FOV} = \Delta \cdot \delta \quad (5.37)$$

En la figura 5.13 se pueden ver los cuatro **FOV** descritos, con una ligera separación en el eje Z para que puedan compararse sus diferentes dimensiones. La baliza tiene 0,17 m de ancho y alto y está situada a 2 m de altura. La cámara simulada es la frontal de un Xiaomi MI 8 [120], en formato retrato y está situada a 1 m de altura, cuyas características ya se han descrito en la página 139. Las celdas tienen 0,05 m de lado. Con estas condiciones, las dimensiones de cada **FOV**, primero el ancho y luego el alto, de abajo a arriba, son:

- **completo:** 0,992 m \times 1,33 m.
- **baliza:** 0,818 m \times 1,15 m.
- **celdas:** 0,8 m \times 1,15 m.
- **celdas pares:** 0,8 m \times 1,1 m.

El **FOV** final tiene 16 celdas de ancho por 12 de alto.

Para facilitar la comprobación de los límites del **FOV** se ha incluido una herramienta en el simulador que permite capturar muestras en cada una de las cuatro esquinas de cada **FOV**, guardando los resultados para su posterior consulta. La utilización del simulador se describe en la sección 4.8.3.

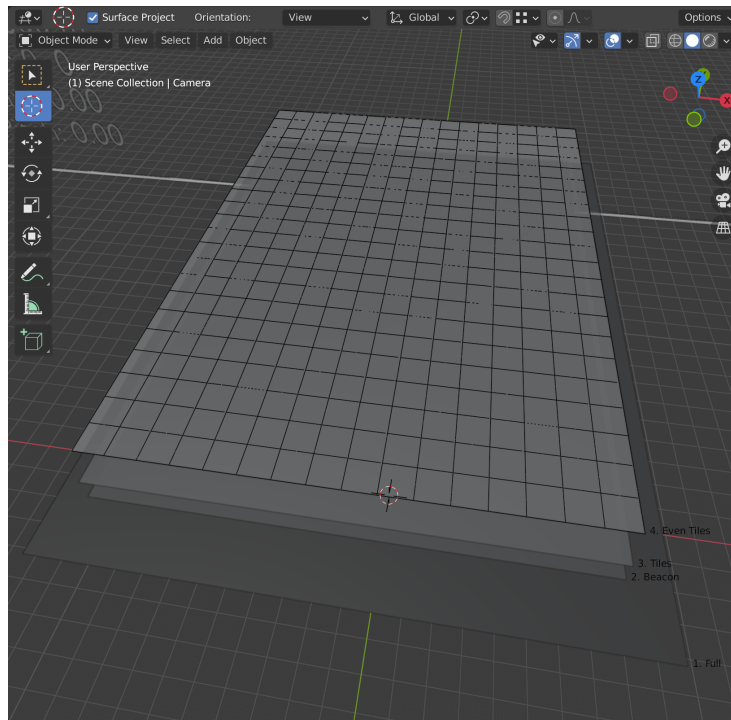


Figura 5.13. Los cuatro FOV calculados.

5.5. Conclusiones

En este capítulo se ha descrito un sistema de posicionamiento en interiores basado en luz visible capaz de estimar la posición de un dispositivo portable con relación a una baliza basada en tecnología LED. Además, se han presentado las bases para determinar la fiabilidad del sistema en base a una serie de muestras tomadas en la zona de cobertura ideal de una de dichas balizas. En su estado actual, el sistema sólo necesita para funcionar una serie de valores proporcionados por el fabricante del dispositivo portable y del sensor óptico, obteniendo el resto de valores de las muestras tomadas con la cámara del dispositivo portable. En el siguiente capítulo se realizarán todas las pruebas necesarias para validar el sistema de posicionamiento descrito en este capítulo.

Resultados experimentales

Durante los capítulos anteriores se han descrito los diferentes componentes necesarios para poder construir y probar un sistema de posicionamiento en interiores basado en luz visible tipo **LED** que permita estimar la posición del usuario mediante el uso de un dispositivo portable. Partiendo de un estudio del actual estado en el que se encuentra la investigación en este campo, se ha desarrollado un sistema de comunicación y construido unas balizas programables que lo utilizan. También se han desarrollado múltiples herramientas software: una aplicación para tomar muestras desde un dispositivo portable, extraer los mensajes enviados por las balizas y descifrarlos, scripts para trabajar desde un ordenador con las muestras tomadas e incluso un simulador del sistema de posicionamiento.

Llegados a este punto se está en disposición de utilizar todas estas herramientas de forma conjunta y realizar las pruebas necesarias para determinar el rendimiento del sistema de posicionamiento tanto en la simulación desarrollada como en el mundo real, y comparar los resultados obtenidos. En este capítulo se describen los bancos de pruebas diseñados, las pruebas realizadas y se analizan los resultados obtenidos en cada uno de ellos.

6.1. Pruebas en la simulación

En la sección 4.8 del capítulo dedicado al software desarrollado para este trabajo se presentó una simulación del sistema de posicionamiento creada en Blender. Este sistema permite simular una sala, la cámara del dispositivo portable y una baliza de **VLC**. La baliza se sitúa en el centro del techo de la sala simulada. La posición de la cámara dentro de la sala puede establecerse con precisión. En cualquier momento, es posible tomar una muestra de lo que la cámara ve dentro de la simulación para su posterior análisis. Además, es posible utilizar esta herramienta para determinar el rendimiento del sistema de posicionamiento propuesto.

6.1.1. Descripción

Para las pruebas de rendimiento realizadas utilizando el simulador, los diferentes elementos presentes se configuran como sigue:

- **escena:** las celdas a través de las que la cámara se desplaza dentro del **FOV** de la cámara tienen 0,05 m de lado.
- **sala:** sus dimensiones son 6 m de ancho \times 4 m de profundidad \times 2 m de altura.
- **baliza:** se trata de una baliza cuadrada cuyo lado mide 0,174 m.
- **cámara:** los valores necesarios para simular la cámara frontal del dispositivo portable utilizado, un Xiaomi Mi 8, se describen en la página 139. En la figura 6.1 se puede ver el aspecto de la simulación utilizada. En la esquina inferior izquierda aparece el panel de configuración de la cámara, con los valores utilizados. Además, en él se establece la distancia entre la cámara y la baliza, es decir, entre la cámara y el techo (dado que la baliza está en el techo de la sala). En este caso ambos están a 1 m de distancia. Como la altura de la sala ha sido establecida en 2 m, la cámara está a 1 m del suelo.

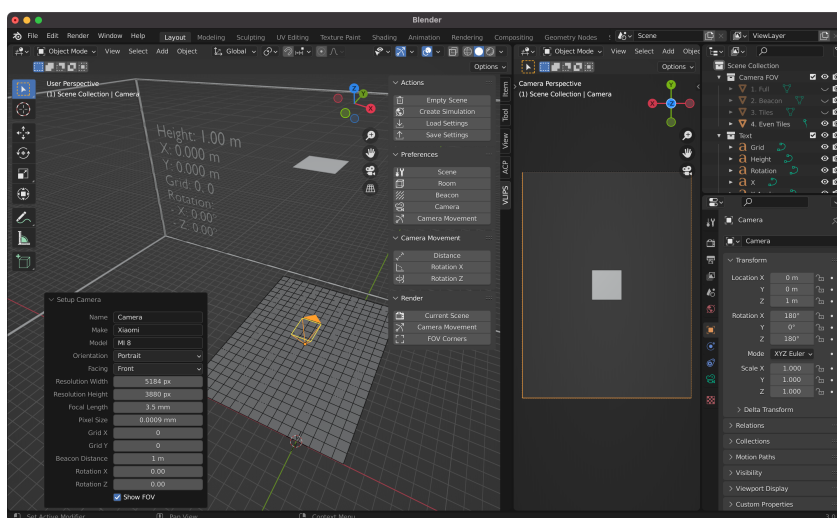


Figura 6.1. Configuración de la simulación para las pruebas, con el panel de ajustes de la cámara abierto.

Una vez configurados los diferentes elementos de la simulación, se procede a tomar muestras en cada una de las intersecciones de las celdas del **FOV**, que tiene unas dimensiones de 0,8 m de ancho \times 1,1 m de largo. Con la configuración antes descrita, se trata de 391 muestras. Además, el simulador permite describir movimientos a través del **FOV** con variaciones en la inclinación en el eje Z y el eje X.

La rotación alrededor del eje Z también se conoce como dirección. Si la pantalla del dispositivo portable se mantiene paralela al plano y mirando hacia arriba, al realizar esta rotación se determina hacia dónde apunta la parte delantera del teléfono.

La rotación alrededor del eje X se conoce como elevación. Partiendo de una posición en la que la superficie del dispositivo portable se mantiene paralela al plano y mirando hacia arriba, al realizar esta rotación la pantalla estará cada vez más inclinada hacia el usuario del dispositivo portable.

Se obvia la rotación alrededor del eje Y, conocida como alabeo, por tratarse de un movimiento que los usuarios de los dispositivos portables rara vez realizan. En él, partiendo de una posición en la que la superficie del dispositivo portable se mantiene paralela al plano y mirando hacia arriba, el usuario inclinaría la pantalla a izquierda o a derecha, con lo que cada vez vería menos de la misma.

Se toman dos grupos de muestras. En el primero de ellos, se mantiene fija la rotación en torno al eje Z en 0° mientras que la rotación alrededor del eje X comienza en 0° y llega hasta 80° , avanzando en incrementos de 10° . La razón por la que se ha escogido este rango de inclinaciones con respecto al eje X se debe a que, cuando es de 90° la cámara frontal del dispositivo portable está perpendicular al techo, con lo que la visibilidad de la baliza es nula. A medida que se aumenta la rotación pasados los 90° , la cámara sigue sin ver a la baliza.

El mismo proceso se repite para el segundo grupo de muestras, aunque manteniendo fija la rotación en torno al eje X, a 0° , mientras que la rotación alrededor del eje Z comienza en 0° y llega hasta 80° , avanzando en incrementos de 10° . En este caso, la toma de muestras no avanza pasados los 80° porque los resultados de las rotaciones con respecto al eje Z en cada uno de los cuatro cuadrantes son simétricos, dado que se está trabajando con una baliza cuadrada.

Como resultado, en cada grupo de pruebas se obtienen 9 conjuntos de 391 muestras, es decir, un total de 7038 muestras para analizar.

El simulador permite describir estos movimientos para automatizar la toma de muestras. En la figura 6.2a las dos casillas de verificación marcadas indican que la cámara tendrá que desplazarse por todo el FOV y que, al terminar, deberá repetirlo variando el ángulo con respecto al eje Z. Además, se puede seleccionar la ubicación en la que se guardan las muestras y añadir un prefijo a los archivos para diferenciarlos de las que ya pudiesen existir en la carpeta seleccionada.

Por otra parte, en la figura 6.2b se puede ver el panel utilizado para indicar la rotación alrededor del eje Z antes descrita: de 0° a 80° en incrementos de 10° . Existe un panel equivalente para indicar la rotación alrededor del eje X.

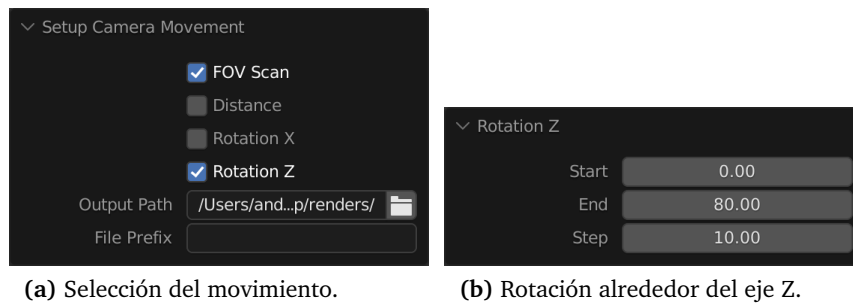


Figura 6.2. Paneles para la configuración de movimientos de cámara en el simulador.

6.1.2. Resultados

Tras la realización de la simulación y el procesamiento de los datos en cada uno de los puntos de test se obtuvieron los resultados que se pueden observar en las figuras 6.3, 6.5. En estas figuras se puede observar que las posiciones en las que se tomó cada muestra están marcadas con cruces negras. Los círculos azules se corresponden con las posiciones medias estimadas por el sistema de posicionamiento. Se ha establecido como margen de confianza el lado de la celda (0,05 m), descartando cualquier estimación de la posición que sitúe al receptor fuera de la celda en la que se encontraba originalmente. Las estimaciones de la posición en las que el error en dos dimensiones está por encima de dicho margen se han considerado anómalas, marcándolas en rojo para evitar que se confundan con otros puntos vecinos. Existen dos razones por las que una posición no tiene estimación asociada: el resultado puede estar fuera de la zona de pruebas, o no ha sido posible calcularla.

En la figura 6.3 se exponen los gráficos generados a partir de las muestras tomadas modificando la rotación alrededor del eje X en incrementos de 10 grados, partiendo de una posición en la que la pantalla del dispositivo portable estaba paralela con la superficie del techo (0°), con la cámara mirando hacia arriba. Una vez superados los 50° de inclinación el valor de la cobertura es del 0 %, razón por la que esos gráficos no se incluyen en la figura.

A medida que el ángulo de rotación alrededor del eje X aumenta, la parte inferior de las gráficas mostradas en la figura 6.3 comienza a despoblarse. Las zonas sin estimaciones de la posición se corresponden con aquellas desde las que la baliza deja de verse totalmente debido a la rotación del dispositivo portable. Ya en la coordenada

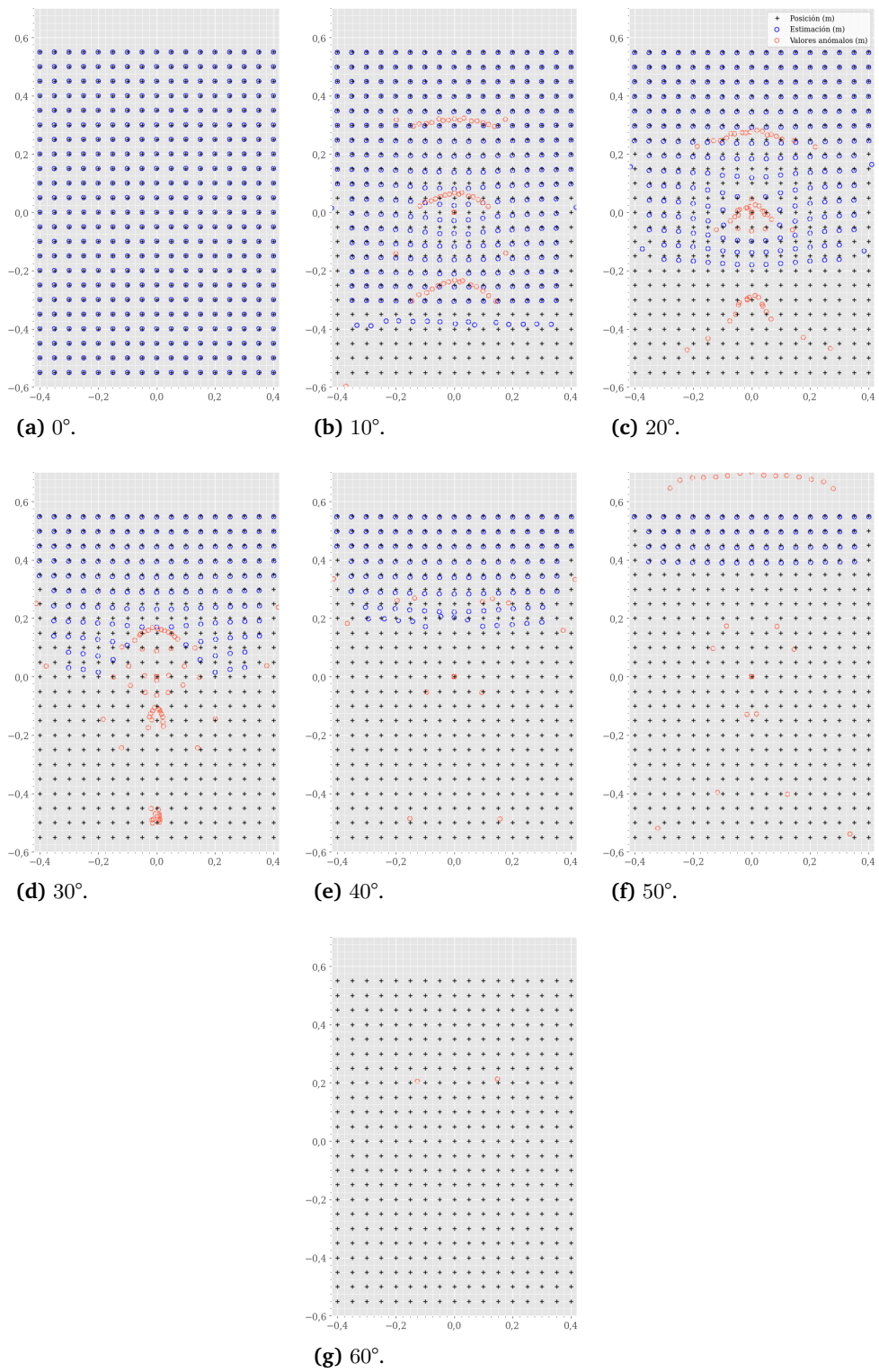


Figura 6.3. Resultados de posicionamiento en todo el FOV con diferentes rotaciones alrededor del eje X.

Y $-0,35$ m, el error en la estimación es más pronunciado, pero no lo suficiente como para que las estimaciones sean consideradas anómalas. En la figura 6.4 se pueden ver las muestras tomadas en las coordenadas Y $-0,35$ m, $-0,40$ m, $-0,45$ m y $-0,50$ m. Se omiten las tomadas en la coordenada Y $-0,55$ m puesto que en ella la baliza no resulta visible.

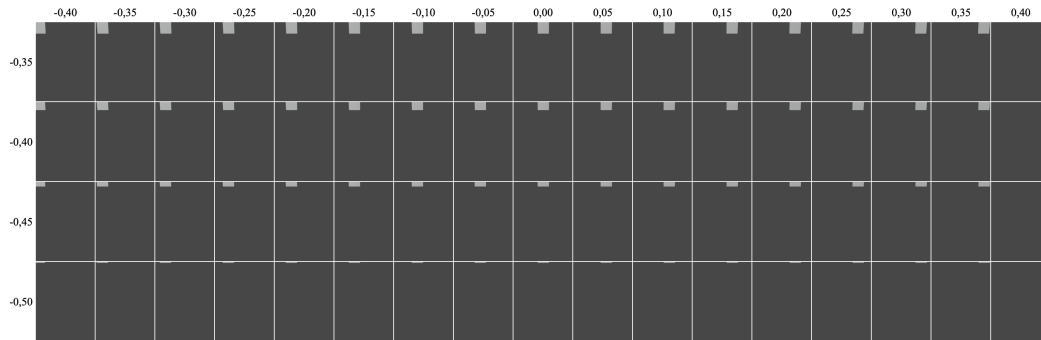


Figura 6.4. Muestras tomadas con rotación X 10° en las coordenadas Y $-0,35$ m, $-0,40$ m, $-0,45$ m y $-0,50$ m, cuando la baliza es parcialmente visible.

Los valores anómalos siguen patrones muy característicos en la figura 6.3. En concreto, la figura 6.3b muestra tres curvas de valores en rojo. Dichas curvas se corresponden con las estimaciones realizadas para las muestras tomadas en las coordenadas Y $-0,40$ m, $-0,45$ m y $-0,50$ m. Cuanto menos baliza se captura en la muestra, más se aleja la estimación de la posición original. Cuanto mayor es la diferencia entre la superficie real de la baliza y la capturada por estar fuera del campo de visión del receptor, mayor el error en la estimación de la posición, ya que aparenta encontrarse más alejada. La primera curva de valores anómalos, entre las coordenadas Y $-0,20$ m y $-0,40$ m, se corresponde con las muestras tomadas en la coordenada Y $-0,40$ m. La segunda curva (entre $0,00$ m y $-0,20$ m), con las tomadas en la coordenada Y $-0,45$ m. Por último, la tercera curva, la más alejada, con las muestras tomadas en Y $-0,50$ m. En la coordenada Y $-0,55$ m la baliza no era visible, por lo que no se ha podido realizar una estimación de la posición.

Este mismo efecto se repite para todas las rotaciones con respecto al eje X hasta que la baliza deja de ser visible. Como se puede apreciar en las figuras 6.3, 6.4, existe simetría alrededor del eje Y.

En la tabla 6.1 se ofrecen algunos parámetros que describen cuantitativamente la precisión en la estimación del posicionamiento en dos dimensiones y la cobertura obtenida para los ángulos de rotación mostrados en la figura 6.3 hasta 50° . Por encima de ese valor la cobertura es del 0%. Estos valores se han calculado sólo para aquellas muestras en las que el error en la estimación está por debajo de los $0,05$ m del margen de confianza. El valor de la cobertura se representa mediante el

porcentaje obtenido como el número de puntos en los que el error en la estimación está por debajo del margen de confianza respecto del total de medidas tomadas en cada punto de test, multiplicado por cien. Como se puede comprobar, la rotación óptima alrededor del eje X es de 0°, aunque el sistema funciona hasta con inclinaciones de 50°. A medida que aumenta el ángulo de inclinación del dispositivo portable con respecto al plano, la precisión y la robustez del posicionamiento empeoran, incrementándose el número de puntos anómalos. La pérdida de robustez se justifica debido a que la inclinación de la cámara provoca que se capture la imagen de la baliza parcialmente y eso dificulta su identificación. Por otro lado, a mayor inclinación mayor distorsión de la imagen de la baliza, lo que empeora la calidad de la reconstrucción de la pose.

Tabla 6.1. Resultados de posicionamiento para diferentes rotaciones alrededor del eje X, en $10^{-3} m$.

Rotación X	0°	10°	20°	30°	40°	50°	60°
Error medio	0,36	7,78	12,56	10,74	9,24	7,32	-
Desviación típica	0,19	8,41	11,94	9,25	7,55	8,02	-
Error mínimo	0,01	1,26	2,03	2,50	0,16	0,16	-
Error máximo	0,91	47,26	49,28	42,57	49,95	49,95	-
Cobertura (%)	100	78,52	59,34	39,90	31,71	16,62	-

De forma análoga, en la figura 6.5 se presentan los gráficos generados a partir de las muestras tomadas modificando la rotación alrededor del eje Z en incrementos de 10 grados, manteniendo siempre la pantalla del dispositivo portable paralela con la superficie del techo, con la cámara mirando hacia arriba, dado que ésa es la rotación óptima alrededor del eje X, minimizando con ello la contribución al error proveniente de esta rotación.

A medida que la rotación alrededor del eje Z aumenta, también lo hacen los puntos de test en los que no ha sido posible realizar una estimación, disminuyendo la cobertura del sistema hasta alcanzar el mínimo (70,33 %) cuando la rotación alrededor del eje Z es de 80°. Como se puede comprobar en la figura 6.5, las zonas en las que existe cobertura parecen describir una forma similar a la que tendría el banco de pruebas si se rotase los mismos grados que se está rotando el dispositivo portable. Esto se debe a que, al rotar el receptor, el emisor aparece rotado en las muestras y, cuando se está cerca de los límites del FOV, parte del emisor puede estar fuera del FOV y capturarse de forma parcial, como puede verse en la figura 6.6, algo también aplicable a la rotación alrededor del eje X. Debido a que el algoritmo de estimación

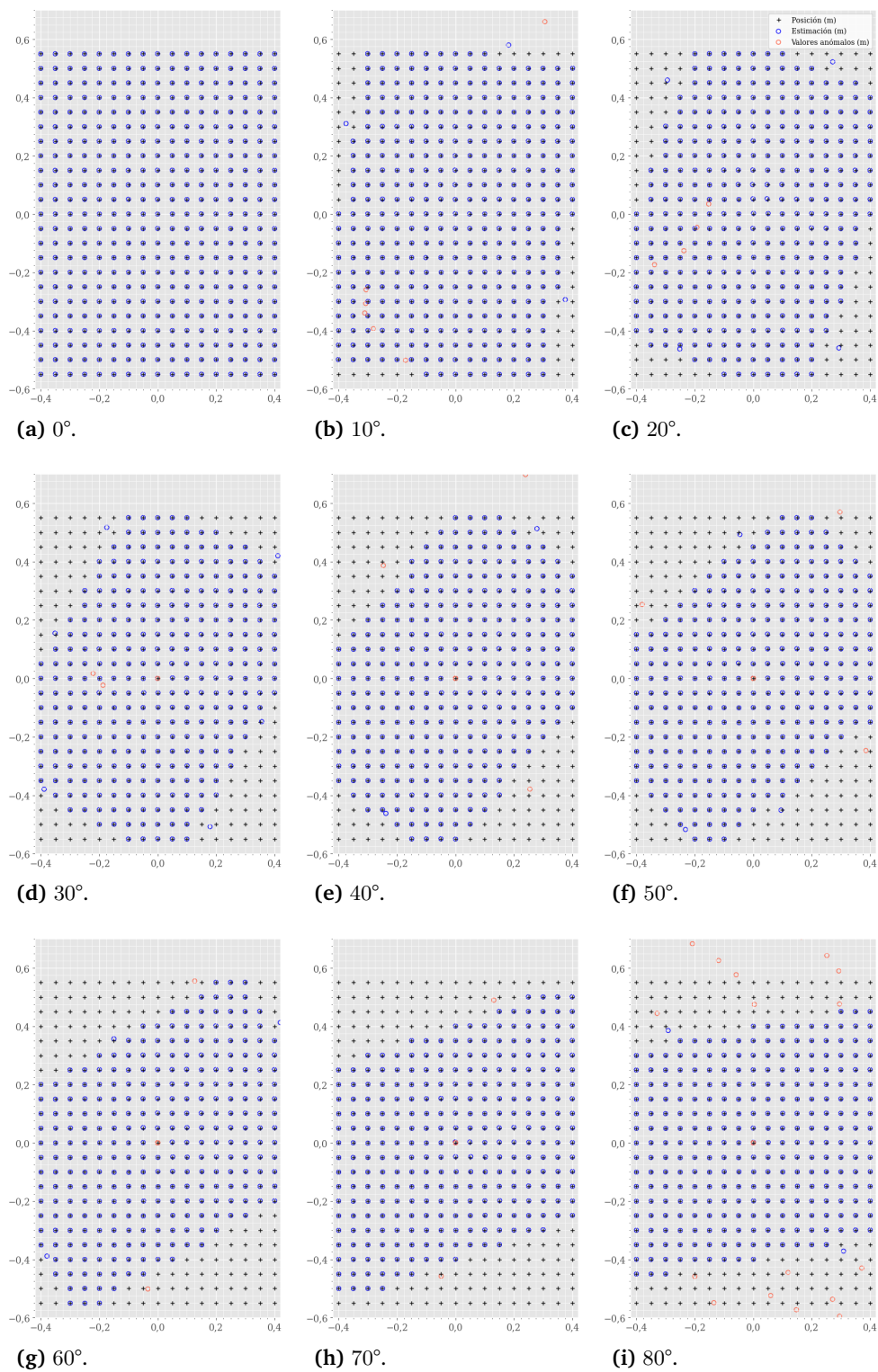


Figura 6.5. Resultados de posicionamiento en todo el FOV con diferentes rotaciones alrededor del eje Z.

de la posición del receptor recupera la pose de la cámara utilizando la geometría de la baliza, si la muestra no la contiene por completo, el error producido al estimar la posición será mayor cuanto más superficie de la baliza se pierda.



Figura 6.6. Captura parcial de la baliza al rotar alrededor del eje Z.

En la tabla 6.2 se pueden ver algunos parámetros que describen cuantitativamente la precisión en la estimación del posicionamiento en dos dimensiones, así como la cobertura obtenida para los ángulos de rotación mostrados en la figura 6.5. Se omite la rotación de 0°, ya que se trata de los mismo resultados que en la primera columna de la tabla 6.1. La rotación óptima alrededor del eje Z es de 0°, aunque el sistema funciona con cualquier rotación alrededor de dicho eje. Como era de esperar, el sistema es más resistente a las rotaciones alrededor del eje Z que a las del eje X.

Tabla 6.2. Resultados de posicionamiento para diferentes rotaciones alrededor del eje Z, en $10^{-3} m$.

Rotación Z	10°	20°	30°	40°	50°	60°	70°	80°
Error medio	0,81	1,16	1,17	1,08	1,36	1,23	0,89	0,81
Desviación típica	3,05	4,36	3,04	2,94	4,26	1,97	0,55	2,58
Error mínimo	0,00	0,04	0,03	0,01	0,12	0,11	0,05	0,03
Error máximo	43,82	44,69	30,01	39,61	45,13	23,80	3,36	36,32
Cobertura (%)	89,00	81,59	76,47	73,40	71,61	71,87	71,36	70,33

6.2. Banco de pruebas de tamaño reducido

Para validar los resultados obtenidos en el simulador se ha desarrollado un banco de pruebas de tamaño reducido que permita reproducir la granularidad y la precisión obtenida en el sistema modelado.

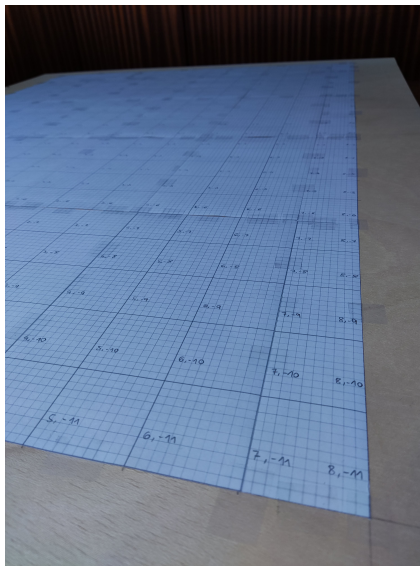
6.2.1. Descripción

El primer paso consiste en crear una rejilla similar a la que se muestra en el simulador, sobre la que se pueda situar el dispositivo portable con precisión. Para ello, se han impreso varias copias de una cuadrícula que, posteriormente, se ha recortado y adherido a una lámina de madera contrachapada hasta formar una rejilla de 0,8 m de ancho por 1,1 m de largo, de idénticas dimensiones al FOV de la simulación que aparece en la figura 6.1.

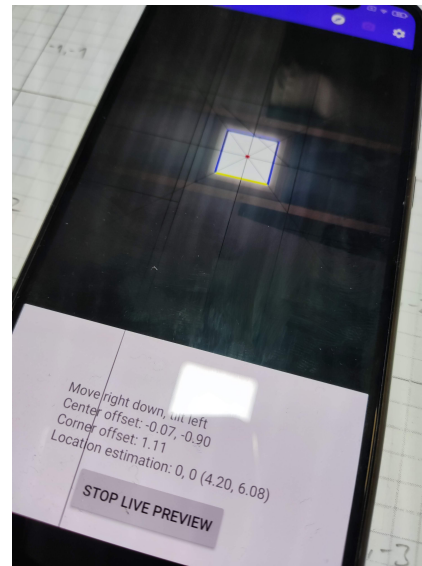
En la figura 6.7a se puede ver un detalle de la misma, donde cada celda está marcada con las coordenadas correspondientes, estando el origen de estas coordenadas en el centro de la cuadrícula. Cada celda tiene un lado de 0,05 m, con lo que se dispone de 391 posiciones. El eje X toma valores etiquetados, de izquierda a derecha, desde -8 hasta +8, mientras que el eje Y toma valores desde 11 hasta -11, de arriba hacia abajo.

Sobre esta rejilla, soportada por un par de caballetes, se encuentran dos mástiles a los que se ha anclado una baliza cuadrada de 0,174 m de lado, a 1 m de altura de la superficie del dispositivo portable, que a su vez se encuentra sobre la rejilla anteriormente descrita. El dispositivo portable se sitúa de tal manera que la cámara frontal se encuentra mirando al techo. Utilizando el modo de calibración de la aplicación de toma de muestras, descrito en la sección 4.5.4, se determina cuándo se encuentra alineado justo bajo el centro de la baliza. En la figura 6.7b se puede ver dicho modo. Una vez localizado este centro, la rejilla se ancla a la superficie para impedir que se desplace. En la figura 6.7c se puede ver el banco de pruebas de tamaño reducido, una vez montado. En este caso, puesto que sólo se está utilizando una baliza, ésta se configura para funcionar como una luminaria corriente. Por lo tanto, no emite ningún código.

El dispositivo portable se sitúa en cada punto de la rejilla, empezando por la esquina superior izquierda y acabando por la inferior derecha, tomando una muestra y avanzando al siguiente punto, hasta tomar las 391 muestras posibles. Así, la primera muestra se toma en la casilla con coordenadas $(-8, 11)$ y la última en $(8, -11)$.



(a) Detalle de la rejilla.



(b) Calibración de la rejilla.



(c) Banco de pruebas.

Figura 6.7. Banco de pruebas reducido.

Se han incorporado guías adicionales a la pantalla de la aplicación para facilitar la alineación del dispositivo portable con respecto a la rejilla, detalle que puede apreciarse en la figura 6.7b.

6.2.2. Resultados

En la figura 6.8 pueden verse los resultados de las pruebas de posicionamiento realizadas en este banco. Mientras que el simulador se utiliza para estudiar todas las posibles variantes y ahorrar trabajo, en la prueba de validación se comprueban las orientaciones estándar, que además han sido las que mejores resultados han proporcionado. Por lo tanto, solamente se están representado los datos respecto a la rotación de 0° respecto al eje X y el Z. De manera análoga a lo establecido en el simulador, se han considerado como valores anómalos aquellos cuyo error en dos dimensiones supera el margen de confianza, establecido como el lado de cada celda (0,05 m).

En la figura 6.9 se puede ver el error cometido en la estimación en cada posición del banco de pruebas en forma de mapa de calor. Como se puede apreciar, los mayores errores se encuentran en los extremos izquierdo y derecho de la rejilla, donde la baliza no se ha podido capturar por completo. Estos resultados confirman el impacto negativo que las capturas parciales de la baliza tienen en el rendimiento del sistema de posicionamiento.

De igual forma, existe una zona en el centro de la parte inferior del banco de pruebas donde el error es ligeramente superior al margen de confianza (0,7 m de máximo frente al margen de confianza de 0,5 m), por lo que las posiciones estimadas se marcan en rojo en la figura 6.8 y en azul más claro en la figura 6.9. Dicho error puede deberse a las imperfecciones existentes en el tablero de contrachapado sobre el que se ha montado la rejilla, que presenta una ligera curvatura.

La cobertura obtenida en el simulador con 0° de rotación X y Z (ver la figura 6.3a) es del 100 %. En este banco de pruebas de tamaño reducido, la cobertura obtenida en las mismas circunstancias es del 87,72 %. La mayor pérdida de cobertura se da en los extremos del banco de pruebas, donde la baliza no se ha podido capturar por completo al tomar las muestras correspondientes. La pequeña zona central de la parte inferior también influye en la cobertura. Queda de manifiesto que resulta imposible alcanzar el nivel de perfección de una simulación al preparar un banco de pruebas en el mundo real. La diferencia entre el simulador y la realidad en las muestras tomadas en la periferia se debe a las aberraciones en la lente del dispositivo

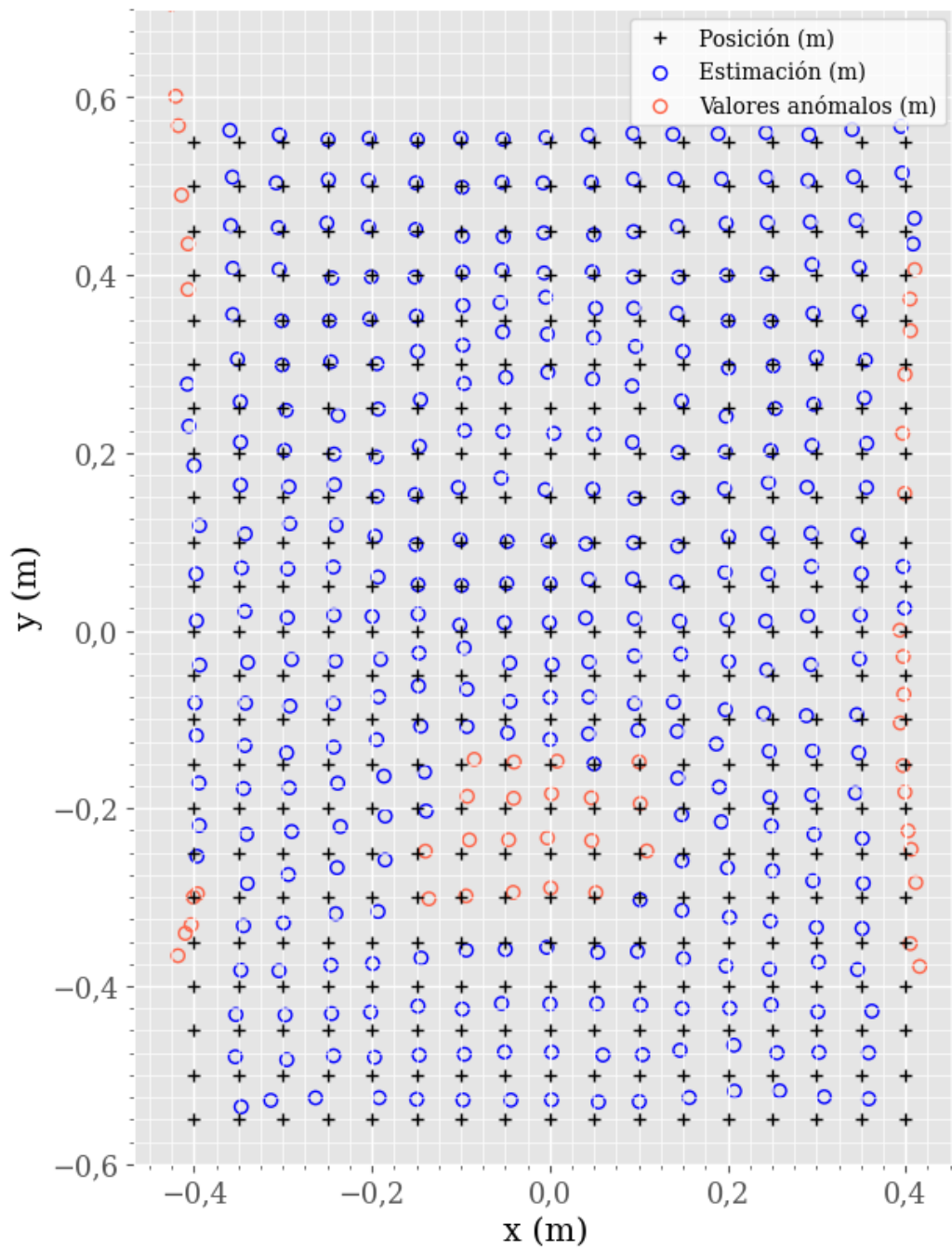


Figura 6.8. Estimación de la posición.

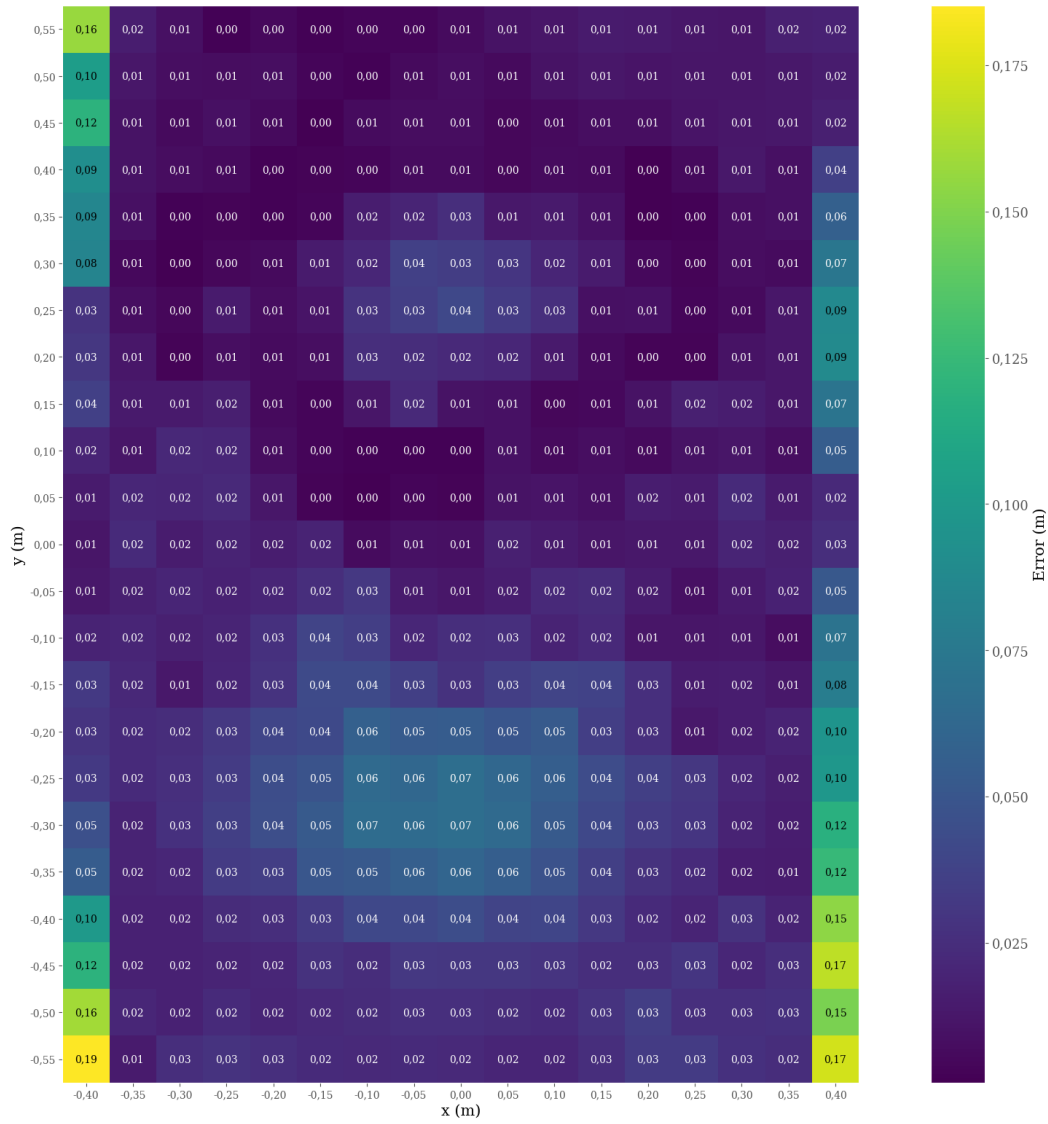


Figura 6.9. Mapa de calor de la estimación de la posición.

portable utilizado como receptor, que el simulador no modela. Incorporar este detalle al simulador es algo que se explorará en futuras evoluciones de este trabajo.

En la tabla 6.3 se proporcionan algunos parámetros que describen cuantitativamente la precisión en la estimación del posicionamiento en dos dimensiones y la cobertura conseguidos en esta prueba. Al igual que en las pruebas realizadas en el simulador, los valores mostrados aquí se han calculado sólo para aquellas muestras en las que el error en la estimación está por debajo del margen de confianza. El error medio es de 17,80 mm, logrando un 87,72 % de cobertura a lo largo de toda la rejilla de puntos de test. El 65,31 % de los valores está dentro del rango definido por la media \pm la desviación típica.

Tabla 6.3. Resultados de posicionamiento en el banco de pruebas reducido, en 10^{-3} m.

Error medio	17,80
Desviación típica	11,00
Error mínimo	1,11
Error máximo	49,96
Cobertura (%)	87,72

Los resultados obtenidos en la simulación quedan refrendados por los obtenidos en el banco de pruebas reducido, con lo que se procede a diseñar el banco de pruebas extendido.

6.3. Banco de pruebas extendido

6.3.1. Descripción

El grupo de investigación Quercus [176] dispone de la sala i3 [177] en la Escuela Politécnica de Cáceres [178], que ha cedido amablemente para preparar este banco de pruebas. Su aspecto puede verse en la figura 6.10. Se trata de una sala de 5,81 m de ancho y 7,24 m de largo, espacio suficiente para aumentar la escalabilidad del sistema y así probar su funcionamiento con varias balizas simultáneamente. El suelo de la sala está compuesto de baldosas de tarima flotante de 0,6 m de lado que pueden verse en la figura 6.12. Estas baldosas pueden servir de ayuda para preparar una rejilla de mayor granularidad de posiciones en las que tomar las muestras. Por otro lado, el techo se encuentra a 2,87 m de altura, muy cerca de la altura media

mencionada en la sección 3.8. Dicho techo está construido con placas de escayola, lo que facilita el montaje de cualquier elemento necesario para colocar las balizas.



Figura 6.10. Sala i3 antes de montar el banco de pruebas.

Antes de dibujar la rejilla de posiciones o colocar las balizas es necesario tener en cuenta los resultados del sistema de comunicación de datos y del sistema de posicionamiento, sin olvidar que se pretende utilizar los dos sistemas de forma conjunta.

De entre los diferentes sistemas de comunicación desarrollados, el de doble toma (descrito en la sección 3.7.4) tiene una ventaja muy importante con respecto a los demás. Como su nombre indica, el sistema de comunicación toma dos fotografías para optimizar el resultado: una con el código de barras y otra con el contorno de la baliza. La segunda, clave para mejorar el alcance del sistema de comunicación, se puede aprovechar para realizar el posicionamiento dentro del campo de cobertura de la baliza, como se describe en la sección 5.3. Este sistema de comunicación tiene un alcance óptimo de hasta 1,3 m de separación entre emisor y receptor. La altura de la sala y el alcance óptimo del sistema de doble toma encajan aceptablemente con las propiedades de la sala i3.

Utilizando el simulador descrito en la sección 4.8 se puede comprobar que, situando la cámara a una distancia de 1,3 m, el FOV de la cámara tiene una dimensiones de $0,6\text{ m} \times 1,2\text{ m}$. Aprovechando las baldosas de la sala, se trataría de una fila de dos baldosas, totalmente insuficiente para probar el sistema. Sin embargo, este FOV está calculado de manera que la baliza se vea por completo en todos los puntos

del mismo. Para este banco de pruebas final se quiere estudiar el funcionamiento del sistema de posicionamiento en circunstancias adversas como la captura de una baliza incompleta o de varias balizas en la misma toma. Por lo tanto, en lugar de utilizar el FOV ideal, se opta por utilizar el completo cuyas dimensiones son $1,29\text{ m} \times 1,72\text{ m}$. Si se aproxima utilizando las baldosas del suelo, se trataría de una zona de cobertura de $1,2\text{ m} \times 1,8\text{ m}$, es decir, 2×3 baldosas. En estas condiciones, si la cámara se situase a $1,3\text{ m}$ de distancia de la baliza y se llevase a cualquiera de los extremos del FOV, sólo media baliza sería visible. Para acercar aún más el banco de pruebas a un sistema real, se ha decidido que la cámara esté a $1,4\text{ m}$ distancia de la baliza, consiguiendo un doble objetivo. Por una parte, más superficie de la baliza se verá en los extremos, aunque seguirá sin verse completamente. Por otra, la distancia entre cámara y baliza estará por encima de la distancia óptima, con lo que se podrá exponer el sistema a situaciones en las que la decodificación del mensaje emitido por la baliza no sea óptima.

Para realizar las pruebas se decide montar cuatro balizas en el techo en una distribución de 2 por 2. Antes de realizar el montaje se vuelve a utilizar el simulador para previsualizar la disposición de la sala. Esta simulación, preparada de forma manual, puede verse en la figura 6.11. En ella, las celdas del suelo tienen un lado de $0,3\text{ m}$, la mitad del lado de las baldosas del suelo de la sala de pruebas, para poder obtener mayor granularidad dentro de la simulación. La superficie completa del banco de pruebas es de $2,4\text{ m}$ de ancho \times $3,6\text{ m}$ de largo. Si cada celda tiene $0,3\text{ m}$ de lado, el eje X tiene 5 posiciones, desde $-1,2\text{ m}$ hasta $1,2\text{ m}$ mientras que el eje Y tiene 7 posiciones, desde $-1,8\text{ m}$ hasta $1,8\text{ m}$.

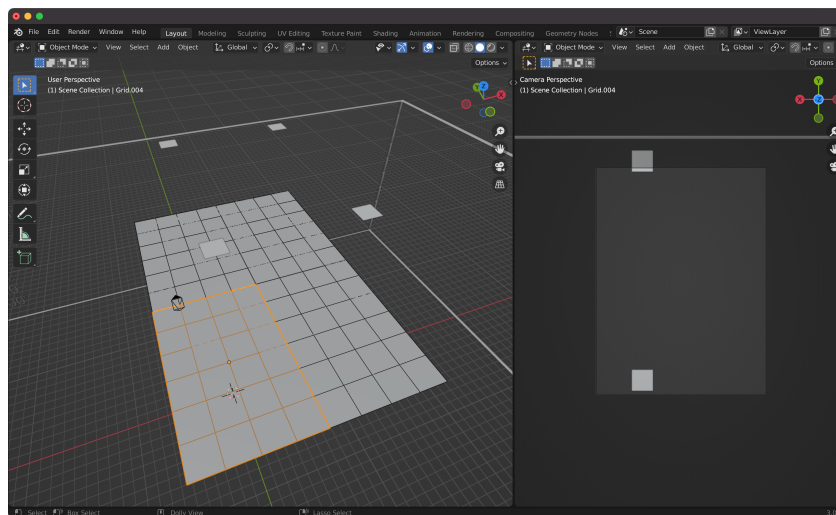


Figura 6.11. Simulación del banco de pruebas en la sala i3 antes del montaje.

Utilizando todos estos datos se está en disposición de montar el banco de pruebas. El suelo se marca utilizando las baldosas como referencia, dividiendo cada una de ellas en cuatro. Las balizas están ancladas al techo. El resultado final puede verse en la figura 6.12. En la misma figura, superpuesto, se puede ver el identificador que se ha asignado a cada una de ellas para diferenciarlas.



Figura 6.12. Montaje final del banco de pruebas en la sala i3.

Las balizas se han centrado utilizando una plomada láser, de manera que sus centros se corresponden con el centro de cada una de las cuatro zonas de cobertura del suelo. Así mismo, las balizas están niveladas para que su superficie sea paralela a

la superficie del suelo. No obstante, siempre se debe tener en consideración que pueden existir pequeños errores instrumentales en estas medidas experimentales.

Cada una de las cuatro balizas emite un código diferente, seleccionado entre los que mejores resultados ofrecieron en las pruebas de decodificación, que pueden verse en la figura 3.18 de la sección 3.7. En la tabla 6.4 se muestran los códigos seleccionados para cada baliza, en decimal y en binario.

Tabla 6.4. Código emitido por cada baliza.

Baliza	Código decimal	Código binario con cabecera
a	4	01110 11 01 00 11
b	6	01110 11 01 01 00
c	9	01110 10 11 00 10
d	10	01110 10 11 01 00

En la figura 6.12 se puede apreciar que, a pesar de que cada baliza emite un código diferente, no existen diferencias aparentes de luminosidad gracias a la elección de la codificación Manchester.

Una vez completado el banco de pruebas se procede a la toma de muestras. Para ello, el dispositivo portable se monta sobre un trípode a una distancia de 1,40 m de las balizas y 1,32 m del suelo, con la pantalla mirando al techo, puesto que se va a utilizar la cámara frontal para las pruebas. La distancia se comprueba utilizando un medidor láser con una precisión de $\pm 1,5$ mm [179]. Como se observó anteriormente, los mejores resultados se obtienen cuando la superficie de la cámara y de la baliza están paralelas, por lo que se ha utilizado un nivel para alinear la pantalla del dispositivo portable.

Una vez preparado el banco de pruebas, se comienza con la toma de muestras. Se utilizan 117 puntos de control, uno en cada esquina de las cuadrícula de 0,3 m preparada en el suelo. La cámara del dispositivo portable se alinea antes de la toma de muestras con los puntos del suelo utilizando una plomada, como puede verse en la figura 6.13. En el momento de tomar las muestras, la única iluminación artificial utilizada la proporcionan las propias balizas, suficiente para trabajar en la sala.

En cada uno de los puntos se efectúan 100 tomas. Dado que se está utilizando el algoritmo de doble toma, cada toma resulta en dos fotografías. En total, tras completar el proceso de toma de muestras, se dispone de 23 400 fotografías.



Figura 6.13. Proceso de toma de muestras.

6.3.2. Resultados

Para analizar los resultados obtenidos en el banco de pruebas de la sala i3 es necesario estudiar dos sistemas: el de comunicación de datos y el de posicionamiento.

Comunicación

En la figura 6.14 se puede ver el gráfico de disponibilidad del sistema de comunicación de datos. Cada celda del gráfico se corresponde con el porcentaje de aciertos en la detección del código emitido al tomar las muestras desde la posición correspondiente dentro de la rejilla de pruebas.

Como se puede comprobar, existe disponibilidad en todas las celdas del banco de pruebas, incluso en las que están en los extremos. Existen varios mínimos cercanos al 20% en las zonas cubiertas por las balizas **a**, **b** y **c**. La zona con peor disponibilidad es la cubierta por la baliza **a**, mientras que la cubierta por la baliza **d** muestra la mejor disponibilidad. Aunque se han escogido aquellos códigos que mejores resultados obtuvieron en las pruebas, el rendimiento de cada código del sistema de comunicación es diferente, como se demuestra en la sección 3.7.2. La media de la disponibilidad en la zona cubierta por cada baliza y la media total se puede ver en la tabla 6.5.

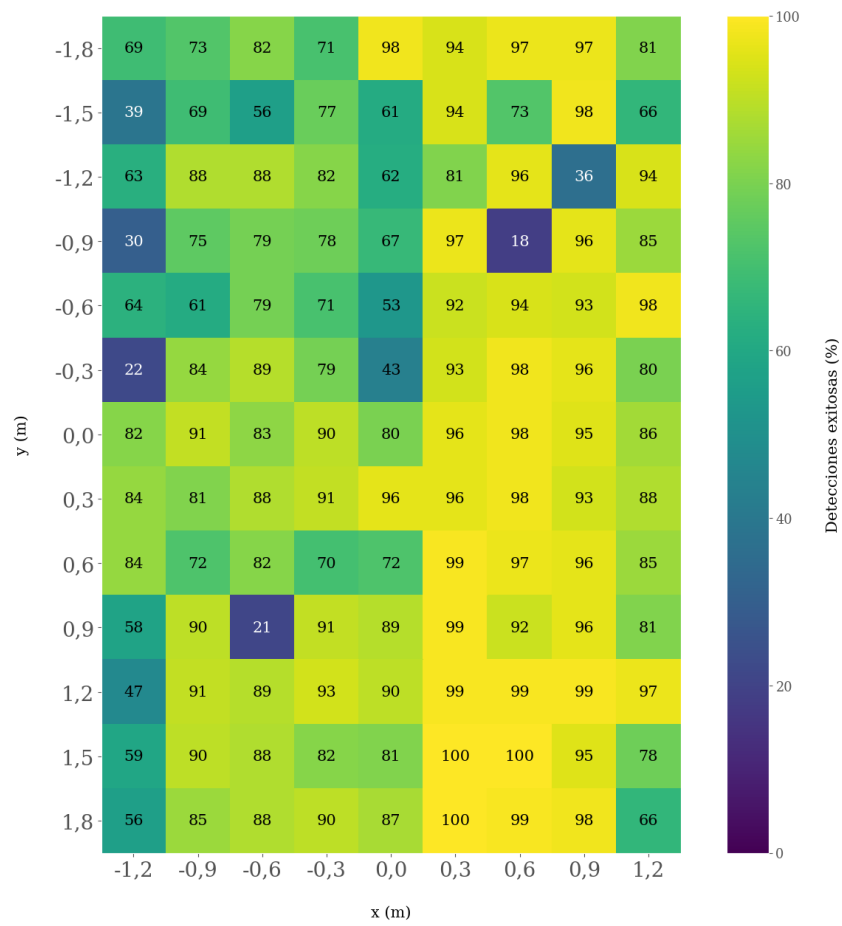


Figura 6.14. Disponibilidad del sistema de comunicación de datos.

Tabla 6.5. Porcentajes de disponibilidad en el banco de pruebas extendido para cada baliza y en total.

Zona	Disponibilidad (%)
a	71
b	82
c	80
d	92
Total	81

Existen varias zonas interesantes a tener en cuenta en los resultados mostrados en la figura 6.14. En primer lugar, las posiciones en la periferia del banco de pruebas que forman un rectángulo a su alrededor ($x = \pm 1,2$ m, $y = \pm 1,8$ m), donde las muestras tomadas no consiguen capturar la baliza en su totalidad. En la figura 6.15 se pueden observar las imágenes adquiridas en las cuatro esquinas del banco de pruebas. A pesar de que la baliza no se captura en su totalidad en estos extremos, el sistema es capaz de obtener el código emitido y, por tanto, determinar en qué zona se encuentra el receptor.

Por otra parte, la columna $x = 0,0$ m y la fila $y = 0,0$ m también resultan interesantes. En ellas, dos balizas son visibles a la vez. En el origen de coordenadas, las cuatro balizas son visibles de manera simultánea, como puede verse en la figura 6.15e. En ambos casos, ninguna de las balizas es capturada por completo. Cuando más de una baliza es visible, se escoge aquella con mayor superficie capturada. De nuevo, el sistema es capaz de detectar el código emitido en todas las celdas de esta zona, aunque parece tener más dificultades para ello en el límite entre las balizas **a** y **b**, donde el porcentaje de detecciones exitosas baja.

Tal y como se vio en el tercer capítulo (ver la sección 3.7.4), el sistema de comunicación de doble toma tiene un rendimiento óptimo a 1,3 m. En este banco de pruebas se ha querido comprobar el rendimiento del mismo situando el receptor a una distancia superior a dicho valor óptimo. Dado que es posible saber bajo qué baliza se encuentra el receptor en todas y cada una de las celdas del banco de pruebas, resulta adecuado continuar estimando la posición del dispositivo portable.

Posicionamiento

En la figura 6.16 se pueden ver las estimaciones de posición conseguidas en la zona cubierta por las cuatro balizas. Cada celda del gráfico se corresponde con la media

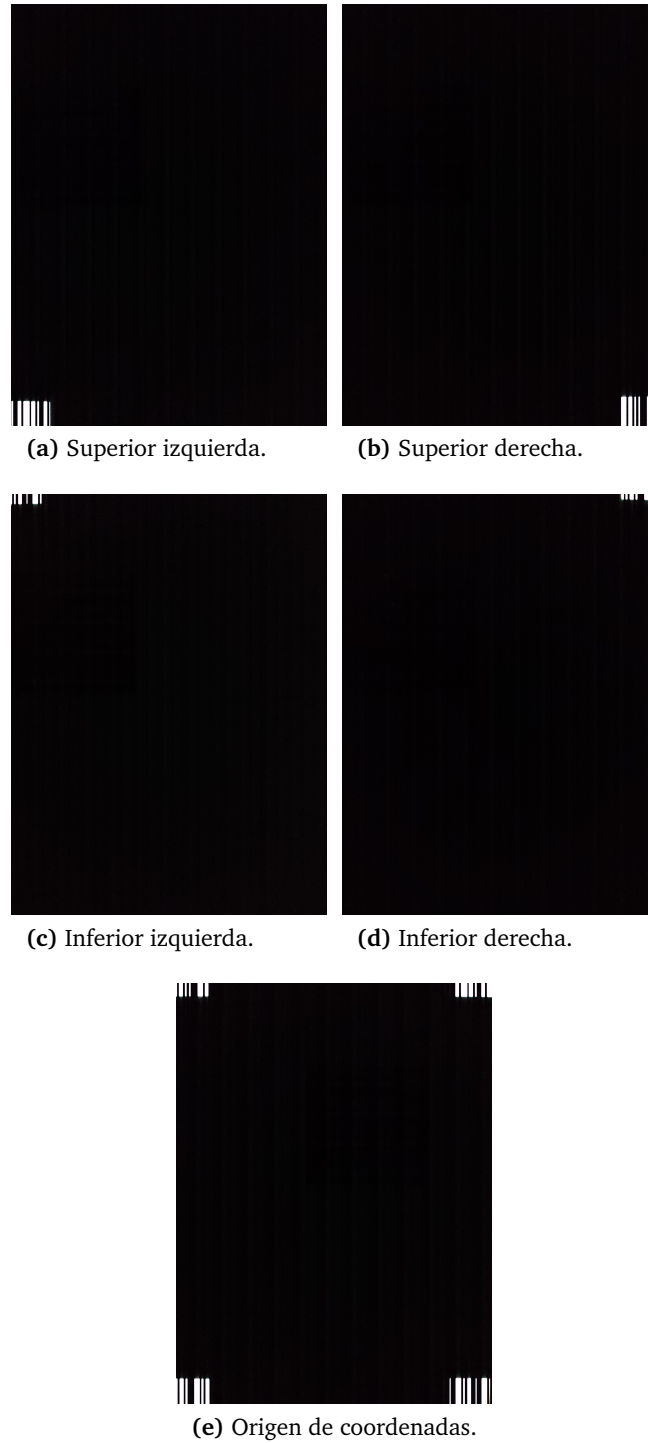


Figura 6.15. Muestras tomadas en las esquinas y el origen de coordenadas del banco de pruebas en la sala i3.

del error en dos dimensiones cometido al estimar la posición dentro de la rejilla de pruebas.

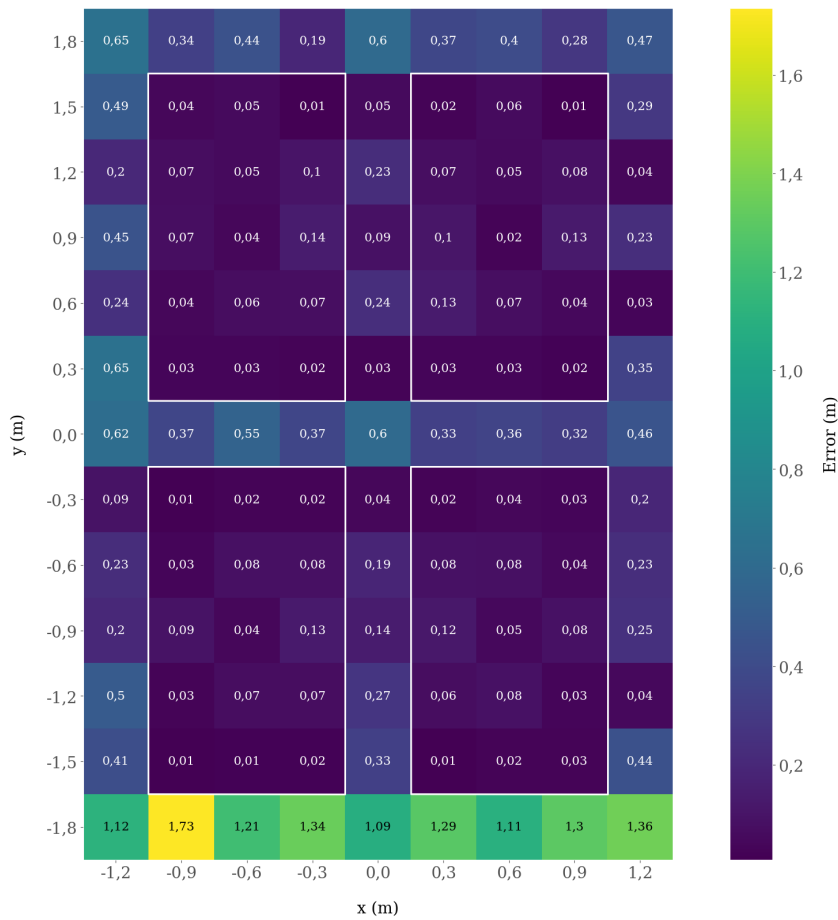


Figura 6.16. Cobertura del sistema de posicionamiento en la sala i3.

Como se puede comprobar, el error es menor (por debajo de los 15 cm) de forma consistente en las celdas centrales de las cuatro zonas (destacadas con un contorno blanco en la figura 6.16), donde la baliza se captura por completo. En la zona periférica de cobertura de cada baliza, sus imágenes aparecen recortadas a lo largo o a lo ancho, algo similar a lo que se se puede apreciar en la figura 6.15. Mientras que el sistema de comunicación de datos sigue funcionando con balizas parciales, el sistema de posicionamiento es mucho más sensible a este efecto. Esto confirma los resultados obtenidos en el simulador, en los que el error la estimación de la posición realizada a partir de muestras en las que la baliza aparecía de forma parcial era mayor que cuando la baliza aparecía de forma completa.

Al igual que en el análisis del sistema de comunicación, en el de posicionamiento también resulta interesante estudiar las zonas de las posiciones periféricas del banco de pruebas que forman un rectángulo a su alrededor ($x = \pm 1,2\text{ m}$, $y = \pm 1,8\text{ m}$).

En estos puntos, las balizas no se capturan de forma completa, lo que empeora el rendimiento del sistema de posicionamiento. Este efecto es aún más pronunciado en la fila de la parte inferior del banco de pruebas ($y = -1,8\text{ m}$) debido a que, como se puede comprobar en la figura 6.15, menos superficie de las balizas es visible desde esas posiciones. Este hecho pone de relieve la importancia que tiene capturar la mayor superficie de las balizas que sea posible para asegurar un funcionamiento óptimo del sistema de posicionamiento. Los resultados obtenidos en la columna $x = 0,0\text{ m}$, la fila $y = 0,0\text{ m}$ y el origen de coordenadas también sufren del mismo problema que en los extremos: la superficie de las balizas aparece recortada, lo que empeora el rendimiento del sistema de posicionamiento.

En la figura 6.17 se pueden ver los resultados de posicionamiento. Se han considerado como valores anómalos aquellos cuyo error en dos dimensiones supera el margen de confianza, establecido como el lado de cada celda ($0,30\text{ m}$). Las cuatro zonas destacadas con un contorno blanco en la figura 6.16, en las que se obtenían los mejores resultados, aparecen aquí destacadas en diferentes colores, pertenecientes a la zona de cobertura de cada baliza (a, b, c y d, de izquierda a derecha y de arriba a abajo). De nuevo, queda de manifiesto que las zonas con mejores estimaciones son aquellas en las que la baliza se captura por completo.

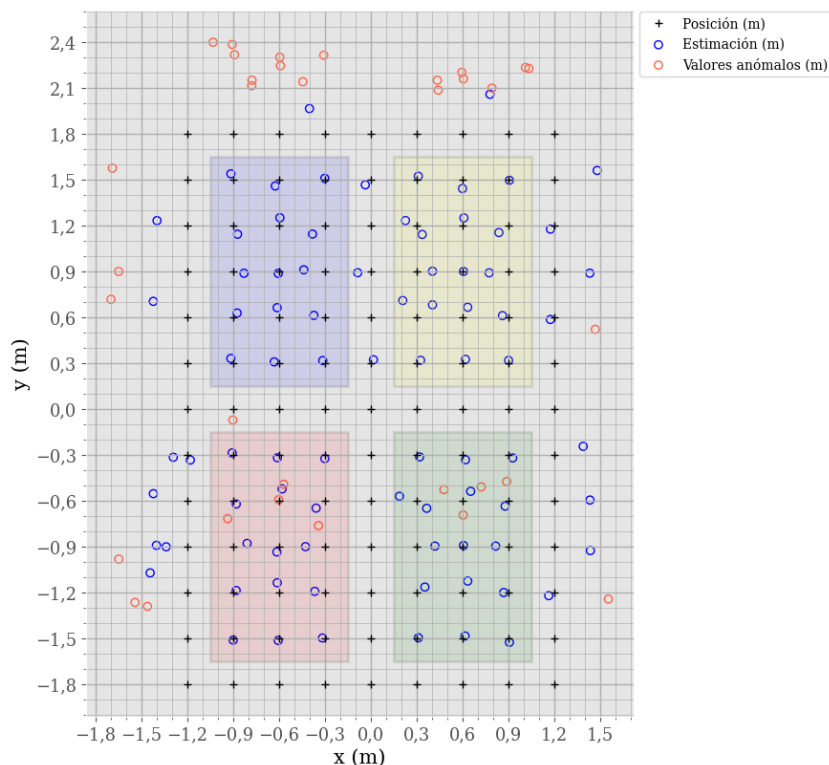


Figura 6.17. Estimación de las posiciones en la sala i3.

Algunos parámetros que describen cuantitativamente la precisión en la estimación del posicionamiento y la cobertura conseguidos en esta prueba, sólo para aquellas muestras en las que el error en la estimación está por debajo del margen de confianza, se pueden consultar en la tabla 6.6. Al igual que en las pruebas realizadas en el simulador y en el banco de pruebas de tamaño reducido, los valores mostrados aquí se han calculado sólo para aquellas muestras en las que el error en la estimación está por debajo del margen de confianza (el lado de la celda, 0,30 m en este caso). El error medio es de 85,92 mm, logrando un 71,79 % de cobertura a lo largo de todo banco de pruebas. El 79,76 % de los valores está dentro del rango definido por la media \pm la desviación típica.

Tabla 6.6. Resultados de posicionamiento en la sala i3.

Error medio (10^{-3} m)	85,92
Desviación típica (10^{-3} m)	76,03
Error mínimo (10^{-3} m)	7,88
Error máximo (10^{-3} m)	285,56
Cobertura (%)	71,79

Estos mismos resultados, pero sólo en las zonas centrales cubiertas por cada baliza, pueden verse en la tabla 6.7.

Tabla 6.7. Resultados de posicionamiento en las zonas centrales de en la sala i3, en metros.

Zona	Error medio	Desviación típica	Error mínimo	Error máximo
a	0,05	0,03	0,01	0,14
b	0,06	0,04	0,01	0,13
c	0,05	0,04	0,01	0,13
d	0,05	0,03	0,01	0,12

Con los resultados de esta prueba en la sala i3 se refrendan los de la simulación y el banco de pruebas reducido. Se dispone, por tanto, de un sistema de comunicación y posicionamiento fiable. Sólo resta realizar una prueba en la que el dispositivo portable esté sostenido en la mano por una persona mientras se mueve por el banco de pruebas.

6.4. Prueba de movimiento

6.4.1. Descripción

Para concluir con las pruebas, se han realizado dos recorridos por la rejilla de la sala i3. El primero de ellos, en espiral rectangular convergente, partiendo desde la esquina inferior izquierda, bajo la baliza **c**, siguiendo hacia la esquina superior izquierda, bajo la baliza **a**. El segundo recorrido, en zigzag, comienza en la esquina superior izquierda y avanza hacia la zona bajo la baliza **b**. La ruta esperada, en ambos casos, está marcada en naranja en los resultados, como referencia. En estas pruebas, el dispositivo portable siempre está apuntando al norte, puesto que aún no se ha comenzado a fusionar la información procedente de otros sensores (como en magnetómetro) en los datos obtenidos al tomar las muestras.

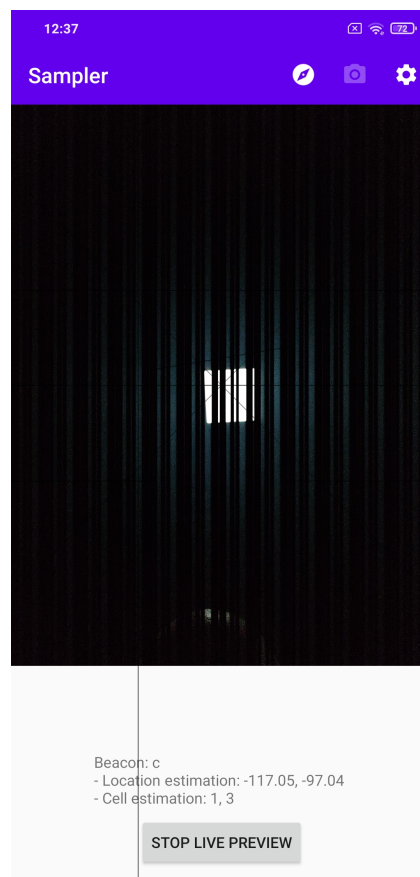
Durante las pruebas, el dispositivo portable muestra en pantalla el nombre de la baliza bajo la que se encuentra y las coordenadas de la posición estimada. A la vez, esos datos se guardan en un archivo **CSV** para su posterior análisis. La aplicación es capaz de tomar las dos muestras, obtener el código de la baliza y estimar la posición en ventanas de 1 sec. El teléfono está sostenido en la mano del usuario a una altura aproximada de 1,4 m, al igual en la anterior prueba realizada en el mismo banco. La pantalla del smartphone se sostiene intentando que esté paralela al techo, y tratando de evitar que la cabeza del usuario interfiera en la baliza al tomar las muestras. El sujeto se desplaza media celda (0,15 m) en cada paso, dejando aproximadamente unos 5 sec entre paso y paso. Este proceso se repite hasta que el recorrido se completa. En la figura 6.18 se puede ver al sujeto mientras obtiene las muestras, así como la pantalla de la aplicación.

6.4.2. Resultados

En la figura 6.19 se puede observar el resultado de la prueba en movimiento describiendo una espiral rectangular convergente que parte de la posición $x = -0,9$ m, $y = -1,5$ m y termina en la posición $x = 0,3$ m, $y = -0,9$ m. Destacado en naranja, el camino que se ha seguido al realizar la prueba. Cada una de las zonas de cobertura de las diferentes balizas está destacada con un tono diferente (de izquierda a derecha y de arriba a abajo, azul, amarillo, rojo y verde para las balizas **a**, **b**, **c** y **d**, respectivamente). La posición estimada para cada muestra se representa con un punto en las coordenadas obtenidas, con un color que se corresponde con el de la baliza detectada.



(a) Sujeto tomando muestras.



(b) Pantalla de la aplicación.

Figura 6.18. Toma de muestras en movimiento.

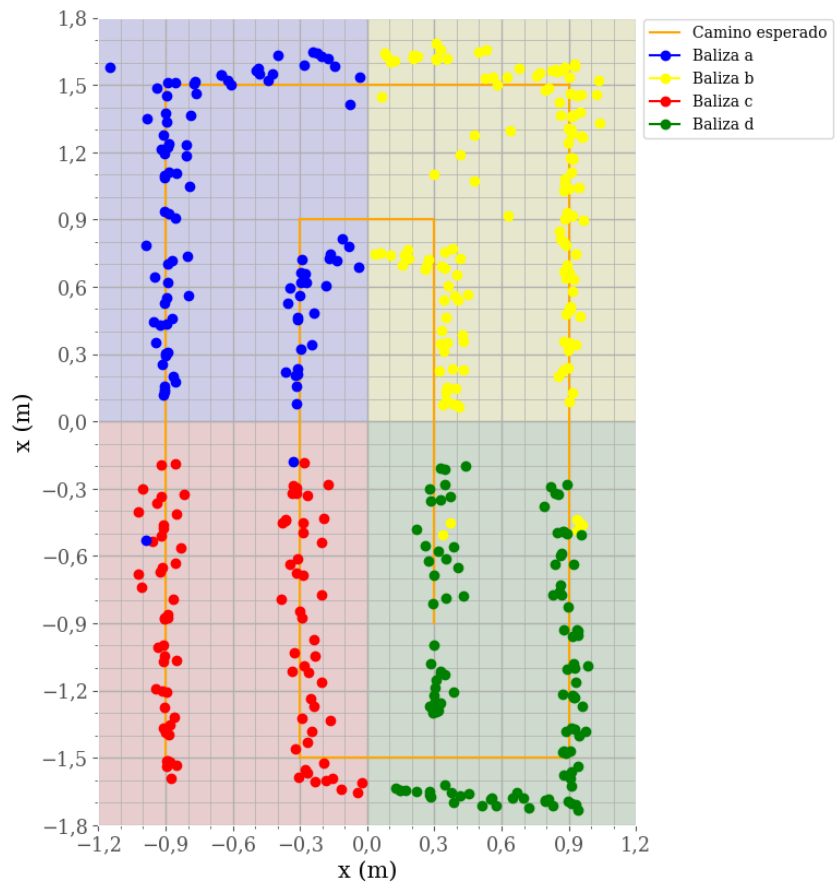


Figura 6.19. Estimación de las posiciones durante movimiento en espiral en la sala i3.

Los resultados obtenidos en las zonas de cobertura de las cuatro balizas son similares, correspondiéndose con el camino seguido. Existe una franja donde no se obtienen resultados, a lo largo del eje Y. En esa zona no es posible capturar una baliza de forma completa y el rendimiento del sistema de posicionamiento se ve afectado, confirmando lo expuesto en las anteriores pruebas realizadas en la sala i3. Los puntos de la baliza **a** (azules) que aparecen en la zona de cobertura de la baliza **c** (roja) son consecuencia de estimaciones con un error de posicionamiento alto dentro de esta franja, debido a que en ellos la baliza no se ha capturado por completo. Lo mismo ocurre en los puntos de la baliza **b** (amarilla) que aparecen en la zona de cobertura de la baliza **d** (verde).

En la figura 6.20 se puede observar el resultado de la prueba en movimiento en zigzag partiendo de la posición $x = -0,9$ m, $y = 1,5$ m y terminando en la posición $x = -0,9$ m, $y = -1,5$ m. La ruta esperada, las zonas de cobertura de cada baliza y las posiciones estimadas se muestran siguiendo las mismas pautas que en la figura 6.19.

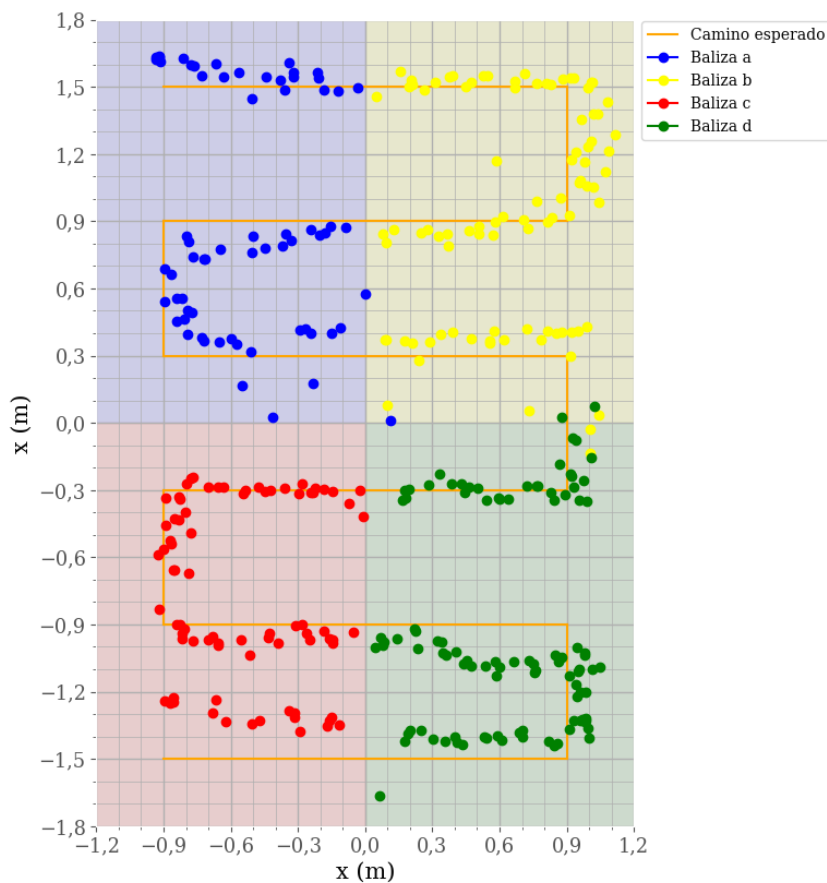


Figura 6.20. Estimación de las posiciones durante movimiento en zigzag en la sala i3.

Como se puede comprobar, la ruta estimada se corresponde con la esperada, ofreciendo de nuevo resultados con mayor desviación de la estimación en las zonas en las que la baliza no se captura por completo, como las zonas de la periferia o centrales.

En ambos casos, la ruta obtenida es consistente con la ruta esperada, lo que permite concluir que el sistema de posicionamiento desarrollado cumple con las expectativas, constituyendo una sólida base sobre la que construir futuros trabajos.

Conclusiones y trabajos futuros

Con este capítulo se cierra el trabajo de tesis. En él se presentan las principales conclusiones que se pueden extraer del mismo, se exponen las líneas de investigación futuras y se listan las publicaciones directamente relacionadas con la tesis.

7.1. Conclusiones

En este trabajo de tesis se ha desarrollado un sistema de posicionamiento en interiores basado en luz visible para dispositivos portables, con una precisión adecuada. Para lograrlo se ha apoyado en dos tecnologías ampliamente extendidas en la actualidad: las luminarias **LED** y los dispositivos portables.

El sistema de comunicación de datos seleccionado utiliza un panel **LED** gestionado por un microcontrolador como emisor y la cámara de un dispositivo portable como receptor. Se ha seleccionado una codificación Manchester que, aunque limita el número de bits que se pueden transmitir, garantiza que la luminosidad relativa de las balizas existentes en la sala sea la misma, permitiendo aprovechar la infraestructura de iluminación existente. El sistema es capaz de emitir 4 bits e identificar 16 balizas diferentes. El alcance del sistema desarrollado es de 1,7 m con un 100 % de detecciones exitosas, que decae paulatinamente hasta que no hay detecciones a los 2,4 m de distancia. Aunque se han desarrollado varios sistemas de comunicación de datos, el seleccionado es el de doble toma, que permite optimizar el alcance y la disponibilidad del sistema a la vez que proporciona los elementos necesarios para realizar la estimación de la posición.

Por otra parte, el sistema de posicionamiento se basa en la reconstrucción de la pose de la cámara del dispositivo portable utilizado como receptor. Para ello es necesario conocer las características de la cámara (dimensiones de las muestras tomadas, tamaño del píxel y distancia focal), así como la geometría de las balizas utilizadas (en el caso de este trabajo de tesis, al ser balizas cuadradas, la longitud del lado). A

partir de esos datos, utilizando el algoritmo SQPnP, resulta posible estimar la posición desde la que se tomaron las muestras.

Este sistema de posicionamiento se ha caracterizado en un simulador, y en dos bancos de pruebas reales: uno de tamaño reducido con una única baliza, y otro en una sala de pruebas de mayor extensión con cuatro balizas emitiendo códigos diferentes. En la simulación, los mejores resultados se consiguen cuando no existe rotación alrededor de los ejes X y Z, logrando un 100 % de cobertura cuando el receptor está a 1 m de distancia del emisor. A medida que se rota el receptor, la cobertura disminuye, siendo más sensible a las rotaciones alrededor del eje X que a las del eje Z. A partir de 60° de rotación alrededor del eje X el sistema deja de ser viable al no existir visibilidad de las balizas, ubicadas en el techo. En cuanto a la rotación alrededor del eje Z, la cobertura mínima sin rotación en el eje X es del 70 %. El error medio máximo en este simulador ronda los 10 mm al rotar el receptor alrededor del eje X. Al rotar alrededor del eje Z, el error medio está por debajo de 1,5 mm. En estos dos bancos de pruebas se han considerado anómalos todos aquellos valores con un error de posicionamiento por encima del lado de la celda utilizada, que mide 0,05 m.

En el banco de pruebas reducido, sin rotaciones alrededor de ningún eje, el porcentaje de cobertura pasa a un 87,72 %. El error medio es de 17,80 mm. En el banco de pruebas completo, con cuatro balizas, la cobertura es del 71,79 % y el error medio 85,92 mm.

El sistema de comunicación de datos funciona incluso si la baliza no se captura por completo, basta con que una porción de la misma esté en la muestra capturada para que se puedan obtener datos. Sin embargo, el sistema de posicionamiento es muy sensible a esta misma circunstancia, lo que se pone de manifiesto en las pruebas realizadas en el simulador y en el mundo real. De hecho, una de las razones por las que las rotaciones alrededor de los ejes X y Z afectan a la cobertura es la captura incompleta de las balizas.

A lo largo de este trabajo se han diseñado y construido unas balizas con una PCB propia en la que se ha montado un microcontrolador, además de otros componentes adicionales. Una vez desplegadas, resulta posible programar las balizas a través de una red Wi-Fi, lo que facilita su mantenimiento. Además de las balizas, se han desarrollado varias herramientas software para probar emisor y receptor, para simular las condiciones de una sala de pruebas, o para tomar muestras y analizarlas. Varias partes de estos desarrollos se han compartido con la comunidad al incluirlos GetSensorData, una aplicación ampliamente utilizada en concursos de posicionamiento para la captura de datos de los sensores de dispositivos portables Android.

Finalmente, podemos concluir que se ha conseguido implementar un sistema de localización en interiores que cumple los objetivos propuestos en este trabajo de tesis. Las limitaciones del ojo humano, junto con la implantación de las luminarias tipo **LED** y el uso de cámaras **CMOS** han permitido construir un sistema de comunicación sobre la infraestructura existente, imperceptible para las personas. De las dos muestras tomadas por sistema de doble toma en la comunicación de datos, una de ellas se aprovecha para las tareas de posicionamiento, optimizando así los recursos disponibles. El dispositivo portable utilizado en las pruebas es capaz de tomar estas dos muestras y procesarlas para obtener el código emitido y estimar la posición en menos de 1 sec, mientras el dispositivo sigue respondiendo a la interacción del usuario. Se han sentado las bases necesarias para seguir avanzando en esta línea de investigación.

7.2. Trabajos futuros

Al desarrollar los diferentes elementos que conforman el sistema de posicionamiento descrito en este trabajo, se han identificado algunos aspectos susceptibles de mejora. Dichas mejoras no se limitan a lograr un mejor posicionamiento, sino que también se busca facilitar el trabajo del investigador, el despliegue de las balizas o el rendimiento del receptor, por mencionar algunas de ellas.

7.2.1. Mejoras en el sistema de comunicación

El sistema de comunicación de datos presentado en este trabajo es capaz de transmitir 4 bits, lo que permite trabajar con 16 balizas diferentes dentro de una misma sala. Aumentar el número de bits emitidos resultaría muy útil, ya que así se podría aumentar la escalabilidad del sistema e incluso optar por emitir información adicional asociada a cada una de las balizas. Ese objetivo se podría abordar por diversas vías como, por ejemplo, trabajar con vídeo en lugar de con fotografías, concatenar diferentes tomas, o investigar sistemas de codificación alternativos.

Lograr establecer una sincronización entre el emisor y receptor también mejoraría el sistema de comunicación. Si el receptor fuese capaz de tomar las muestras en el mismo momento en el que el emisor terminase de emitir su código, se conseguiría garantizar que la cabecera estuviese siempre al comienzo de la muestra, optimizando así el alcance del sistema. Para ello sería necesario estudiar a fondo el sistema de

gestión de la cámara de los dispositivos portables, así como utilizar un reloj común tanto en emisores como en receptores.

En la sección 3.7.3 se presenta el sistema de decodificación basado en toma de ráfagas. Cuanto mayor sea el número de fotografías de cada toma de datos, mejor es el rendimiento del sistema de decodificación. Sin embargo, el tiempo de procesamiento aumenta sensiblemente, limitando el desempeño del sistema. Encontrar una relación entre la frecuencia de emisión, la frecuencia de captura del sensor óptico en el receptor y la velocidad a la que se toman las ráfagas permitiría optimizar el número muestras tomadas, de manera que con un número más reducido de ellas se garantizaría un mayor éxito en el proceso de decodificación.

Por otro lado, determinar si existe una relación entre el tamaño de las muestras (ancho y alto de las mismas) y el alcance del sistema permitiría calcular el tamaño óptimo de las mismas con el que se puede mantener el porcentaje de aciertos. Cuanto menor sea el tamaño de las muestras, menor será el tiempo que se tardará en procesarlas.

7.2.2. Mejoras en la baliza

Las balizas utilizadas en este trabajo están basadas en paneles LED. En el momento del despliegue del sistema, lo ideal sería diseñar un módulo externo que pudiese acoplarse a paneles ya existentes. Podría tratarse de un componente que sustituyese el *driver* que acompaña a estos paneles. Este módulo debería ofrecer las mismas características de las balizas actuales como, por ejemplo, ser programable a través de una interfaz web. Todos los drivers de los paneles LED domésticos tienen la misma entrada (220 V de corriente alterna) pero no la misma salida de corriente continua, que depende del tamaño del panel. Este módulo permitiría aprovechar por completo los paneles LED existentes, sin necesidad de sustituirlos por paneles diseñados para el posicionamiento en interiores.

Por otra parte, durante la realización de las pruebas del sistema, una de las principales dificultades está relacionada con la calibración. Ha sido necesario utilizar niveles y plomadas (tanto convencionales como láser) para garantizar que las balizas y los receptores estaban alineados con la mayor fiabilidad posible, y para situar el banco de pruebas con la baliza en el centro. La siguiente evolución de la baliza para el desarrollo del sistema de posicionamiento podría incluir un emisor láser en centro exacto de superficie luminosa para saber dónde está el origen de coordenadas. Ese láser debe poder controlarse desde la interfaz web. Además, podría incluir un

inclinómetro para conocer la inclinación de la baliza y poder situarla perfectamente horizontal, también controlable a través de la interfaz web o, al menos, conocer ese valor de inclinación y poder utilizarlo para corregir la estimación de la posición. Por último, podría incluir un magnetómetro con el que conocer el ángulo de desfase con respecto al norte magnético, también para realizar correcciones en la estimación de la posición.

La baliza actual es susceptible de mejoras en su diseño: es necesario desmontarla para actualizar su *firmware*. Resultaría muy práctico colocar un conector micro-USB para programar el microcontrolador, así como algún interruptor que permitiese apagarla y encenderla con facilidad, o incluso de forma remota mediante IoT.

Por último, las balizas deben controlarse de forma independiente, aunque las presentes en cada sala forman parte de una unidad. Resultaría muy adecuado ofrecer un panel de control unificado que permitiese acceder a las características de todas las balizas. Integrar este panel con los datos recopilados por la aplicación móvil encargada del posicionamiento podría abrir la puerta a estudios muy interesantes como, por ejemplo, control de la afluencia o tráfico en las proximidades de cada baliza.

7.2.3. Mejoras en el simulador

El simulador desarrollado en Python para Blender que se presenta en la sección 4.8 ha resultado muy útil para alcanzar los objetivos de este trabajo. Sin embargo, como evidencian los resultados presentados en la sección 6.2, existe margen de mejora para modelar algunos detalles de la realidad, ya que se están ignorando aspectos como las posibles aberraciones presentadas en las lentes de las cámaras de los dispositivos portables o la presencia de contaminación lumínica procedente de otras fuentes. Cuanto más próxima esté la simulación a la realidad, más fiables serán los resultados simulados, permitiendo acelerar el estudio del sistema.

En su estado actual, el simulador se utiliza para ver el aspecto de una sala con una baliza y determinar la zona de cobertura ideal conseguida. Entonces, se pueden describir diferentes movimientos de cámara y tomar muestras que, posteriormente, se analizan. Incorporar la estimación en tiempo real de la posición al propio simulador permitiría comparar con facilidad el rendimiento de diferentes algoritmos de posicionamiento, ahorrando tiempo y esfuerzo.

Por último, el simulador sólo permite trabajar con una baliza. Añadir múltiples balizas a un entorno simulado y probar las diferentes combinaciones posibles de

ubicación para cubrir diferentes superficies permitiría avanzar aún más en el sistema de posicionamiento en interiores desarrollado en este trabajo.

7.2.4. Mejoras en el sistema de posicionamiento

Una de las limitaciones del sistema de posicionamiento que se observaron en la sección 6.2.2 y la sección 6.3.2 es la disminución del rendimiento en cualquier zona en la que la superficie de la baliza no se capture por completo como, por ejemplo, en las zonas periféricas. Sin embargo, dado que se conoce la geometría de la superficie de la baliza, se podría intentar estimar la superficie completa a partir de los datos conocidos de la misma y comparar los resultados obtenidos tras esta estimación con los esperados.

De nuevo, aprovechando la geometría de la superficie luminosa de la baliza y su posible funcionamiento utilizando luminarias con formas más complejas que las cuadradas, permitiría aprovechar en mayor medida las infraestructuras luminosas ya existentes.

Por último, el sistema de posicionamiento se implementa en dispositivos portables, que tienen una gran capacidad de procesamiento y recogen un gran volumen de datos sensoriales. Aumentar la precisión del sistema fusionando todos los datos disponibles sería una opción a tener en cuenta.

7.2.5. Mejoras en el receptor

Aunque aumentar el rendimiento del proceso de decodificación y posicionamiento no resulta sencillo, se trata de un paso clave para conseguir mejorar el sistema de localización. En primer lugar, porque así se reduce la latencia del proceso. En segundo lugar, porque consume menos recursos del dispositivo portable. Esta mejora pasa por estudiar diferentes aspectos del proceso, desde la captura de la imagen hasta los diferentes pasos de tratamiento de la misma, sin olvidar el desarrollo de técnicas específicas para optimizar el sistema.

Para terminar, es necesario ampliar el número de dispositivos en los que el sistema de posicionamiento se puede probar para extraer la mayor cantidad de datos sobre su rendimiento. Se propone el desarrollo de paquetes de software que otros desarrolladores puedan incluir con facilidad en aplicaciones existentes para dotarlas con las capacidades de posicionamiento presentadas en este trabajo. Posteriormente, los datos obtenidos a través de estas nuevas aplicaciones, funcionando en variados

dispositivos portables, se podrían utilizar para mejorar el sistema de posicionamiento.

7.3. Publicaciones y actividades relacionadas con la tesis

A lo largo del desarrollo de este trabajo ha sido posible compartir los resultados obtenidos en diferentes publicaciones y actuaciones, que se listan a continuación.

7.3.1. Publicaciones en revistas internacionales

- J. D. Gutiérrez, A. R. Jiménez, F. Seco, F. J. Álvarez, T. Aguilera y F. Melchor, “GetSensorData: An Extensible Android-based Application for Multi-Sensor Data Registration”, SoftwareX, 2022. **En evaluación [180]**.
- J. D. Gutierrez, T. Aguilera, F. J. Álvarez, y F. J. Aranda, “Precision Local Positioning by Mobile Device Recognition of Coded Visible Light Beacons”, IEEE Transactions on Instrumentation and Measurement (TIM). **Pendiente de envío al término del congreso I2MTC 2022.**

7.3.2. Publicaciones en congresos internacionales

- J. D. Gutierrez, T. Aguilera, F. J. Álvarez, y F. J. Aranda, “A Blender-based Simulation Tool for Visible Light Positioning with Portable Devices”, en International Instrumentation and Measurement Technology Conference (I2MTC), p. 6. Mayo 2022, Ottawa (Canadá) [181].
- J. D. Gutiérrez, F. J. Aranda, T. Aguilera, y F. J. Álvarez, “LIMUS: Exploration of Technological Prototypes for Location-Based Services in Museums”, en International Conference on Indoor Positioning and Indoor Navigation (IPIN). Octubre 2019, Pisa (Italia) [110].
- J. D. Gutiérrez, F. J. Álvarez, T. Aguilera, J. A. Paredes, y J. Morera, “Visible Light Positioning for Smartphones Based on Biphase Mark Coding: A Proof of Concept”, en International Conference on Indoor Positioning and Indoor Navigation (IPIN), p. 4. Septiembre 2018, Nantes (Francia) [105].

7.3.3. Publicaciones en congresos nacionales

- F. J. Álvarez, T. Aguilera, J. D. Gutiérrez, F. J. Aranda, y J. A. Moreno, “Identificación de marcas en entornos museísticos: comparación de tecnologías acústica, Bluetooth de baja energía y luz visible”, en 27º Seminario Anual de Automática, Electrónica Industrial y Automática (SAAEI). Septiembre 2020, Ciudad Real (España) [182].
- J. D. Gutiérrez, F. J. Álvarez, T. Aguilera, J. Morera, y J. A. Moreno, “LIMUS: una guía inteligente para espacios museísticos basada en el uso de dispositivos portables”, en XXV Seminario Anual de Automática, Electrónica Industrial e Instrumentación (SAAEI). Julio 2018, Barcelona (España) [183].

Bibliografía

- [1] T. Harford, «Cautionary Tales Ep 8 – You Have Reached Your Destination», *Tim Harford*, dic. 2019 [Online]. Disponible en: <http://timharford.com/2019/12/cautionary-tales-ep-8-you-have-reached-your-desination/>
- [2] S. Nirjon, J. Liu, G. DeJean, B. Priyantha, Y. Jin, y T. Hart, «COIN-GPS: Indoor Localization from Direct GPS Receiving», en *Proceedings of the 12th Annual International Conference on Mobile Systems, Applications, and Services - MobiSys '14*, 2014, pp. 301-314 [Online]. Disponible en: <http://dl.acm.org/citation.cfm?doid=2594368.2594378>
- [3] Y.-S. Kuo, P. Pannuto, K.-J. Hsiao, y P. Dutta, «Luxapose: Indoor Positioning with Mobile Phones and Visible Light», en *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, 2014, pp. 447-458 [Online]. Disponible en: <http://doi.acm.org/10.1145/2639108.2639109>
- [4] M. Saadi, L. Wattisuttikulkiy, Y. Zhao, y P. Sangwongngam, «Visible Light Communication: Opportunities, Challenges and Channel Models», *International Journal of Electronics & Informatics*, vol. 2, n.º 1, pp. 1-11, 2013.
- [5] P. Connolly y D. Boone, «Indoor Location in Retail: Where Is the Money», *Business Models Analysis Report*, vol. 151, p. 152, 2013.
- [6] L. Li, P. Hu, C. Peng, G. Shen, y F. Zhao, «Epsilon: A Visible Light Based Positioning System», en *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*, 2014, pp. 331-343 [Online]. Disponible en: <https://www.usenix.org/conference/nsdi14/technical-sessions/presentation/li>
- [7] H. Haas, «Wireless Data from Every Light Bulb». 2011 [Online]. Disponible en: https://www.ted.com/talks/harald_haas_wireless_data_from_every_light_bulb. [Accedido: 10-jun-2019]
- [8] A. Bergh, G. Craford, A. Duggal, y R. Haitz, «The Promise and Challenge of Solid-State Lighting», *Physics today*, vol. 54, n.º 12, pp. 42-47, 2001.
- [9] european_union, «Phase-out of Inefficient Lamps Postponed to 1 September 2018», *Energy - European Commission*, abr. 2015 [Online]. Disponible en: <https://ec.europa.eu/energy/en/news/phase-out-inefficient-lamps-postponed-1-september-2018>

- [10] Y. Zhuang *et al.*, «A Survey of Positioning Systems Using Visible LED Lights», *IEEE Communications Surveys Tutorials*, vol. 20, n.º 3, pp. 1963-1988, third-quarter 2018 [Online]. Disponible en: <https://ieeexplore.ieee.org/document/8292854>
- [11] P. H. Pathak, X. Feng, P. Hu, y P. Mohapatra, «Visible Light Communication, Networking, and Sensing: A Survey, Potential and Challenges», *IEEE Communications Surveys Tutorials*, vol. 17, n.º 4, pp. 2047-2077, Fourthquarter 2015.
- [12] D. Karunatilaka, F. Zafar, V. Kalavally, y R. Parthiban, «LED Based Indoor Visible Light Communications: State of the Art», *IEEE Communications Surveys Tutorials*, vol. 17, n.º 3, pp. 1649-1678, thirdquarter 2015.
- [13] H. Haas, «How Fast Can LiFi Be? | LiFi Research», nov. 2015 [Online]. Disponible en: <https://www.lifi.eng.ed.ac.uk/lifi-news/2015-11-28-1320/how-fast-can-lifi-be>. [Accedido: 05-mar-2020]
- [14] S. Rajagopal, R. D. Roberts, y S. Lim, «IEEE 802.15.7 Visible Light Communication: Modulation Schemes and Dimming Support», *IEEE Communications Magazine*, vol. 50, n.º 3, pp. 72-82, mar. 2012.
- [15] eu, «New Lightbulb Rules Will Enable Household Energy Savings and Help Reduce Greenhouse Gas Emissions», *European Commission - European Commission*, ago. 2019 [Online]. Disponible en: https://ec.europa.eu/info/news/new-lightbulb-rules-will-enable-household-energy-savings-and-help-reduce-ghg-emissions-2018-aug-31_en. [Accedido: 05-mar-2020]
- [16] I. Moreno y C.-C. Sun, «Modeling the Radiation Pattern of LEDs», *Opt. Express*, vol. 16, n.º 3, pp. 1808-1819, feb. 2008 [Online]. Disponible en: <http://www.opticsexpress.org/abstract.cfm?URI=oe-16-3-1808>
- [17] U.S. Department of Energy, «Lifetime of White LEDs», p. 2, jul. 2008 [Online]. Disponible en: https://www1.eere.energy.gov/buildings/publications/pdfs/sl/lifetime_white_leds_aug16_r1.pdf. [Accedido: 05-mar-2020]
- [18] C. Jarvis, *The Best Camera Is the One That's with You: iPhone Photography by Chase Jarvis*. New Riders, 2009.
- [19] «Number of Smartphone and Mobile Phone Users Worldwide in 2020: Demographics, Statistics, Predictions - Financesonline.Com» [Online]. Disponible en: <https://financesonline.com/number-of-smartphone-users-worldwide/>. [Accedido: 29-abr-2022]
- [20] «Number of Tablet Users Worldwide from 2013 to 2021 (in Billions)*» [Online]. Disponible en: <https://www.statista.com/statistics/377977/tablet-users-worldwide-forecast/>. [Accedido: 29-abr-2022]

- [21] «Museo Arqueológico de Badajoz», 2017 [Online]. Disponible en: <http://museoarqueologicobadajoz.juntaex.es/web/view/portal/index/standardPage.php?id=211>. [Accedido: 02-mar-2020]
- [22] A. R. Jiménez, F. Seco, y J. Torres-Sospedra, «Tools for Smartphone Multi-Sensor Data Registration and GT Mapping for Positioning Applications», en *2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2019, pp. 1-8.
- [23] «GetSensorDataSuite», *GitLab*. [Online]. Disponible en: <https://gitlab.com/getsensordatasuite>
- [24] T. Aguilera Benítez, «Desarrollo de un sistema de posicionamiento local acústico para dispositivos móviles con modulación de espectro expandido», 2016 [Online]. Disponible en: <https://dehesa.unex.es:8443/handle/10662/4240>. [Accedido: 05-sep-2021]
- [25] M. F. Keskin, A. D. Sezer, y S. Gezici, «Localization via Visible Light Systems», *Proceedings of the IEEE*, vol. 106, n.º 6, pp. 1063-1088, jun. 2018.
- [26] C. Doctorow, «WiFi Isn't Short for "Wireless Fidelity"», *Boing Boing*. nov-2005 [Online]. Disponible en: <https://boingboing.net/2005/11/08/wifi-isnt-short-for.html>. [Accedido: 10-jun-2019]
- [27] C. Long y M. Groth, «Bibliography of Early Optical (Audio) Communications». jun-2005 [Online]. Disponible en: <http://www.bluehaze.com.au/modlight/ModLightBiblio.htm>. [Accedido: 10-jun-2019]
- [28] D. K. Jackson, T. K. Buffaloe, y S. B. Leeb, «Fiat Lux: A Fluorescent Lamp Digital Transceiver», *IEEE Transactions on Industry Applications*, vol. 34, n.º 3, pp. 625-630, 1998.
- [29] M. Day, «LED-driver Considerations». 2005 [Online]. Disponible en: <https://www.ti.com/lit/an/slyt084/slyt084.pdf>. [Accedido: 26-ago-2021]
- [30] H. Parikh, J. Chokshi, N. Gala, y T. Biradar, «Wirelessly Transmitting a Grayscale Image Using Visible Light», en *2013 International Conference on Advances in Technology and Engineering (ICATE)*, 2013, pp. 1-6.
- [31] R. Wood, «Mobile Data Will Grow 6.3 Times between 2013 and 2018 and Growth Will Be Strongest Outside Europe and North America - Insight - News | EPiServer Site». oct-2013 [Online]. Disponible en: <https://web.archive.org/web/20171020035008/http://www.analysismason.com/About-Us/News/Insight/Mobile-data-Oct2013/>. [Accedido: 19-ago-2019]

- [32] S. Wilson, «Wireless Network Data Traffic: Worldwide Trends and Forecasts 2019–2024». abr-2019 [Online]. Disponible en: <https://www.analysysmason.com/Research/Content/Regional-forecasts-/wireless-traffic-forecast-rdnt0/>. [Accedido: 19-ago-2019]
- [33] C. Gabriel, «Wireless Broadband Alliance Industry Report 2013: Global Trends in Public Wi-Fi». 2013 [Online]. Disponible en: <https://web.archive.org/web/20160422054028/http://www.wballiance.com/wba/wp-content/uploads/downloads/2013/11/WBA-Industry-Report-2013.pdf>. [Accedido: 19-ago-2019]
- [34] M. Biagi, T. Borogovac, y T. D. C. Little, «Adaptive Receiver for Indoor Visible Light Communications», *Journal of Lightwave Technology*, vol. 31, n.º 23, pp. 3676-3686, dic. 2013.
- [35] Y. Wang, Y. Wang, N. Chi, J. Yu, y H. Shang, «Demonstration of 575-Mb/s Downlink and 225-Mb/s Uplink Bi-Directional SCM-WDM Visible Light Communication Using RGB LED and Phosphor-Based LED», *Opt. Express, OE*, vol. 21, n.º 1, pp. 1203-1208, ene. 2013 [Online]. Disponible en: <https://www.osapublishing.org/oe/abstract.cfm?uri=oe-21-1-1203>. [Accedido: 19-ago-2019]
- [36] M. Kavehrad, «Sustainable Energy-Efficient Wireless Applications Using Light», *IEEE Communications Magazine*, vol. 48, n.º 12, pp. 66-73, dic. 2010.
- [37] J. K. Kim y E. F. Schubert, «Transcending the Replacement Paradigm of Solid-State Lighting», *Opt. Express, OE*, vol. 16, n.º 26, pp. 21835-21842, dic. 2008 [Online]. Disponible en: <https://www.osapublishing.org/oe/abstract.cfm?uri=oe-16-26-21835>. [Accedido: 19-ago-2019]
- [38] H. Ma, L. Lampe, y S. Hranilovic, «Integration of Indoor Visible Light and Power Line Communication Systems», en *2013 IEEE 17th International Symposium on Power Line Communications and Its Applications*, 2013, pp. 291-296.
- [39] A. Jovicic, J. Li, y T. Richardson, «Visible Light Communication: Opportunities, Challenges and the Path to Market», *IEEE Communications Magazine*, vol. 51, n.º 12, pp. 26-32, dic. 2013.
- [40] «Infrared Data Association». 2012 [Online]. Disponible en: <http://www.irda.org/>. [Accedido: 19-ago-2019]
- [41] D. A. Steigerwald *et al.*, «Illumination with Solid State Lighting Technology», *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 8, n.º 2, pp. 310-320, mar. 2002.

- [42] A. Bernanose, M. Comte, y P. Vouaux, «Sur Un Nouveau Mode d'émission Lumineuse Chez Certains Composés Organiques», *Journal de Chimie Physique*, vol. 50, pp. 64-68, 1953.
- [43] «Ultra-Parallel Visible Light Communications (UP-VLC)». jul-2018 [Online]. Disponible en: <https://web.archive.org/web/20180707183742/http://gow.epsrc.ac.uk/NGBOViewGrant.aspx?GrantRef=EP/K00042X/1>. [Accedido: 27-ago-2021]
- [44] V. Fernández-Luna, D. Sánchez-de Alcázar, J. P. Fernández-Blázquez, A. L. Cortajarena, P. B. Coto, y R. D. Costa, «Deciphering Limitations to Meet Highly Stable Bio-Hybrid Light-Emitting Diodes», *Advanced Functional Materials*, vol. 29, n.º 42, p. 1904356, 2019 [Online]. Disponible en: <https://onlinelibrary.wiley.com/doi/abs/10.1002/adfm.201904356>
- [45] N. Fujimoto y H. Mochizuki, «477 Mbit/s Visible Light Transmission Based on OOK-NRZ Modulation Using a Single Commercially Available Visible LED and a Practical LED Driver with a Pre-Emphasis Circuit», en *Optical Fiber Communication Conference/National Fiber Optic Engineers Conference 2013*, 2013, p. JTh2A.73 [Online]. Disponible en: <http://www.osapublishing.org/abstract.cfm?URI=NFOEC-2013-JTh2A.73>
- [46] N. Fujimoto y H. Mochizuki, «614 Mbit/s OOK-based Transmission by the Duobinary Technique Using a Single Commercially Available Visible LED for High-Speed Visible Light Communications», en *2012 38th European Conference and Exhibition on Optical Communications*, 2012, pp. 1-3.
- [47] S. Zhang *et al.*, «1.5 Gbit/s Multi-Channel Visible Light Communications Using CMOS-Controlled GaN-Based LEDs», *J. Lightwave Technol.*, vol. 31, n.º 8, pp. 1211-1216, abr. 2013 [Online]. Disponible en: <http://jlt.osa.org/abstract.cfm?URI=jlt-31-8-1211>
- [48] M. B. Pursley, «23 - Analog Communications», en *Reference Data for Engineers (Ninth Edition)*, W. M. Middleton y M. E. Van Valkenburg, Eds. Woburn: Newnes, 2002, pp. 23-1 [Online]. Disponible en: <https://www.sciencedirect.com/science/article/pii/B978075067291750025X>. [Accedido: 28-ago-2021]
- [49] H. Sugiyama, S. Haruyama, y M. Nakagawa, «Brightness Control Methods for Illumination and Visible-Light Communication Systems», en *2007 Third International Conference on Wireless and Mobile Communications (ICWMC'07)*, 2007, pp. 78-78.

- [50] T. Ohtsuki, «Rate Adaptive Indoor Infrared Wireless Communication Systems Using Repeated and Punctured Convolutional Codes», en *1999 IEEE International Conference on Communications (Cat. No. 99CH36311)*, 1999, vol. 1, pp. 609-613 vol.1.
- [51] H. Sugiyama y K. Nosu, «MPPM: A Method for Improving the Band-Utilization Efficiency in Optical PPM», *Journal of Lightwave Technology*, vol. 7, n.º 3, pp. 465-472, mar. 1989.
- [52] W. Popoola, E. Poves, y H. Haas, «Generalised Space Shift Keying for Visible Light Communications», en *2012 8th International Symposium on Communication Systems, Networks Digital Signal Processing (CSNDSP)*, 2012, pp. 1-4.
- [53] W. O. Popoola, E. Poves, y H. Haas, «Error Performance of Generalised Space Shift Keying for Indoor Visible Light Communications», *IEEE Transactions on Communications*, vol. 61, n.º 5, pp. 1968-1976, 2013.
- [54] H. Elgala, R. Mesleh, H. Haas, y B. Pricope, «OFDM Visible Light Wireless Communication Based on White LEDs», en *2007 IEEE 65th Vehicular Technology Conference - VTC2007-Spring*, 2007, pp. 2185-2189.
- [55] E. Monteiro y S. Hranilovic, «Design and Implementation of Color-Shift Keying for Visible Light Communications», *J. Lightwave Technol.*, vol. 32, n.º 10, pp. 2053-2060, 2014 [Online]. Disponible en: <http://jlt.osa.org/abstract.cfm?URI=jlt-32-10-2053>
- [56] Z. Yang, Z. WANG, J. Zhang, C. Huang, y Q. Zhang, «Polarization-Based Visible Light Positioning», *IEEE Transactions on Mobile Computing*, vol. 18, n.º 3, pp. 715-727, mar. 2019.
- [57] A. S. Tanenbaum, «Network Protocols», *ACM Comput. Surv.*, vol. 13, n.º 4, pp. 453-489, dic. 1981 [Online]. Disponible en: <https://doi.org/10.1145/356859.356864>
- [58] S. Gezici, «A Survey on Wireless Position Estimation», *Wireless personal communications*, vol. 44, n.º 3, pp. 263-282, 2008.
- [59] M. F. Keskin, S. Gezici, y O. Arikan, «Direct and Two-Step Positioning in Visible Light Systems», *IEEE Transactions on Communications*, vol. 66, n.º 1, pp. 239-254, ene. 2018.
- [60] D. Kim, J. K. Park, y J. T. Kim, «High-Efficient and Low-Cost Biased Multilevel Modulation Technique for IM/DD-Based VLP Systems», *IEEE Access*, vol. 8, pp. 218954-218965, 2020.
- [61] M. Sauter, *From GSM to LTE.: An Introduction to Mobile Networks and Mobile Broadband*. John Wiley & Sons, 2010.

- [62] J. R. Barry, *Wireless Infrared Communications*, vol. 280. Springer Science & Business Media, 2012.
- [63] S. Yang, E. Jung, y S. Han, «Indoor Location Estimation Based on LED Visible Light Communication Using Multiple Optical Receivers», *IEEE Communications Letters*, vol. 17, n.º 9, pp. 1834-1837, sep. 2013.
- [64] W. Zhang, M. I. Sakib Chowdhury, y M. Kavehrad, «Asynchronous Indoor Positioning System Based on Visible Light Communications», *Optical Engineering*, vol. 53, n.º 4, pp. 1-10 - 10, 2014 [Online]. Disponible en: <https://doi.org/10.1117/1.OE.53.4.045105>
- [65] B. O'Keefe, «Firebrick Team 2017 | Electrical and Computer Engineering Design Handbook». 2017 [Online]. Disponible en: <https://sites.tufts.edu/eeseniordesignhandbook/electrical-and-computer-engineering-design-handbook/senior-capstone-projects-summary-for-the-2016-17-academic-year/firebrick-team-2017/>. [Accedido: 31-jul-2019]
- [66] T. Q. Wang, Y. A. Sekercioglu, A. Neild, y J. Armstrong, «Position Accuracy of Time-of-Arrival Based Ranging Using Visible Light With Application in Indoor Localization Systems», *Journal of Lightwave Technology*, vol. 31, n.º 20, pp. 3302-3308, oct. 2013.
- [67] J. Luo, L. Fan, y H. Li, «Indoor Positioning Systems Based on Visible Light Communication: State of the Art», *IEEE Communications Surveys Tutorials*, vol. 19, n.º 4, pp. 2871-2893, Fourthquarter 2017 [Online]. Disponible en: <https://ieeexplore.ieee.org/abstract/document/8015106>
- [68] H. B. Lee, «A Novel Procedure for Assessing the Accuracy of Hyperbolic Multilateration Systems», *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-11, n.º 1, pp. 2-15, ene. 1975.
- [69] S. Jung, S. Hann, y C. Park, «TDOA-based Optical Wireless Indoor Localization Using LED Ceiling Lamps», *IEEE Transactions on Consumer Electronics*, vol. 57, n.º 4, pp. 1592-1597, nov. 2011.
- [70] J. H. Y. Nah, R. Parthiban, y M. H. Jaward, «Visible Light Communications Localization Using TDOA-based Coherent Heterodyne Detection», en *2013 IEEE 4th International Conference on Photonics (ICP)*, 2013, pp. 247-249.
- [71] T.-H. Do y M. Yoo, «TDOA-based Indoor Positioning Using Visible Light», *Photon Netw Commun*, vol. 27, n.º 2, pp. 80-88, abr. 2014 [Online]. Disponible en: <https://doi.org/10.1007/s11107-014-0428-4>. [Accedido: 15-ago-2019]

- [72] A. Arafa, S. Dalmiya, R. Klukas, y J. F. Holzman, «Angle-of-Arrival Reception for Optical Wireless Location Technology», *Opt. Express, OE*, vol. 23, n.º 6, pp. 7755-7766, mar. 2015 [Online]. Disponible en: <https://www.osapublishing.org/oe/abstract.cfm?uri=oe-23-6-7755>. [Accedido: 16-ago-2019]
- [73] M. Yasir, S. Ho, y B. N. Vellambi, «Indoor Localization Using Visible Light and Accelerometer», en *2013 IEEE Global Communications Conference (GLOBECOM)*, 2013, pp. 3341-3346.
- [74] S. Cincotta, C. He, A. Neild, y J. Armstrong, «High Angular Resolution Visible Light Positioning Using a Quadrant Photodiode Angular Diversity Aperture Receiver (QADA)», *Opt. Express, OE*, vol. 26, n.º 7, pp. 9230-9242, abr. 2018 [Online]. Disponible en: <https://www.osapublishing.org/oe/abstract.cfm?uri=oe-26-7-9230>. [Accedido: 16-ago-2019]
- [75] S. Cincotta, C. He, A. Neild, y J. Armstrong, «Indoor Visible Light Positioning: Overcoming the Practical Limitations of the Quadrant Angular Diversity Aperture Receiver (QADA) by Using the Two-Stage QADA-Plus Receiver», *Sensors*, vol. 19, n.º 4, 2019 [Online]. Disponible en: <https://www.mdpi.com/1424-8220/19/4/956>
- [76] A. De-La-Llana-Calvo, J. L. Lazaro-Galilea, A. Gardel-Vicente, D. Rodriguez-Navarro, y I. Bravo-Munoz, «Indoor Positioning System Based on LED Lighting and PSD Sensor», *International Conference on Indoor Positioning and Indoor Navigation*, p. 8, 2019.
- [77] M. F. Keskin y S. Gezici, «Comparative Theoretical Analysis of Distance Estimation in Visible Light Positioning Systems», *Journal of Lightwave Technology*, vol. 34, n.º 3, pp. 854-865, feb. 2016.
- [78] Y. U. Lee y M. Kavehrad, «Two Hybrid Positioning System Design Techniques with Lighting LEDs and Ad-Hoc Wireless Network», *IEEE Transactions on Consumer Electronics*, vol. 58, n.º 4, pp. 1176-1184, nov. 2012 [Online]. Disponible en: <https://ieeexplore.ieee.org/abstract/document/6414983>
- [79] S. C. Ergen, «ZigBee/IEEE 802.15. 4 Summary», *UC Berkeley, September*, vol. 10, n.º 17, p. 11, 2004.
- [80] M. Nakajima y S. Haruyama, «New Indoor Navigation System for Visually Impaired People Using Visible Light Communication», *EURASIP Journal on Wireless Communications and Networking*, vol. 2013, n.º 1, p. 37, feb. 2013 [Online]. Disponible en: <https://doi.org/10.1186/1687-1499-2013-37>

- [81] P. Cherntanomwong y W. Chantharasena, «Indoor Localization System Using Visible Light Communication», en *2015 7th International Conference on Information Technology and Electrical Engineering (ICITEE)*, 2015, pp. 480-483 [Online]. Disponible en: <https://ieeexplore.ieee.org/abstract/document/7408994>
- [82] S. Gezici y H. V. Poor, «Position Estimation via Ultra-Wide-Band Signals», *Proceedings of the IEEE*, vol. 97, n.º 2, pp. 386-403, feb. 2009.
- [83] K. Kaemarungsi y P. Krishnamurthy, «Modeling of Indoor Positioning Systems Based on Location Fingerprinting», en *IEEE INFOCOM 2004*, 2004, vol. 2, pp. 1012-1022 vol.2.
- [84] J. Vongkulbhisal, B. Chantaramolee, Y. Zhao, y W. S. Mohammed, «A Fingerprinting-Based Indoor Localization System Using Intensity Modulation of Light Emitting Diodes», *Microwave and Optical Technology Letters*, vol. 54, n.º 5, pp. 1218-1227, 2012 [Online]. Disponible en: <https://onlinelibrary.wiley.com/doi/abs/10.1002/mop.26763>
- [85] S. Feng, X. Li, R. Zhang, M. Jiang, y L. Hanzo, «Hybrid Positioning Aided Amorphous-Cell Assisted User-Centric Visible Light Downlink Techniques», *IEEE Access*, vol. 4, pp. 2705-2713, 2016.
- [86] T. Yamazato y S. Haruyama, «Image Sensor Based Visible Light Communication and Its Application to Pose, Position, and Range Estimations», *IEICE Transactions on Communications*, vol. E97.B, n.º 9, pp. 1759-1765, 2014.
- [87] J. Weng, P. Cohen, M. Herniou, *et al.*, «Camera Calibration with Distortion Models and Accuracy Evaluation», *IEEE Transactions on pattern analysis and machine intelligence*, vol. 14, n.º 10, pp. 965-980, 1992.
- [88] Z. Zhang, «A Flexible New Technique for Camera Calibration», *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, n.º 11, pp. 1330-1334, nov. 2000.
- [89] R. Szeliski, *Computer Vision: Algorithms and Applications*. Springer Science & Business Media, 2010 [Online]. Disponible en: <http://szeliski.org/Book/>
- [90] T.-H. Do y M. Yoo, «An In-Depth Survey of Visible Light Communication Based Positioning Systems», *Sensors*, vol. 16, n.º 5, 2016 [Online]. Disponible en: <http://www.mdpi.com/1424-8220/16/5/678>
- [91] M. S. Rahman, M. M. Haque, y Ki-Doo Kim, «High Precision Indoor Positioning Using Lighting LED and Image Sensor», en *14th International Conference on Computer and Information Technology (ICCIT 2011)*, 2011, pp. 309-314.

- [92] M. S. Rahman, M. M. Haque, y K.-D. Kim, «Indoor Positioning by LED Visible Light Communication and Image Sensors», *International Journal of Electrical and Computer Engineering*, vol. 1, n.º 2, p. 161, 2011.
- [93] Byung Yeon Kim, Jung-Sik Cho, Youngil Park, y Ki-Doo Kim, «Implementation of Indoor Positioning Using LED and Dual PC Cameras», en *2012 Fourth International Conference on Ubiquitous and Future Networks (ICUFN)*, 2012, pp. 476-477.
- [94] M. Moon, S. Choi, J. Park, y J. Y. Kim, «Indoor Positioning System Using LED Lights and a Dual Image Sensor», *J. Opt. Soc. Korea*, vol. 19, n.º 6, pp. 586-591, dic. 2015 [Online]. Disponible en: <http://www.osapublishing.org/josk/abstract.cfm?URI=josk-19-6-586>
- [95] M. Sakata, Y. Yasumuro, M. Imura, Y. Manabe, y K. Chihara, «ALTAIR: Automatic Location Tracking System Using Active IR-tag», en *Proceedings of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems, MFI2003.*, 2003, pp. 299-304.
- [96] S. Yang, H. Kim, Y. Son, y S. Han, «Three-Dimensional Visible Light Indoor Localization Using AOA and RSS With Multiple Optical Receivers», *Journal of Lightwave Technology*, vol. 32, n.º 14, pp. 2480-2485, jul. 2014.
- [97] J. Fang *et al.*, «High-Speed Indoor Navigation System Based on Visible Light and Mobile Phone», *IEEE Photonics Journal*, vol. 9, n.º 2, pp. 1-11, abr. 2017.
- [98] J. Kim, S. Yang, Y. Son, y S. Han, «High-Resolution Indoor Positioning Using Light Emitting Diode Visible Light and Camera Image Sensor», *IET Optoelectronics*, vol. 10, n.º 5, pp. 184-192, 2016.
- [99] Y. Ji *et al.*, «A Single LED Lamp Positioning System Based on CMOS Camera and Visible Light Communication», *Optics Communications*, vol. 443, pp. 48-54, 2019 [Online]. Disponible en: <http://www.sciencedirect.com/science/article/pii/S0030401819301919>
- [100] N. Kanopoulos, N. Vasanthavada, y R. L. Baker, «Design of an Image Edge Detection Filter Using the Sobel Operator», *IEEE Journal of Solid-State Circuits*, vol. 23, n.º 2, pp. 358-367, abr. 1988.
- [101] C. Liang, L. Chang, y H. H. Chen, «Analysis and Compensation of Rolling Shutter Effect», *IEEE Transactions on Image Processing*, vol. 17, n.º 8, pp. 1323-1330, ago. 2008.
- [102] M. Vasilakis, «DynaLight: A Dynamic Visible Light Communication Link for Smartphones», 2015 [Online]. Disponible en: <http://resolver.tudelft.nl/uuid:6dc20955-c2ff-48b4-aa3c-1f91dffccba0>. [Accedido: 16-may-2019]

- [103] C. Chow, C. Chen, y S. Chen, «Enhancement of Signal Performance in LED Visible Light Communications Using Mobile Phone Camera», *IEEE Photonics Journal*, vol. 7, n.º 5, pp. 1-7, oct. 2015.
- [104] B. Fette *et al.*, «Communication Protocols and Modulation», en *RF and Wireless Technologies: Know It All*, Elsevier, 2007, pp. 26-27.
- [105] J. D. Gutiérrez, F. J. Álvarez, T. Aguilera, J. A. Paredes, y J. Morera, «Visible Light Positioning for Smartphones Based on Biphase Mark Coding: A Proof of Concept», en *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2018, p. 4.
- [106] N. Otsu, «A Threshold Selection Method from Gray-Level Histograms», *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, n.º 1, pp. 62-66, ene. 1979.
- [107] GISS, «Presentation of the Limus Project», GISS. mar-2019 [Online]. Disponible en: <https://giss.unex.es/2019/03/21/presentation-of-the-limus-project/>. [Accedido: 12-may-2019]
- [108] GISS, «GISS: Grupo de Investigación en Sistemas Sensoriales», GISS. abr-2018 [Online]. Disponible en: <https://giss.unex.es/>. [Accedido: 20-mar-2020]
- [109] Raspberry Pi Foundation, «Teach, Learn, and Make with Raspberry Pi – Raspberry Pi». feb-2012 [Online]. Disponible en: <https://www.raspberrypi.org>. [Accedido: 20-mar-2020]
- [110] J. D. Gutiérrez, F. J. Aranda, T. Aguilera, y F. J. Álvarez, «LIMUS: Exploration of Technological Prototypes for Location-Based Services in Museums», 2019.
- [111] Adafruit, «Overview | Adafruit Feather HUZZAH ESP8266 | Adafruit Learning System». nov-2015 [Online]. Disponible en: <https://learn.adafruit.com/adafruit-feather-huzzah-esp8266/overview>. [Accedido: 20-may-2019]
- [112] «iPhone 5s - Technical Specifications». [Online]. Disponible en: https://support.apple.com/kb/sp685?locale=en_GB. [Accedido: 02-ago-2019]
- [113] «Máquina Láser de Grabado y Corte PC10/80 | PerezCamps - Fresadoras CNC y Máquinas Láser de Grabado y Corte». [Online]. Disponible en: https://www.perezcamps.com/es/maquina-laser-de-grabado-y-corte-pc10-80_2450. [Accedido: 20-mar-2020]
- [114] MIT, «Welcome | FabLabs», *FabLabs.io - The Fab Lab Network*. 2001 [Online]. Disponible en: <https://fablabs.io/>. [Accedido: 20-mar-2020]
- [115] «Boxes.Py». [Online]. Disponible en: <https://www.festi.info/boxes.py/>. [Accedido: 20-mar-2020]

- [116] NodeMCU, «NodeMcu – An Open-Source Firmware Based on ESP8266 Wifi-Soc.». 2014 [Online]. Disponible en: http://www.nodemcu.com/index_en.html. [Accedido: 27-jun-2019]
- [117] «A Beginner’s Guide to the ESP8266». [Online]. Disponible en: <https://tttapa.github.io/ESP8266/Chap07%20-%20Wi-Fi%20Connections.html>. [Accedido: 26-jul-2019]
- [118] «CMX200D3 - Crydom CMX Series». [Online]. Disponible en: <http://www.crydom.com/en/products/catalog/cmx-series-dc-pcb-mount.pdf>. [Accedido: 20-mar-2020]
- [119] «KiCad EDA». [Online]. Disponible en: <https://www.kicad.org/>. [Accedido: 03-oct-2021]
- [120] «Mi Global Home». [Online]. Disponible en: <https://www.mi.com/global/mi8>. [Accedido: 21-sep-2021]
- [121] W. Fulton, «The Math of Camera Field of View Calculations (FOV)», 2018 [Online]. Disponible en: <https://www.scantips.com/lights/fieldofviewmath.html>. [Accedido: 21-abr-2020]
- [122] «Xiaomi Mi 8 - Specifications», *DeviceSpecifications* [Online]. Disponible en: <https://www.devicespecifications.com/en/model/c52c4895>. [Accedido: 20-sep-2021]
- [123] «Exmor», *Wikipedia*. 2020 [Online]. Disponible en: <https://en.wikipedia.org/wiki/Exmor>. [Accedido: 23-sep-2021]
- [124] «Exchangeable Image File Format for Digital Still Cameras: Exif Version 2.32». 2019 [Online]. Disponible en: http://cipa.jp/std/documents/download_e.html?DC-008-Translation-2019-E. [Accedido: 25-sep-2021]
- [125] W. B. Pennebaker y J. L. Mitchell, *JPEG: Still Image Data Compression Standard*. Springer Science & Business Media, 1992.
- [126] «FXScript Reference: RGB and YUV Color». [Online]. Disponible en: http://joemaller.com/fcp/fxscript_yuv_color.shtml. [Accedido: 03-oct-2021]
- [127] R. Fry y P. Fourzon, *The Saga of Special Effects*. Englewood Cliffs, NJ: Prentice-Hall, 1977.
- [128] O. Vartanian *et al.*, «Architectural Design and the Brain: Effects of Ceiling Height and Perceived Enclosure on Beauty Judgments and Approach-Avoidance Decisions», *Journal of Environmental Psychology*, vol. 41, pp. 10-18, 2015 [Online]. Disponible en: <https://www.sciencedirect.com/science/article/pii/S0272494414001030>

- [129] P. M. Visscher, «Sizing up Human Height Variation», *Nature genetics*, vol. 40, n.º 5, pp. 489-490, 2008.
- [130] G. van Rossum y Python Software Foundation, «Welcome to Python.Org», *Python.org*, 1990 [Online]. Disponible en: <https://www.python.org/>. [Accedido: 20-sep-2021]
- [131] J. D. Gutiérrez Gallardo, «Communication: Encoder». abr-2020 [Online]. Disponible en: <https://gitlab.com/enreas-projects/thesis/communication/encoder>. [Accedido: 20-sep-2021]
- [132] J. D. Gutiérrez Gallardo, «Thesis». abr-2020 [Online]. Disponible en: <https://gitlab.com/enreas-projects/thesis/>. [Accedido: 20-sep-2021]
- [133] I. Kravets, «PlatformIO: An Open Source Ecosystem for IoT Development», *PlatformIO*, 2014 [Online]. Disponible en: <https://platformio.org>. [Accedido: 21-sep-2021]
- [134] J. D. Gutiérrez Gallardo, «Communication: Emitter». abr-2020 [Online]. Disponible en: <https://gitlab.com/enreas-projects/thesis/communication/emitter>. [Accedido: 21-sep-2021]
- [135] «PyCharm: The Python IDE for Professional Developers by JetBrains». [Online]. Disponible en: <https://www.jetbrains.com/pycharm/>. [Accedido: 28-sep-2021]
- [136] P. J. Brown, «Writing Interactive Compilers and Interpreters», *Wiley Series in Computing*, Chichester: Wiley, 1979, 1979.
- [137] J. D. Gutiérrez Gallardo, «Communication: Decoder». abr-2020 [Online]. Disponible en: <https://gitlab.com/enreas-projects/thesis/communication/decoder>. [Accedido: 22-sep-2021]
- [138] J. D. Gutiérrez Gallardo, «Communication: Encoder - README.Md». abr-2020 [Online]. Disponible en: <https://gitlab.com/enreas-projects/thesis/communication/decoder/-/blob/master/README.md>. [Accedido: 28-sep-2021]
- [139] «Index | TIOBE - The Software Quality Company» [Online]. Disponible en: <https://www.tiobe.com/tiobe-index/>. [Accedido: 25-sep-2021]
- [140] «Google I/O 2019: Empowering Developers to Build the Best Experiences on Android + Play», *Android Developers Blog* [Online]. Disponible en: <https://android-developers.googleblog.com/2019/05/google-io-2019-empowering-developers-to-build-experiences-on-Android-Play.html>. [Accedido: 24-sep-2021]

- [141] J. D. Gutiérrez Gallardo, «Communication: Sampler». abr-2020 [Online]. Disponible en: <https://gitlab.com/enreas-projects/thesis/communication/sampler>. [Accedido: 22-sep-2021]
- [142] «LOPSI Research Group: Localization and Exploration for Intelligent Systems». [Online]. Disponible en: <http://lopsi.car.upm-csic.es/>. [Accedido: 29-ene-2022]
- [143] «IPIN Conference». [Online]. Disponible en: <http://ipin-conference.org/>. [Accedido: 29-sep-2021]
- [144] «GitLab». [Online]. Disponible en: <https://about.gitlab.com/>. [Accedido: 30-sep-2021]
- [145] «MATLAB». [Online]. Disponible en: <https://www.mathworks.com/products/matlab.html>. [Accedido: 01-oct-2021]
- [146] «Git - Git-Request-Pull Documentation». [Online]. Disponible en: <https://git-scm.com/docs/git-request-pull>. [Accedido: 01-oct-2021]
- [147] P. Divine, «String Case Styles: Camel, Pascal, Snake, and Kebab Case». jul-2019 [Online]. Disponible en: <https://betterprogramming.pub/string-case-styles-camel-pascal-snake-and-kebab-case-981407998841>. [Accedido: 30-sep-2021]
- [148] J. D. Gutiérrez Gallardo, «Communication: Battery». abr-2020 [Online]. Disponible en: <https://gitlab.com/enreas-projects/thesis/communication/battery>. [Accedido: 20-abr-2020]
- [149] B. Foundation, «Blender.Org - Home of the Blender Project - Free and Open 3D Creation Software», *blender.org*. [Online]. Disponible en: <https://www.blender.org/>. [Accedido: 22-sep-2021]
- [150] «Introduction — Blender Manual» [Online]. Disponible en: <https://docs.blender.org/manual/en/latest/advanced/scripting/introduction.html>. [Accedido: 22-sep-2021]
- [151] «ISOCELL Slim 3T1 | Mobile Image Sensor | Samsung Official». [Online]. Disponible en: <https://www.samsung.com/semiconductor/image-sensor/mobile-image-sensor/S5K3T1/>. [Accedido: 21-sep-2021]
- [152] «Mobile | TV | Home Electronics | Home Appliances | Samsung US», *Samsung Electronics America*. [Online]. Disponible en: <https://www.samsung.com/us/>. [Accedido: 20-sep-2021]
- [153] «Sony Semiconductor Solutions Group», *Sony Semiconductor Solutions Group*. [Online]. Disponible en: <https://www.sony-semicon.co.jp/e/>. [Accedido: 21-sep-2021]

- [154] E. International, «ECMA-404—The JSON Data Interchange Format». dic-2017 [Online]. Disponible en: <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf>. [Accedido: 24-sep-2021]
- [155] «Add-Ons — Blender Manual». [Online]. Disponible en: <https://docs.blender.org/manual/en/latest/editors/preferences/addons.html>. [Accedido: 22-sep-2021]
- [156] P. Harvey, «ExifTool». [Online]. Disponible en: <https://exiftool.org/>. [Accedido: 27-sep-2021]
- [157] «The Official YAML Web Site». [Online]. Disponible en: <https://yaml.org/>. [Accedido: 15-nov-2021]
- [158] J. D. Gutiérrez Gallardo, «Positioning: ACP». abr-2020 [Online]. Disponible en: https://gitlab.com/enreas-projects/thesis/acp_addon. [Accedido: 22-sep-2021]
- [159] J. E. Solem, *Programming Computer Vision with Python: Tools and Algorithms for Analyzing Images*. " O'Reilly Media, Inc.", 2012.
- [160] R. Kingslake, *A History of the Photographic Lens*. Academic Press, 1989.
- [161] S. Mallick, «Geometry of Image Formation». feb-2020 [Online]. Disponible en: <https://learnopencv.com/geometry-of-image-formation/>. [Accedido: 24-feb-2022]
- [162] D. Forsyth y J. Ponce, *Computer Vision: A Modern Approach*. Prentice hall, 2011.
- [163] B. Rosenhahn, C. Perwass, y G. Sommer, «Foundations about 2D-3D Pose Estimation», *CVonline: The Evolving, Distributed, Non-Proprietary, On-Line Compendium of Computer Vision*, Edinburgh, 2004.
- [164] W. E. L. Grimson, *Object Recognition by Computer: The Role of Geometric Constraints*. Mit Press, 1991.
- [165] S. Mallick, «Head Pose Estimation Using OpenCV and Dlib | LearnOpenCV #». sep-2016 [Online]. Disponible en: <https://learnopencv.com/head-pose-estimation-using-opencv-and-dlib/>. [Accedido: 25-feb-2022]
- [166] Y. I. Abdel-Aziz, H. M. Karara, y M. Hauck, «Direct Linear Transformation from Comparator Coordinates into Object Space Coordinates in Close-Range Photogrammetry*», *Photogrammetric Engineering & Remote Sensing*, vol. 81, n.º 2, pp. 103-107, 2015 [Online]. Disponible en: <https://www.sciencedirect.com/science/article/pii/S0099111215303086>

- [167] J. J. Moré, «The Levenberg-Marquardt Algorithm: Implementation and Theory», en *Numerical Analysis*, Springer, 1978, pp. 105-116.
- [168] K. Madsen, H. B. Nielsen, y O. Tingleff, «Methods for Non-Linear Least Squares Problems», 2004.
- [169] E. Eade, «Gauss-Newton/Levenberg-Marquardt Optimization», *Tech. Rep.*, 2013.
- [170] V. Lepetit, F. Moreno-Noguer, y P. Fua, «EPnP: An Accurate $O(n)$ Solution to the PnP Problem», *International Journal of Computer Vision*, vol. 81, n.º 2, p. 155, jul. 2008 [Online]. Disponible en: <https://doi.org/10.1007/s11263-008-0152-6>
- [171] G. Terzakis y M. Lourakis, «A Consistently Fast and Globally Optimal Solution to the Perspective-n-Point Problem», en *Computer Vision – ECCV 2020*, 2020, pp. 478-494.
- [172] J. B. Fraleigh y R. A. Beauregard, *Linear Algebra*. Addison Wesley Publishing Company, 1990.
- [173] «OpenCV». [Online]. Disponible en: <https://opencv.org/>. [Accedido: 27-jun-2019]
- [174] D. H. Douglas y T. K. Peucker, «Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or Its Caricature», *Cartographica: the international journal for geographic information and geovisualization*, vol. 10, n.º 2, pp. 112-122, 1973.
- [175] J. Sklansky, «Finding the Convex Hull of a Simple Polygon», *Pattern Recognition Letters*, vol. 1, n.º 2, pp. 79-83, 1982.
- [176] «QUERCUS SEG». [Online]. Disponible en: <https://quercusseg.unex.es/>. [Accedido: 14-abr-2022]
- [177] «I3lab». [Online]. Disponible en: <https://i3lab.unex.es/>. [Accedido: 14-abr-2022]
- [178] «—Portal de La UEX - Bienvenido a La Universidad de Extremadura». [Online]. Disponible en: <https://www.unex.es/conoce-la-uex/centros/epcc/>. [Accedido: 14-abr-2022]
- [179] «SUAOKI S9». [Online]. Disponible en: <https://www.amazon.es/gp/product/B071YPS2J9/>. [Accedido: 21-abr-2022]
- [180] J. D. Gutiérrez *et al.*, «GetSensorData: An Extensible Android-based Application for Multi-Sensor Data Registration», *SoftwareX*, 2022.

- [181] J. D. Gutierrez, T. Aguilera, F. J. Álvarez Franco, y F. J. Aranda, «A Blender-based Simulation Tool for Visible Light Positioning with Portable Devices», en *2022 International Instrumentation and Measurement Technology Conference (I2MTC)*, 2022, p. 6.
- [182] F. J. Álvarez, T. Aguilera, J. D. Gutiérrez, F. J. Aranda, y J. A. Moreno, «Identificación de Marcas En Entornos Museísticos: Comparación de Tecnologías Acústica, Bluetooth de Baja Energía y Luz Visible», en *27º Seminario Anual de Automática, Electrónica Industrial y Automática (SAAEI)*, 2020.
- [183] J. D. Gutiérrez, F. J. Álvarez, T. Aguilera, J. Morera, y J. A. Moreno, «LI-MUS: Una Guía Inteligente Para Espacios Museísticos Basada En El Uso de Dispositivos Portables», en *XXV Seminario Anual de Automática, Electrónica Industrial e Instrumentación (SAAEI)*, 2018 [Online]. Disponible en: http://www.saei.org/edicion18/docs/Libro_Resumenes_SAAEI18.pdf

