



TESIS DOCTORAL

**UNA NUEVA METODOLOGÍA DE SEGMENTACIÓN DE IMÁGENES BASADA EN
CONTORNOS ACTIVOS. APLICACIÓN A LA SEGMENTACIÓN DE IMÁGENES
MÉDICAS PARA LA RECONSTRUCCIÓN TRIDIMENSIONAL DE ESTRUCTURAS
ANATÓMICAS**

VALENTÍN MASERO VARGAS

**DEPARTAMENTO DE INGENIERÍA DE SISTEMAS INFORMÁTICOS Y
TELEMÁTICOS**

2015



TESIS DOCTORAL

**UNA NUEVA METODOLOGÍA DE SEGMENTACIÓN DE IMÁGENES BASADA EN
CONTORNOS ACTIVOS. APLICACIÓN A LA SEGMENTACIÓN DE IMÁGENES MÉDICAS
PARA LA RECONSTRUCCIÓN TRIDIMENSIONAL DE ESTRUCTURAS ANATÓMICAS**

VALENTÍN MASERO VARGAS

**DEPARTAMENTO DE INGENIERÍA DE SISTEMAS
INFORMÁTICOS Y TELEMÁTICOS**

Conformidad del Director:

Fdo: José Moreno del Pozo

2015

*A toda mi familia,
especialmente a mis padres,
a Inma y a Pablo*

Agradecimientos

Todo el trabajo reflejado aquí ha sido posible por la ayuda y compromiso de mucha gente. La lista de nombres de personas que me han ayudado de una u otra manera a lo largo de estos años sería difícil de incluir aquí sin cometer el error de olvidar a alguien. Aún así, lo intentaré.

En primer lugar, a mi director de tesis, sin el cual todo esto no habría sido posible, por su ayuda desde los comienzos hasta el final y por su inestimable ayuda en el campo académico, científico y personal.

En segundo lugar a mi familia y amigos, por las largas ausencias e ingente cantidad de horas sin dedicarles la atención necesaria.

Además, ha habido muchos amigos y compañeros que me han ayudado de una o de otra forma en algún momento de este trayecto:

En los primeros años, cuando empecé en el mundo de la Informática aplicada a la sanidad, tuve el inestimable apoyo y ayuda de todo el personal del entonces Centro de Cirugía de Mínima Invasión (CCMI) donde disfruté durante un tiempo de una beca predoctoral, incorporándome posteriormente al Departamento de Informática de la

Universidad de Extremadura. De estos años en los que desarrollé mi labor en el CCMI, estoy agradecido, sobre todo, al director científico del mismo D. Jesús Usón Gargallo, al entonces investigador y responsable del Área de Laparoscopia Francisco Miguel Sánchez Margallo, a Rosa M^a García y a Magdalena, y Maximiliano Martín, entonces colaborador externo del área de Informática del CCMI y del que también obtuve ayuda durante esos años.

A todos los compañeros del área médico-quirúrgica del CCMI que me ofrecieron su inestimable apoyo y ayuda ofreciéndome sus amplios conocimientos del campo sanitario y científico. Aún a riesgo de olvidar a algún compañero y amigo del CCMI, tengo que estar especialmente agradecido a mis amigos y compañeros Francisco Miguel Sánchez, Federico Soria, Rafael Lima, María Fernanda Martín, Carmen Calles, Marcos Lima y Javier Hernández. También al coordinador del Servicio de mantenimiento Juan Godoy por su gran ayuda en tareas técnicas y con las instalaciones de telecomunicaciones.

A los miembros de la Wake Forest University (WFU), que me acogieron de estancia predoctoral en Carolina del Norte (EE.UU.), que me ayudaron tanto y me aportaron tanto conocimiento sobre el mundo de la Imagen médica, el procesamiento de imágenes y la Ingeniería Médica en general. Todos ellos me permitieron conocer una tecnología que en España era aún incipiente y que en EE.UU. llevaba ya muchos años desarrollándose. Principalmente a los investigadores y doctores Yaorong Ge, David Vining y William Chimiak, por su apoyo en el campo científico y personal durante mi estancia en la WFU. Gracias a ellos comencé a trabajar con los contornos activos. Al director del departamento de la WFU, Peter Santago por el apoyo institucional y haber facilitado en todo lo posible la infraestructura necesaria durante mi estancia en Carolina del Norte.

A los estudiantes de doctorado de la WFU con los que tuve oportunidad de charlar e intercambiar valiosas conversaciones sobre el campo de Procesamiento de Imágenes, Imagen Médica e Ingeniería Médica, especialmente a mis amigos Ahn y Chris.

Dentro de la Universidad de Extremadura he de estar agradecido a un gran número de compañeros, a los que no me atrevo a enumerar por miedo a olvidar a alguien. Por ello me limitaré a mencionar a aquellos que me han ayudado más directamente: mi director de tesis y amigo Pepe Moreno, por su inestimable ayuda en el campo científico en general y aportar sus conocimientos sobre procesamiento digital de imágenes e informática gráfica. Y a mi amigo Juan Miguel, que me aportó siempre una inestimable ayuda y una visión diferente de

la situación, aportando una visión muy amplia sobre los más diversos temas relacionados con la Informática y la investigación.

Resumen

Este trabajo ha tenido como objetivo principal desarrollar una metodología para la reconstrucción tridimensional de estructuras anatómicas a partir de imágenes médicas con las que se pudiese obtener posteriormente modelos tridimensionales de calidad. Dentro del proceso completo de la reconstrucción tridimensional, una de las fases más importantes y complejas es la segmentación de cada una de las imágenes. Por esto, nuestro trabajo se ha focalizado en la fase de segmentación de las imágenes médicas iniciales, ya que el resultado de esta fase determina en gran medida la precisión de los modelos tridimensionales y, por tanto, de las imágenes tridimensionales finales que se mostrarán al profesional sanitario encargado de interpretar dichas imágenes.

Presentamos un nuevo método de segmentación basado en contornos activos que hemos llamado Contornos Activos Auto-Reconfigurables (CAAR), que son un tipo de contornos activos que se ejecutan en dos fases. En la primera fase se ejecuta un algoritmo de contorno activo cuyos resultados son comprobados y mejorados de forma automática durante la segunda fase. Esta metodología se basa en detectar automáticamente las zonas en las que no se ha obtenido un buen resultado y elegir

posteriormente, también de forma automática, la estrategia óptima a utilizar en cada zona teniendo en cuenta las particularidades de cada zona, por lo que es frecuente que en cada zona se aplique un algoritmo corrector diferente.

El hecho de utilizar varios tipos diferentes de contornos activos dentro del algoritmo de segmentación CAAR, le permite que pueda adaptarse a una gran variedad de situaciones diferentes. Además, en el futuro, si se encontrasen situaciones nuevas con un tipo de dificultad no encontrada hasta ese momento, se podría adaptar el CAAR a ese nuevo ámbito de aplicación cambiando los algoritmos de contornos activos involucrados dentro del CAAR.

Para comprobar la exactitud de los resultados arrojados por la metodología desarrollada, se la ha comparado de forma experimental con varios tipos de algoritmos de contornos activos ya existentes y se han obtenido mejores resultados con los CAAR. Para esta comparación se han utilizado, entre otros, el algoritmo clásico de contornos activos de Kass, el algoritmo de Bernard, los contornos activos sin bordes, el algoritmo de Li, los contornos activos geodésicos, los Balloons, el algoritmo de Lankton y el algoritmo de Shi. Esta comparación se ha realizado en base a una serie de métricas que describen el contorno de forma objetiva.

Abstract

The main objective of this work has been the development of a methodology for the three-dimensional reconstruction of anatomical structures from medical images that will subsequently facilitate quality, three-dimensional models. Within the complete process of three-dimensional reconstruction, one of the most important and complex stages is the segmentation of each medical image. Therefore, our work has focused on the phase related to the initial segmentation of medical images, since the results of this phase largely determine the accuracy of the three-dimensional models and thus the final three-dimensional images to be shown to the healthcare professional in charge of interpreting these images.

We present a new segmentation method based on active contours which we have named Self-Reconfigurable Active Contours (SRAC), a type of active contour that has two phases. In the first phase an active contour algorithm is run, the results of which are checked and enhanced automatically during the second phase. This methodology is based on automatically detecting the areas where good results have not been obtained. Then the optimal strategy to use in each area is chosen, also

automatically, taking into account the distinctive features of the individual area, which commonly results in a different corrective algorithm being applied to each.

The use of several different types of active contour within the SRAC segmentation algorithm enables it to adapt to a large variety of situations. Moreover, in the future, if we encounter a new situation with a kind of complication not seen previously, we could easily adapt the SRAC to the new situation by modifying the active contour algorithms within the SRAC.

To check the accuracy of the results obtained from the developed methodology, they have been compared experimentally with various existing active contour algorithms and the best results were obtained with SRAC. The comparisons were made using, among others, the classic Kass active contour algorithm, the Bernard algorithm, active contours without edges, the Li algorithm, geodesic active contours, the Lankton algorithm, the Balloon model and the Shi algorithm. These comparisons were made based on a set of metrics that describe the contours in an objective manner.

Índice

Capítulo 1. Introducción	1
1.1. Segmentación	2
1.1.1. Segmentación de imágenes basada en contorno activos	3
1.1.2. Objetivos iniciales de este trabajo	3
1.1.3. Organización de la memoria	4
Capítulo 2. Antecedentes	7
2.1. Introducción	7
2.2. Segmentación de imágenes	8
2.2.1. Métodos basados en contornos	8
2.2.1.1. Detección del contorno basado en el gradiente	9
2.2.1.2. Detección del contorno basado en la laplaciana	12
2.2.2. Métodos basados en regiones	14
2.2.2.1. Umbralización	14
2.2.2.1.1. Umbralización basada en la búsqueda de mínimos	16
2.2.2.1.2. Umbralización basada en las técnicas de reconocimiento de formas	16
2.2.2.2. Clustering	17
2.2.2.3. Crecimiento de regiones	17
2.3. Segmentación de imágenes basada en contornos activos	19
2.3.1. Introducción a los contornos activos	19
2.3.1.1. Definición formal del procedimiento	21
2.3.1.2. Energía Interna	23
2.3.1.3. Energía Externa	25
2.3.1.3.1. Fuerzas de restricción	26
2.3.1.3.2. Energía de línea	26
2.3.1.3.3. Energía de las fronteras	26
2.3.1.3.4. Energía de terminación	27
2.3.1.4. Definición Discreta	28

2.3.1.5. Técnicas de optimización	32
2.3.2. Balloons	36
2.3.3. Snakes eléctricos	37
2.3.4. Contornos activos borrosos	40
2.3.5. Contornos activos duales.	41
2.3.6. Snakes topológicamente adaptables.	45
2.3.7. Superficies deformables adaptables (T-surfaces).	54
2.3.7.1. Formulación de las T-surfaces	54
2.3.7.2. Descomposición de la imagen por celdas afines en tres dimensiones	56
2.3.8. Snakes estadísticos	58
2.3.9. Modelo de contorno activo adaptable a la textura.	63
2.3.10. Snakes Virtuales	67
2.3.11. B-Splines snakes	72
2.3.12. Contornos activos paramétricos	77
2.3.13. Sectorred snakes	79
2.3.14. Watersnakes	80
2.3.15. Level Set	82
2.3.16. Contornos activos geodésicos	84
2.3.17. Contornos activos sin bordes	85
2.3.18. Método de Li	87
2.3.19. Método de Bernard	88
2.3.20. Método de Shi	89
2.3.21. Método de Lankton	90
2.3.22. Snakes a Nivel de Píxel.	92
2.4. Conclusiones	93
Capítulo3. Metodología.	95
3.1. Introducción	95
3.2. Segmentación con contornos activos	95
3.2.1. Problemas encontrados en contornos activos actuales	96
3.2.1.1. Problemas encontrados en algoritmo de Caselles	96

3.2.1.2. Problemas encontrados en algoritmo de Chan	104
3.2.1.3. Problemas encontrados en algoritmo de Lankton	113
3.2.1.4. Problemas encontrados en algoritmo de Bernard	123
3.2.1.5. Problemas encontrados en algoritmo de Shi	127
3.2.1.6. Problemas encontrados en algoritmo de Li	134
3.2.2. Segmentación con Contornos Activos propuestos	139
3.2.2.1. Contornos Activos Auto-Reconfigurables	141
3.2.2.2. Contornos Activos Reconfiguradores	143
3.2.2.3. Contornos Activos con Reclamos Automáticos	144
3.2.2.4. Pushing Active Contours	147
3.2.2.5. Sistemas de Selección de Estrategia Reconfiguradora	150
3.2.2.6. Búsqueda Longitudinal en Abanico	150
3.2.2.7. Análisis de las Cualidades de la Zona (ACZ)	156
Capítulo 4. Resultados y Discusión	159
4.1. Introducción	159
4.2. Resultados	159
4.2.1. Ejemplos de ejecución de todos los métodos para la imagen 1	161
4.2.2. Ejemplos de ejecución de todos los métodos para la imagen 2	172
4.2.3. Estudio conjunto de los resultados	184
4.3. Discusión	189
Capítulo 5. Conclusiones	193
Capítulo 6. Trabajos futuros	197
Bibliografía	199

Capítulo 1

Introducción

En la sociedad actual, uno de los pilares principales del llamado estado del bienestar es la sanidad. El nivel de desarrollo científico y tecnológico de la sanidad de nuestros días ha conseguido aumentar considerablemente la esperanza de vida de la población en los países desarrollados. Y dentro de la medicina actual, una de las pruebas diagnósticas que han contribuido considerablemente en las últimas décadas a la mejora del diagnóstico médico, son las pruebas basadas en técnicas radiológicas y en las imágenes tridimensionales (3D) que se pueden conseguir a partir de las primeras.

El proceso de conseguir una imagen 3D suele pasar por crear previamente una estructura 3D, que suele construirse a partir de un conjunto de datos 2D o 3D [UDUPA00]. Este proceso recibe, entre otros, el nombre de Reconstrucción 3D o Reconstrucción Volumétrica. La reconstrucción 3D es una fase muy importante en el proceso de conseguir buena calidad en las imágenes 3D finales. Después de conseguir una reconstrucción 3D, tiene que ser abordada una fase posterior de visualización.

La reconstrucción volumétrica ha sido usada en sanidad, entre otras aplicaciones, para ayudar a diagnosticar, simular, facilitar la planificación quirúrgica o desarrollar cirugía guiada por imagen.

Para obtener una estructura 3D final, es necesario extraer (segmentar) la región de interés desde la cantidad de información total, es decir, debemos aislar las regiones o características de interés para facilitar el proceso completo de reconstrucción volumétrica. Este paso de extracción de la región de interés es más conocido como segmentación. Por tanto, podemos decir que la segmentación es una fase muy importante para conseguir una buena reconstrucción 3D.

1. Segmentación

La segmentación es habitualmente una fase de procesamiento inicial para extraer regiones o características de interés para otra tarea posterior (por ejemplo, reconstrucción 3D). Una introducción general a este tema puede encontrarse en [HARALICK92], [GONZALEZ92], [RUSS99], [AYACHE95], [UDUPA00].

Muchas aplicaciones en el procesamiento de imágenes médicas se basan en la definición fiable de los contornos de los objetos [RANGANATH95]. La extracción de estos contornos se lleva a cabo en la fase de segmentación. Por lo tanto, la segmentación es generalmente necesaria para las operaciones de formación de imágenes en 3D. Junto con otros autores, consideramos que esta es la parte más difícil de todas las operaciones en las imágenes 3D [UDUPA00b]. Por lo tanto, consideramos que la segmentación es el paso más difícil en el proceso de reconstrucción 3D. Por esta razón, en este trabajo nos vamos a centrar principalmente en la segmentación.

Para la extracción de contornos dentro de imágenes médicas, algunas de las aplicaciones de segmentación más exitosas se han hecho usando contornos activos. [DUNCAN00] Estas técnicas se han utilizado para la interpretación de imágenes médicas, por ejemplo, Amini et al. [AMINI91], Ayache et al. [AYACHE92] y Cootes et al. [COOTES94].

1.1. Segmentación de imágenes basada en contornos activos

En los últimos años, los contornos activos han logrado un gran éxito en la segmentación en general [MCINERNEY96], [BLAKE98] y en la segmentación de imágenes médicas [RANGANATH95], [HONEA97], [HUG99], [DELINGETTE01] en particular.

Al principio aparecieron dos enfoques principales de contornos activos. Primero aparecieron los contornos activos con una forma explícita (o paramétrica). Los primeros científicos que usaron un enfoque paramétrico fueron Kass, Witkin y Terzopoulos [KASS88]. Después de eso, aparecieron autores que presentaron diferentes variaciones, por ejemplo, Cohen et al [COHEN92] introdujeron el uso de modelos de elementos finitos, Hug et al . [HUG99] utilizaron por primera vez curvas de subdivisión y Metaxas y Terzopoulos concibieron los modelos analíticos [METAXAS91].

El segundo enfoque fueron los contornos activos en forma implícita. La representación de un contorno activo implícito fue introducido por Malladi et al. [MALLADI95] y posteriormente fue utilizado por Sethian [SETHIAN96] y muchos otros. Algunos de ellos fueron Caselles et al. [CASELLES97], que introdujo el concepto de contornos activos geodésicos utilizando un enfoque muy similar.

1.2. Objetivos iniciales de este trabajo

Los objetivos que se plantearon inicialmente para esta tesis eran, básicamente, los siguientes:

- Desarrollar una metodología para la reconstrucción volumétrica tridimensional de estructuras anatómicas a partir de imágenes médicas con la que se pudiesen obtener modelos tridimensionales de calidad. Para ello, se planteaba como tarea de especial importancia la segmentación de las imágenes médicas, puesto que el

resultado de esta fase determinaría en gran medida la precisión de los modelos tridimensionales y, por tanto, de las imágenes tridimensionales finales que se mostrarían al profesional sanitario encargado de interpretar dichas imágenes. Por esta razón, se decía que este trabajo se iba a centrar en el proceso de segmentación de las imágenes médicas iniciales.

También se plantearon, como objetivos más concretos, conseguir:

-Segmentación de imágenes médicas orientada a obtener contornos de calidad que definiesen las estructuras anatómicas de interés.

-Optimización de la fase de segmentación. Para esto se estudiarían varios métodos de segmentación, comparando los resultados. Mediante esta comparación se esperaba determinar que técnica de segmentación era la mejor para conseguir los objetivos anteriores. Para ello, se estudiarían los métodos existentes con mayor auge en este tipo de segmentación como, por ejemplo, los contornos activos o snakes.

1.3. Organización de la memoria

El presente documento está compuesto por 6 capítulos (incluido el presente), donde se muestran los objetivos iniciales de esta investigación, los fundamentos teóricos, la metodología que se ha seguido, los resultados obtenidos, las conclusiones a las que se ha llegado al final de este trabajo, así como algunas líneas que podrían seguirse en el futuro para continuar la presente investigación.

En el primer capítulo, en el que nos encontramos, se presenta una introducción general, indicando los objetivos iniciales de este trabajo, y una breve introducción a los puntos principales de la investigación que nos ocupa.

En el segundo capítulo, Antecedentes, se revisan las técnicas más utilizadas en el campo de la segmentación de imágenes digitales y, más brevemente, se hace una introducción sobre técnicas de reconstrucción tridimensional.

En el tercer capítulo, Metodología, se muestran el conjunto de métodos que se han desarrollado y seguido para alcanzar los objetivos de nuestra investigación.

En el cuarto capítulo, Resultados y Discusión, se muestran algunas de las pruebas experimentales realizadas y sus resultados correspondientes, pasando posteriormente a discutir los resultados obtenidos.

En el capítulo 5, Conclusiones, se muestran las conclusiones a las que se han llegado al final de este trabajo y comentaremos las principales aportaciones del mismo.

En el capítulo 6, Trabajos futuros, comentaremos algunas líneas de trabajo que se podrían seguir después de la finalización de esta tesis.

En la sección Bibliografía se muestran los recursos bibliográficos a los que se ha referenciado durante los capítulos anteriores.

Capítulo 2

Antecedentes

2.1. Introducción

En este capítulo vamos a hacer una revisión de los antecedentes de las técnicas sobre las que nos hemos basado para desarrollar nuestra metodología. En primer lugar haremos una revisión de los métodos más habituales en la segmentación de imágenes, pasando después a una revisión más profunda de las técnicas existentes de contornos activos.

2.2. Segmentación de imágenes

Las técnicas de segmentación más comúnmente utilizadas sobre imágenes médicas se pueden clasificar en dos tipos [UDUPA90]: los métodos basados en detección de contornos (también llamados métodos de detección de fronteras), que son aquellos que directamente producen una estructura de información de contornos y los llamados enfoques basados en regiones, que son aquellos que producen información sobre el espacio ocupado por la estructura. En el primer caso, la salida del proceso de segmentación es la estructura del contorno. En el segundo caso, la salida es una escena en la que se distinguen los píxeles que forman parte del espacio ocupado por la estructura y los que no.

2.2.1. Métodos basados en contornos

En la segmentación de imágenes médicas, si la modalidad de imagen utilizada para estudiar la anatomía del paciente permite distinguir varias regiones de tejidos dentro de la parte del cuerpo escaneada, se intenta discernir el contorno de la estructura anatómica de interés respecto de las regiones de otros tejidos. De forma más genérica, para todo tipo de imágenes, se le suele llamar objeto de interés (OI).

El cálculo de los contornos de los tejidos en la imagen se puede plantear como una operación de filtrado de la imagen original, más concretamente, de un filtrado de paso alto. Por tanto, puesto que los píxeles situados en los bordes del objeto presentan grandes variaciones en el nivel de intensidad en zonas concretas de la imagen, es posible aplicar un filtro de paso alto (que solamente deje pasar las altas frecuencias de la señal filtrada) pues, en tal caso, en la imagen resultante aparecerán resaltados los píxeles de los bordes. En este caso, el problema radica en determinar la frecuencia de corte más apropiada para localizar el contorno del objeto.

Así, para detectar los bordes, se suelen aplicar filtros lineales de dos tipos [GONZALEZ92]: basados en el gradiente de la función de luminosidad o basados en su laplaciana.

2.2.1.1. Detección del contorno basado en el gradiente

El operador gradiente de una función $f(x,y)$ bidimensional es el vector:

$$\nabla[f(x,y)] = f_G(x,y) = \frac{\partial f}{\partial x} \hat{x} + \frac{\partial f}{\partial y} \hat{y} \quad (2.1)$$

Siendo ∇ el símbolo del operador gradiente y \hat{x} e \hat{y} los vectores unitarios en ambas direcciones.

Ya que no es posible aplicar la definición estricta del operador gradiente, dada por la expresión anterior, a una imagen digital, puesto que las variables independientes x e y son discretas. Por tanto, es obligado emplear aproximaciones discretas del gradiente.

Una aproximación simple es:

$$\begin{aligned} G_x(x,y) &= \frac{\partial I(x,y)}{\partial x} \approx I(x,y) - I(x-1,y) \\ G_y(x,y) &= \frac{\partial I(x,y)}{\partial y} \approx I(x,y) - I(x,y-1) \end{aligned} \quad (2.2)$$

$$0 \leq x \leq N-1 \quad 0 \leq y \leq N-1$$

y esta aproximación al gradiente puede expresarse como una operación de filtrado espacial sin más que establecer los siguientes núcleos o máscaras de convolución de dimensión 3x3 [HARALICK92]:

$$\begin{array}{ccc} 0 & -1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{array} \quad \begin{array}{ccc} 0 & 0 & 0 \\ -1 & 1 & 1 \\ 0 & 0 & 0 \end{array}$$

h1 h2

Siendo por tanto:

$$G_x(x, y) = I(x, y) * h_1(x, y); 0 \leq x \leq N-1, 0 \leq y \leq N-1$$

$$G_y(x, y) = I(x, y) * h_2(x, y); 0 \leq x \leq N-1, 0 \leq y \leq N-1 \quad (2.3)$$

Estas máscaras h_1 y h_2 , centradas en cada píxel genérico filtrado, tienen un valor nulo fuera de su propia ventana de existencia, de dimensión 3×3 . Y puesto que no se suele considerar la dirección del gradiente, limitándose el filtrado a manejar el módulo del mismo, la salida del filtro espacial anterior es:

$$|I_G(x, y)| = \sqrt{G_x^2(x, y) + G_y^2(x, y)}; 0 \leq x, y \leq N-1 \quad (2.4)$$

Así, desarrollando la expresión anterior y aproximando el módulo del gradiente para disminuir la carga computacional, llegamos a las siguientes expresiones [Maravall,93]:

$$G_x = \frac{\partial I(x, y)}{\partial x} \approx [I(x+1, y-1) + 2I(x+1, y) + I(x+1, y+1)] - [I(x-1, y-1) + 2I(x-1, y) + I(x-1, y+1)]$$

$$G_y = \frac{\partial I(x, y)}{\partial y} \approx [I(x-1, y+1) + 2I(x, y+1) + I(x+1, y+1)] - [I(x-1, y-1) + 2I(x, y-1) + I(x+1, y-1)] \quad (2.5)$$

que dan lugar a los siguientes núcleos del filtro espacial:

-1	-2	-1
0	0	0
1	2	1

G_x

-1	0	1
-2	0	2
-1	0	1

G_y

Así, dependiendo de la configuración de estas matrices, podemos construir diversos filtros:

Filtro de Roberts

Este filtro se propuso por primera vez en un trabajo pionero de Roberts [ROBERTS65]. Es el más sencillo de todos:

$$\begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \quad \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$$

G_x G_y

Filtro de Sobel

Este filtro coincide con la expresión 2.5 que ya vimos anteriormente, pero introduciendo además un factor de escala de normalización[SOBEL73] :

$$\frac{1}{4} \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix} , \quad \frac{1}{4} \begin{pmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{pmatrix}$$

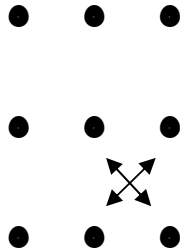
G_x G_y

Filtro de Prewitt

$$\frac{1}{3} \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix} , \quad \frac{1}{3} \begin{pmatrix} 1 & 0 & -1 \\ 1 & 0 & -1 \\ 1 & 0 & -1 \end{pmatrix}$$

G_x[PREWITT70] G_y[PREWITT70]

Maravall [MARAVALL93] también presenta varios filtros no lineales y que pueden dar buenos resultados. Por ejemplo, el siguiente filtro tiene una máscara no lineal que compara los píxeles en sentido diagonal:



que se puede expresar de la forma:

$$G_x(x,y) = I(x+1, y+1) + I(x,y) \quad (2.6)$$

$$G_x(x,y) = I(x, y+1) - I(x+1,y)$$

$$I_G(x,y) = \max \{ |G_x(x,y)|, |G_y(x,y)| \}$$

2.2.1.2. Detección del contorno basado en la laplaciana

El operador laplaciano, o simplemente laplaciana, de una función bidimensional $f(x,y)$ es el escalar:

$$\Delta[f(x,y)] = \nabla[\nabla(f(x,y))] = f_L(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2} \quad (2.7)$$

Es decir, la laplaciana de una función 2-D $f(x,y)$ es una derivada segunda que cuantifica el cambio del cambio de una función, o lo que es lo mismo, el cambio de una derivada. Por esto, es muy sensible al ruido. Es decir, la laplaciana de un píxel contaminado con ruido puede ser muy elevada, con lo que causaría problemas para detectar los contornos. Por esto, se suele aplicar junto con operadores del tipo gradiente.

Para utilizar la laplaciana con imágenes digitales $I(x,y)$, debemos hacer una aproximación a la laplaciana, igual que hacíamos con el gradiente:

$$\Delta(I(x, y)) \approx I(x-1, y) + I(x, y-1) + I(x, y+1) + I(x+1, y) - 4I(x, y) \quad (2.8)$$

lo que se podría implementar mediante la máscara:

0	1	0
1	-4	1
0	1	0

Por último, resumiendo el apartado 2.2.1, podemos decir que la segmentación de imágenes digitales mediante el método de detección de contornos se puede dividir en los siguientes pasos: el primero es una etapa de preprocesado o acondicionamiento de brillo y/o contraste. El segundo paso, es el cálculo de la derivada primera o segunda de la señal. Finalmente, dependiendo de la derivada usada, la primera o la segunda, se marcan como píxeles del contorno los máximos locales o los cruces por cero respectivamente [GONZALEZ92].

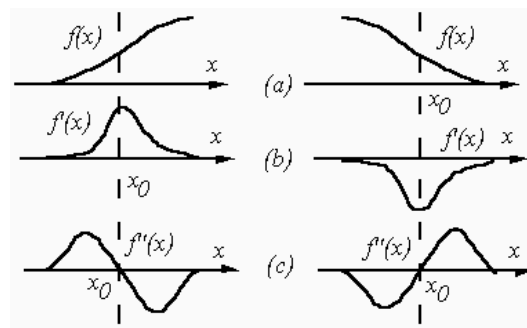


Figura 2.1. Uso de la derivada para la detección de contornos

El resultado de la detección de bordes es una imagen binaria indicando donde están los bordes o una lista de bordes, opcionalmente con descripciones adjuntas tales como la dirección y el contraste [LIANG93]. Por ejemplo, dio muy buenos resultados en segmentación de imágenes RM el operador de detección de bordes del parámetro derivativo de Laplace utilizando interacción con el usuario [MARR80] [BOMANS90].

2.2.2. Métodos basados en regiones

La premisa en los métodos basados en regiones es que los píxeles que están dentro de la estructura de interés pueden ser distinguidos de aquellos que están fuera o de aquellos que les rodean mediante un conjunto de características o propiedades [UDUPA90]. Por tanto, el problema a resolver en este tipo de segmentación puede considerarse como determinar si cada píxel pertenece a la estructura o no. En la mayoría de las aplicaciones de este tipo de métodos, el resultado de este proceso es una imagen binaria donde una densidad de 1 indica que el píxel pertenece a la estructura y una densidad de 0 significa que el píxel está fuera de la estructura.

2.2.2.1. Umbralización

El más simple de los métodos basados en regiones es la umbralización [SAHOO88]. En 1972 Chow y Kaneko [CHOW72] usaron umbralización para segmentar ventrículos a partir de cineangiogramas del corazón humano.

En este método se considera que un píxel pertenece a la estructura si una cierta característica del píxel sobrepasa un valor fijado como umbral [PREWITT66]. La característica puede ser la densidad del píxel o puede ser una entidad derivada de la distribución de densidad del píxel tal como la densidad media calculada en la vecindad de cada píxel. Una forma más general de umbralización es usar un umbral inferior y un umbral superior y clasificar el píxel como perteneciente a la estructura si el valor de la característica asociada con el píxel está entre esos dos valores umbrales. A menudo es conveniente usar múltiples características y entonces usar un umbral inferior y otro superior para cada característica.

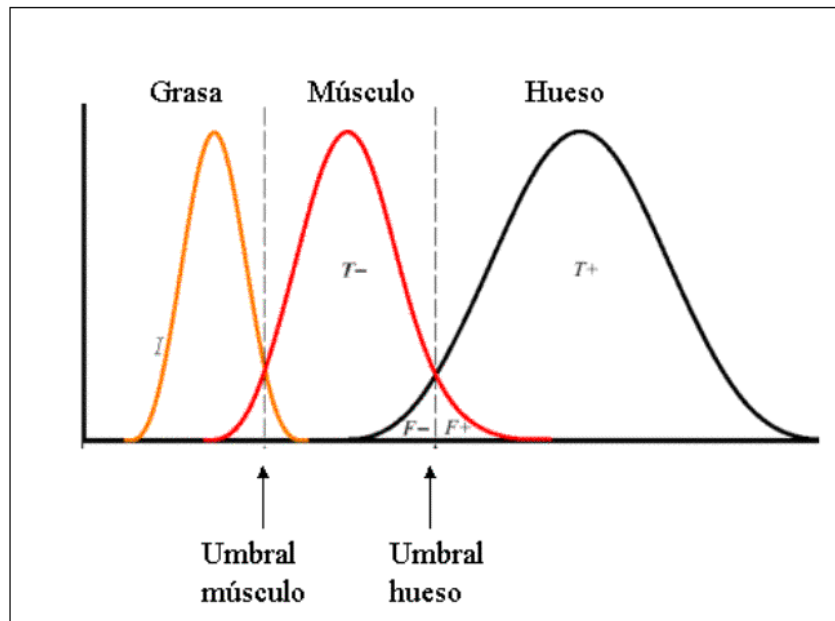


Figura 2.2. Una hipotética distribución de frecuencia $f(I)$ de valores de intensidad $I(x,y)$ para grasa, músculo y hueso en una imagen CT. Los valores de intensidad bajos corresponden a tejidos grasos, mientras que los valores de intensidad altos corresponden a valores correspondientes a estructuras óseas. Los valores de intensidad intermedios corresponden a tejidos de músculo. $F+$ son los falsos positivos y $F-$ hace referencia a los falsos negativos. $T+$ y $T-$ son los positivos correctos y los negativos correctos. Basado en [PICHUMANI97]

La umbralización basada exclusivamente en los niveles de luminosidad o de gris de los píxeles de la imagen se denomina global, en contraposición a la umbralización local en la que aparte de esa propiedad global de un píxel se utiliza información local del mismo; es decir, propiedades de un píxel que dependen de su localización dentro de la imagen [MARAVALL93].

Así, dentro de la umbralización global, podemos distinguir dos formas de proceder a la hora de implementar un método de umbralización: umbralización basada en la búsqueda de mínimos y la umbralización basada en las técnicas de reconocimiento de patrones o formas [MARAVALL93].

2.2.2.1.1. Umbralización basada en la búsqueda de mínimos

En este método se busca localizar niveles de intensidad luminosa que umbralicen adecuadamente el histograma de una imagen. En el histograma de una imagen, normalmente se pueden distinguir una serie de montículos o lóbulos que están formados por los niveles de intensidad de los píxeles de los objetos físicos similares y que, por tanto, se considera que pertenecen a un mismo objeto de la imagen. Idealmente, entre dos lóbulos consecutivos habrá un valle de separación entre ellos, el nivel de intensidad más bajo del valle, en principio, podría definirse como un umbral de separación entre dos lóbulos.

Pero en casos en los que haya ruido en la imagen, podrían aparecer falsos mínimos y falsos máximos, lo cual redundaría negativamente en la efectividad del procedimiento y muy especialmente en el tiempo de cálculo.

Para solucionar esto, se suelen aplicar técnicas de preprocesamiento encaminadas a disminuir el ruido (también llamado artefactos en la terminología utilizada en procesamiento de imágenes) y a aumentar el contraste.

2.2.2.1.2. Umbralización basada en las técnicas de reconocimiento de formas

Los lóbulos que se suelen encontrar en el histograma de una imagen recuerdan a la campana de Gauss, típica de las distribuciones normales. Así, utilizando este parecido, se puede considerar que cada lóbulo corresponde a la distribución de probabilidad de un objeto. Esto es válido en sentido estricto, ya que el histograma $h(p)$ cumple las restricciones de una función de densidad de probabilidad.

$$0 \leq h(p) \leq 1; \sum_{p=0}^{2^q-1} h(p) = 1 \quad (2.9)$$

Así, si tenemos dos lóbulos, que aproximamos mediante dos campanas de Gauss, estaremos modelando dos clases (es decir, los patrones de los dos objetos presentes en la imagen) como dos variables aleatorias normales o gaussianas.

El problema práctico a la hora de segmentar una imagen con este método es calcular las medias y las desviaciones típicas de cada lóbulo. Lo más difícil consiste en delimitar la amplitud de las campanas de Gauss, o lo que es lo mismo, el rango de los lóbulos.

2.2.2.2. Clustering

Otro método es el de agrupación de píxeles, más conocido como clustering. Este método consiste principalmente en definir un vector X de descriptores o características de los píxeles a segmentar en la imagen. Los primeros trabajos en la segmentación de imágenes mediante clustering son los de Duda y Hart [DUDA73] y el de Coleman y Andrews [COLEMAN79].

2.2.2.3. Crecimiento de regiones

El crecimiento de regiones consiste en elegir una serie de píxeles iniciales, denominados puntos semilla, a los que se les irán añadiendo píxeles de propiedades similares. Para ello se examina si los píxeles contiguos tienen valores parecidos para la o las características que tienen los puntos semilla. Si es así, pertenecen a la misma región y pasan a tener los mismos valores que los puntos iniciales. Es decir, si se cumple que la diferencia entre los niveles de gris, por ejemplo, es menor que un valor prefijado [ESCALERA01]:

$$|f(x, y) - f(s_{xi}, s_{yi})| < T \quad (2.10)$$

pertenecerán al mismo objeto que ese punto semilla. Esto se comprueba para los ocho vecinos para después comparar cada uno de los vecinos de los píxeles que se hayan asignado a alguna región. Es un método iterativo que termina cuando ningún píxel cambia sus propiedades.

El primer problema a resolver es cómo determinar los puntos iniciales. Una posibilidad es elegir píxeles cuyo nivel de gris corresponda a picos del histograma. Otra posibilidad es obtener esos píxeles mediante un método previo. Estos métodos previos están basados en la mayor parte de los casos en técnicas de crecimiento de regiones, orientados, o bien a crecimiento de regiones que trabajan con una propiedad concreta de las mismas, o bien a crecimiento de regiones basado en detectores de bordes. Los primeros se basan en que los puntos vecinos de una misma clase tienen valores similares. Si al comparar los puntos, la diferencia en esa característica es menor a un valor establecido, se considera que pertenecen a la misma región.

Los métodos de crecimiento de regiones basados en detectores de bordes se basan en que los valores de los puntos que se encuentran en el contorno del objeto cambian bruscamente. Así, primero se aplicaría alguno de los métodos de detectores de contornos que se han estudiado en la sección 2.2.1 como, por ejemplo, los basados en la laplaciana o en el gradiente, como los métodos de Roberts[ROBERTS65], Sobel [SOBEL73] o Prewitt [PREWITT70].

Otra posibilidad es el uso de la llamada técnica watershed que consiste en, aplicar un detector de contornos a la imagen y obtener otra que puede ser asimilada a un relieve que se va inundando con agua hasta localizar los puntos que separan los objetos que hay en la imagen.

2.3. Segmentación de imágenes basada en contornos activos

2.3.1. Introducción a los contornos activos

Los contornos activos fueron propuestos por primera vez por Kass, Witkin y Terzopoulos en 1987 en la 1ª Conferencia internacional sobre visión por ordenador [KASS87], y que fue posteriormente publicado en enero de 1998 en el International Journal of Computer Vision [KASS88].

De forma intuitiva los modelos deformables, contornos activos o snakes, son contornos que se ajustan a las fronteras de los objetos bajo la influencia de dos fuerzas: fuerzas externas que dirigen el contorno hacia ciertas características de la imagen tales como los bordes, y fuerzas internas que actúan como restricciones para mantener la suavidad de la curva, impidiendo en mayor o menor medida que la curva se deforme bruscamente, o como restricciones complementarias del modelo. Su comportamiento es de tal manera que en cada iteración el contorno activo se va aproximando poco a poco a la frontera que se pretende detectar con una forma de moverse que recuerda al de una serpiente, de ahí su denominación de snake. Cuando se empieza a detectar este contorno, el snake se aproxima poco a poco hasta alcanzar la forma final, que será más o menos exacta según el tipo de contorno activo utilizado y según la configuración de sus distintos parámetros.

El ‘snake’ tradicional formulado por Kass et al. [KASS88] es la solución a un problema de minimización funcional de las ecuaciones de Lagrange. Éste es un método que puede adaptarse dinámicamente a las características de la imagen y por tanto capaz de reconocer sus diferentes contornos.

Los ‘snakes’ significan un cambio del clásico punto de vista de la detección de bordes y la segmentación (primer nivel de la visión artificial), en el que la detección de bordes es un proceso de abajo-arriba (bottom-up) y en el que las características extraídas en el primer paso se deben tratar, interpolando normalmente, por un proceso de nivel superior para obtener una representación. Esta situación presenta

una gran desventaja dado que la aproximación se consigue tras varios procesos, lo que hace arrastrar errores generados en fases previas sin posibilidad de corrección.

Como ya se ha comentado, el ‘snake’ es una curva activa que reacciona ante las características de la imagen que va a segmentar delimitando el borde de una determinada región dentro de dicha imagen. El desplazamiento de esta curva activa por la imagen se realiza teniendo en cuenta los valores de una función de energía. Esta función de energía está determinada por dos factores: la forma de la curva y las características locales de la imagen.

Los contornos activos buscan puntos con un valor alto de gradiente. Se dan dos problemas principales, que pueden perjudicar a esta búsqueda: la presencia de ruido en la imagen, que puede hacer caer al modelo en mínimos locales; y el dominio de atracción de los bordes (zona donde el borde del objeto atrae con más fuerza al snake para ajustarlo a su forma), que si es demasiado pequeño puede hacer que el snake no encuentre la solución deseada. Estos problemas pueden solucionarse filtrando la imagen con un filtro gaussiano de paso bajo, antes de realizar ninguna operación sobre ella. Los modelos deformables, como hemos visto, actúan buscando una serie de características relevantes en la imagen, para lo cual deben examinar el gradiente de la misma. Pueden usarse varias aproximaciones para el cálculo del gradiente. Lo más común es utilizar la magnitud del gradiente de la imagen.

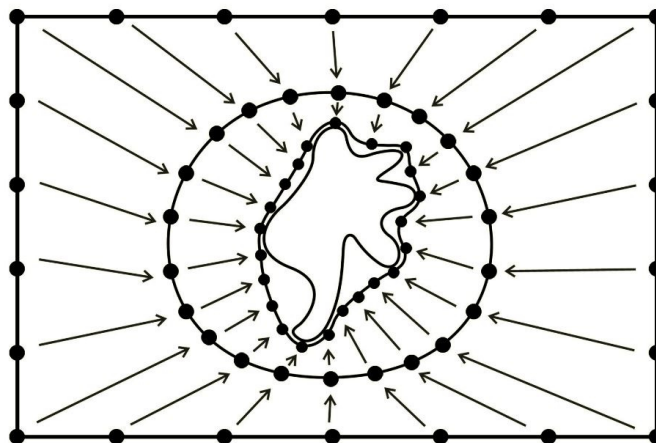


Figura 2.3. Evolución de un contorno activo que ha sido inicializado con forma de rectángulo y después se muestran varios pasos intermedios en los que va aproximándose progresivamente a la frontera del objeto.

Desde que se crearon los contornos activos, se ha logrado con ellos un gran éxito en la segmentación de imágenes en general [MCINERNEY96], [BLAKE98] y de imágenes médicas en particular [RANGANATH95], [HONEA97], [HUG99], [DELINGETTE01]. Como ejemplo ilustrativo podemos mencionar la obra de Marais et al., [MARAIS96] donde se miden la asimetría de los hemisferios cerebrales del cerebro (figura 3.A).

2.3.1.1. Definición formal del procedimiento

En este punto vamos a mostrar formalmente el funcionamiento y propiedades de los contornos activos. El método comienza con la inicialización del contorno activo, que será introducida el usuario, salvo en algunas aplicaciones donde se trabaje siempre con unas condiciones muy constantes (en este caso se puede inicializar el contorno en una posición fija). Cuando el trabajo global se refiera a una pila de imágenes (por ejemplo, las conseguidas en un tomógrafo axial computerizado) esta operación sólo se realizará en la primera de ellas, con el resultado de ésta se inicializará la siguiente y así sucesivamente.

En el siguiente paso, las fuerzas que actúan sobre el contorno activo atraen a ese contorno hacia las características de interés de la imagen. Este proceso finaliza cuando se minimiza la energía total del snake.

Primero vamos a definir cada uno de los términos de la función de energía y posteriormente describiremos el método utilizado para minimizar esta función de energía.

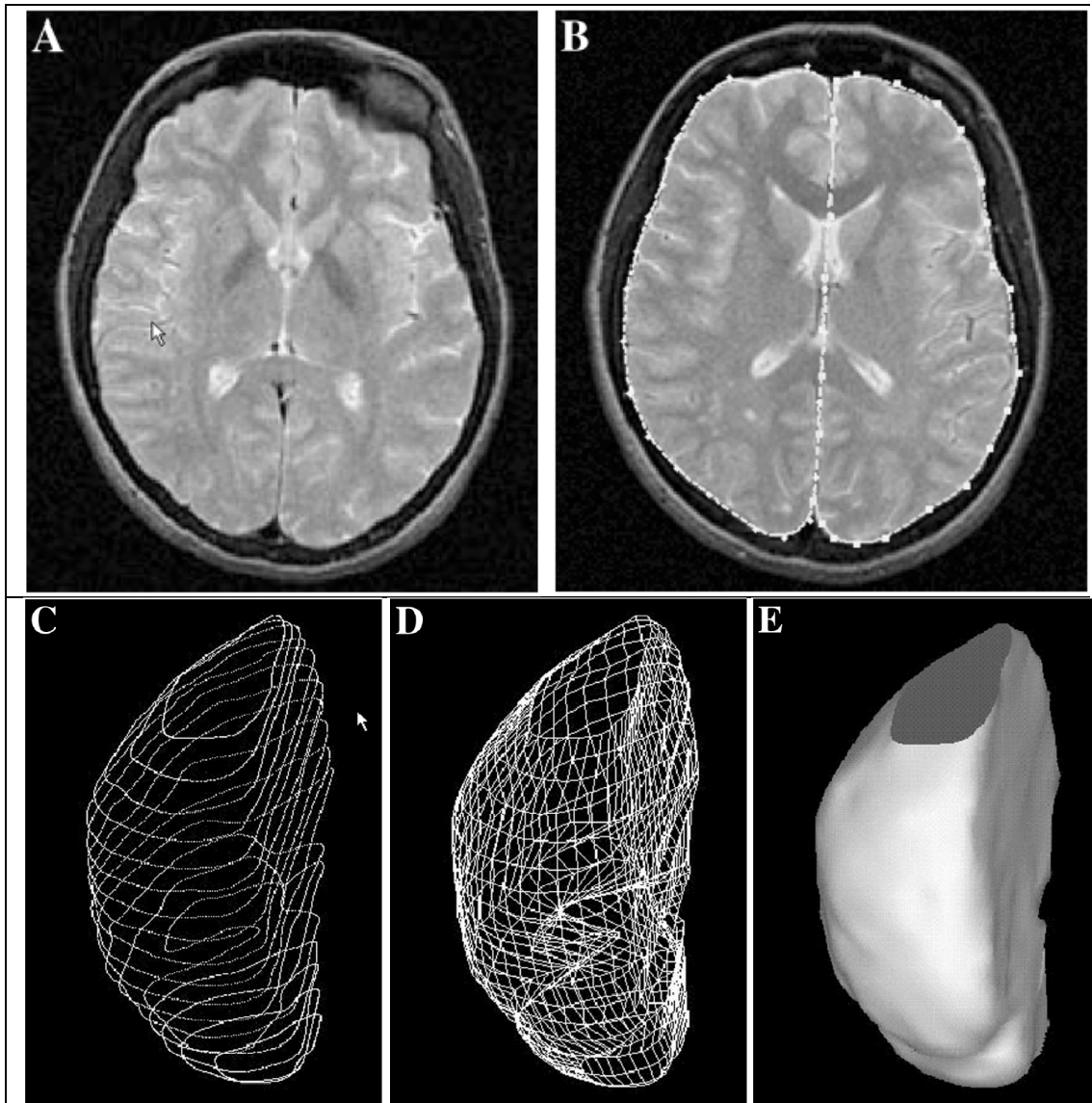


Figura 2.4. Reconstrucción tridimensional de un hemisferio cerebral usando contornos activos (imágenes desarrolladas en la Universidad de Oxford, cortesía de Patrick Marais). A) Una sección bidimensional de una resonancia magnética. B) Dos hemisferios segmentados usando contornos activos. Los contornos activos fueron adelgazados para permitir una mejor visualización. C) Contornos 2D obtenidos en la fase de segmentación. Se muestran los contornos activos que constituyen la frontera del hemisferio derecho. D) Malla 3D obtenida después de la reconstrucción volumétrica a partir de contornos 2D obtenidos en la fase de segmentación. E) Visualización 3D del hemisferio cerebral [MARAIS96].

La forma del contorno que queremos discriminar se encontrará contenida en la imagen $I(x, y)$ y vendrá dada por la siguiente función, que puede entenderse como la representación de la energía que posee el contorno, correspondiendo la forma final del contorno al mínimo de esta energía.

$$E_{\text{snake}}(V(s)) = E_{\text{int}}(V(s)) + E_{\text{ext}}(V(s))$$

E_{int} es la energía interna del contorno activo, que representa la elasticidad del contorno, controlando el estiramiento y la resistencia de la curva a la deformación.

E_{ext} es la energía externa, que suele diseñarse de forma que atraiga al contorno activo hacia aspectos concretos de la imagen, normalmente la frontera de un objeto, que suele detectarse en función del gradiente en esa posición.

Por tanto, cada pixel de la imagen tiene el valor de energía resultante de aplicar la expresión anterior.

2.3.1.2. Energía Interna.

La energía interna se puede expresar como:

$$E_{\text{int}}(V(s)) = \int_0^1 \alpha(s) \left| \frac{\partial V(s)}{\partial s} \right| + \beta(s) \left| \frac{\partial^2 V(s)}{\partial s^2} \right|$$

y caracteriza un contorno flexible y que se puede estirar como una goma elástica. Dos funciones paramétricas determinan las características del contorno:

- $\alpha(s)$: controla la tensión del contorno influyendo en la primera derivada.
- $\beta(s)$: controla su rigidez ponderando la segunda derivada.

La mayoría de los trabajos existentes reducen tanto $\alpha(s)$ como $\beta(s)$ a funciones constantes (α y β), reduciendo así la complejidad del modelo sin que esto implique una reducción en la capacidad de detección del snake. Según los valores de α y β se pueden presentar las siguientes situaciones:

• $\alpha > 0, \beta = 0$: El valor positivo de α provoca que aparezca el factor de estiramiento en la curva, de manera que cada punto tenderá a estar lo más próximo posible a sus vecinos, ya que esta es la configuración que proporciona un valor más bajo de energía. Así tendremos un contorno continuo interactuando con la imagen y no solamente una colección de puntos aislados. Por otro lado, el valor nulo de β implica la no existencia en el modelo de imposiciones relativas a la curvatura que presenta el snake a lo largo de su contorno. Desde el punto de vista matemático nos encontramos con una discontinuidad de segundo orden, y desde el punto de vista gráfico permitimos la aparición de esquinas en el contorno del snake.

• $\alpha > 0, \beta > 0$: El valor positivo de β anula la discontinuidad de segundo orden, de manera que el snake generará curvas continuas y suaves (sin esquinas).

• $\alpha = \beta = 0$: La fuerza interna es nula. En este caso no existe nada que ligue los puntos del snake entre si de manera que el modelo deformable no actúa como tal, sino como un conjunto de puntos que interactúan individualmente con la imagen sobre la que se sitúan, siendo atraídos por el campo de energía que esta define. Nada impide que el snake acabe completamente fragmentado. Permite la aparición de discontinuidades de primer orden.

Dotando a α y β de valores negativos se obtendrían efectos contrarios. Cuanto más alto fuese el valor absoluto de α y β más tendencia tendrían los puntos del snake a enredarse unos con otros. Por otro lado, cuanto mayor fuese el valor absoluto de β , más probabilidad existiría de que las soluciones generadas definieran curvas puntiagudas, con grandes irregularidades en lugar de formas suaves.

En cualquier caso, la elección de los coeficientes de la energía interna se realiza en función de los intereses de cada caso, ajustándolos manualmente al dominio de trabajo. Es importante destacar también que debe existir cierta compensación entre estos coeficientes, de manera que ninguno de ellos eleve el valor de su término de energía asociado hasta el punto de hacer insignificante la aportación del otro, o incluso la propia energía externa de la imagen.

Un valor demasiado alto de α podría provocar que el snake se colapsara en un punto debido a que el factor dominante es aquel que tiende a mantener los puntos del

contorno lo más cercanos posible entre si. En el caso de un contorno circular, por ejemplo, este tendería a disminuir progresivamente su radio hasta acabar en un único punto.

Si le damos a β un valor muy alto, pero pequeño en comparación con α , el contorno activo ignoraría estructuras finas, pequeñas protuberancias y detalles. En caso contrario, un valor elevado de β llevaría al snake a obtener soluciones excesivamente suaves, perdiendo valles y crestas pronunciadas en el contorno del objeto, y obteniendo contornos demasiado burdos como para ser considerados buenas soluciones al problema.

Por tanto, la determinación de los valores de los parámetros α y β se lleva a cabo de forma empírica, modificando sus valores de manera sustancial cuando cambiamos de un dominio a otro, o incluso dentro de un mismo dominio si variamos el tipo de objetos a detectar o las condiciones del entorno.

2.3.1.3. Energía Externa.

En este apartado vamos a estudiar el modo en que diseñaron la energía externa Kass, Witkin y Terzopoulos[KASS88], en cuyo modelo se han basado tradicionalmente la gran mayoría de métodos de contornos activos posteriores, de ahí que se le suele llamar algoritmo "clásico" de contornos activos. Aunque la mayoría de los métodos de contornos activos propuestos posteriormente se diferencian unos de otros por la forma en que definen la energía externa.

En la ecuación de la función de energía del contorno activo aparece la energía externa E_{ext} , que de forma genérica se describe en el modelo original como:

$$E_{ext}(V(s)) = \int_0^1 E_{imagen}(V(s)) + E_{rest}(V(s))$$

Por lo que la fórmula general de energía del snake podría expresarse así

$$E_{snake}(V(s)) = \int_0^1 E_{int}(V(s)) + E_{imagen}(V(s)) + E_{rest}(V(s)) ds$$

A su vez el término de energía E_{imagen} se descompone en varios términos [KASS88]:

$$E_{imagen} = W_{linea}E_{linea} + W_{frontera}E_{frontera} + W_{term}E_{term}$$

Donde los distintos w_j representan los pesos asignados a cada término.

2.3.1.3.1. Fuerzas de restricción.

El término E_{rest} representa una serie de fuerzas de restricción o Energía de restricciones que permiten al usuario hacer que el snake sea atraído por unas zonas concretas de la imagen. Kass et al. [KASS88] comentan varias posibles fuerzas de este tipo, que son introducidas en la imagen artificialmente, como pueden ser muelles que atraigan hacia ellos al nodo del snake que se encuentre conectado a su extremo, que por otro lado estaría unido a algún punto característico de la imagen, atrayendo hacia él mismo al nodo (crear un muelle entre dos puntos x_1 y x_2 simplemente consiste en introducir como fuerza

$$E_{rest} = -K(x_1 - x_2)^2$$

2.3.1.3.2. Energía de línea.

La forma más simple es representar la función de energía de línea como la intensidad de la imagen $I(x, y)$. Dependiendo del signo del parámetro de ponderación w_{linea} puede hacer que el snake sea atraído por líneas claras u oscuras. Este término por sí sólo aporta información importante, pero presenta el inconveniente de que puede llevar al snake hacia puntos de ruido que pueden llegar a colapsar el modelo en un contorno falso.

2.3.1.3.3. Energía de las fronteras.

Encontrar fronteras en una imagen puede llevarse a cabo mediante una función muy simple, como puede ser la siguiente:

$$E_{\text{frontera}} = -|\nabla I(x, y)|^2$$

Estableciendo el valor de esta función, el snake se verá atraído por las fronteras o los bordes de los objetos que tienen un valor elevado en la magnitud de su gradiente. Este término presenta además un complemento bastante adecuado para el anterior, E_{linea} pues resulta mucho menos sensible al ruido que éste. Uno de los mayores problemas de los contornos activos es que son muy sensibles a la inicialización ya que son propensos a caer en mínimos locales. Una forma de evitarlo es realizar el ajuste del contorno de forma progresiva, trabajando sobre distintos niveles de resolución de la imagen (que es equivalente a trabajar a diferentes escalas). Para ello usamos la siguiente fórmula:

$$E_{\text{frontera}} = - (G_{\sigma}(x,y) * \nabla^2 I(x,y))$$

Donde G es una gaussiana de la desviación típica σ .

Los mínimos de esta función se encuentran en los cruces por cero. Posteriormente, basándose en esta variación, aparecerán los denominados modelos multiescala.

2.3.1.3.4. Energía de terminación.

Por último, para encontrar el final de los segmentos de línea y esquinas se hace uso de la curvatura de las líneas de nivel, en una imagen ligeramente suavizada. Sea $C(x, y) = G_{\sigma} * I(x, y)$ la versión suavizada de la imagen, con $\theta = \tan(C_x/C_y)$ el ángulo del gradiente, $n = (\cos \theta, \sin \theta)$ y $n_{\perp} = (-\sin \theta, \cos \theta)$ vectores unitarios perpendiculares a la dirección del gradiente.

Entonces la curvatura del contorno en $C(x, y)$ viene dada por:

$$E_{term} = \frac{\partial \theta}{\partial n_{\perp}} = \frac{\frac{\partial^2 C}{\partial n^2}}{\frac{\partial C}{\partial n}} = \frac{C_{yy}C_x^2 - 2C_{xy}C_xC_y + C_{xx}C_y^2}{(C_x^2 + C_y^2)^{\frac{3}{2}}}$$

Combinando E_{borde} con E_{term} hacemos que el snake sea atraído por los bordes o finales de segmento, puesto que ese es el cometido de las energías externas, atraer al contorno hacia los bordes de la imagen.

2.3.1.4. Definición Discreta

La formulación genérica de los snakes se basa en curvas continuas parametrizadas por la longitud del arco de curva s . Sin embargo, a la hora de abordar el problema desde el punto de vista computacional, se debe proceder a la discretización de los diferentes elementos con los que se trabaja. Esto es una consecuencia directa de que la imagen esté formada por elementos discretos, llamados píxeles, que crean un campo de potencial discreto: la energía externa asociada. A la ubicación de cada píxel sobre la imagen original se le hace corresponder un valor de energía determinado (típicamente un valor de gradiente o una medida de la distancia desde el píxel al punto de borde más cercano). Por otro lado, también es necesario discretizar el propio snake, transformando su contorno en una aproximación poligonal del mismo, V , formada por N vértices que denotaremos por v_i , $i = 0 \dots N - 1$. Cada vértice del snake tendrá por coordenadas

$$v_i = (x_i, y_i).$$

El problema se reduce a encontrar el conjunto V de nodos que minimicen la siguiente expresión:

$$\sum_i (E_{int}(v_i) + E_{ext}(v_i))$$

Con el fin de simplificar computacionalmente el problema, en muchas situaciones se incorporan al modelo restricciones que reducen de manera considerable el número de operaciones a realizar, considerando el proceso de forma iterativa. Una de estas restricciones consiste en limitar el movimiento de cada punto

del snake a una ventana de desplazamiento de forma que en cada iteración se pueda escoger entre m posibles localizaciones para cada punto. Esta ventana será de tamaño $m \times m$, con m impar, de forma que el vértice actual se sitúa en el centro de la ventana.

Así, para un total de N puntos tendríamos $m \times N$ evaluaciones de funciones.

Esto podría reducirse aún más si trabajamos con un algoritmo que optimice el desplazamiento de todos y cada uno de los vértices del snake. Por otro lado, el comportamiento dinámico del proceso de ajuste de contornos se puede caracterizar por dos etapas, la deformación y el remuestreo, que se repiten alternativamente:

- Deformación. Durante el proceso de deformación el modelo inicial se transforma moviendo los vértices en la dirección normal de cada uno de ellos.

La deformación se controla con la minimización de la función de energía asociada. Este proceso se detiene si todos los vértices alcanzan un estado de equilibrio, equivalente a un mínimo en la energía del sistema.

- Remuestreo. El proceso de remuestreo controla la resolución del modelo deformable. Esta resolución depende del número de vértices y del espacio entre ellos, y debe estar relacionada con la escala en la que se procesa la imagen. Por un lado, bajas resoluciones son útiles a escalas gruesas en las que sólo interesa extraer información global de la imagen. Por otro lado, altas resoluciones son útiles a escalas más finas, donde puede extraerse información mucho más detallada de la forma del objeto. Este proceso se encarga también de adaptar automáticamente la resolución a la escala de la imagen, fijando de antemano una distancia d_k . Para cada segmento de curva del modelo que tenga un tamaño $|Q_i| > d_k$ se genera un nuevo punto de control entre P_i y P_{i+1} . El valor del parámetro d_k tiene una gran influencia en la calidad del resultado final.

Las características más destacables del modelo clásico de snake de Kass et al. [KASS88] son:

- Modelo de procesamiento de arriba-abajo (“Top-down”).
- Uso de información global para actualizar los puntos del contorno.
- Método de cálculo variacional.

• Tiene una serie de restricciones “blandas” que son inherentes al modelo utilizado y que deben satisfacerse sólo en cierto grado, estas restricciones aparecen en la literatura como términos de penalización que se añaden a la función objetivo, como pueden ser fuerzas atractivas o repulsivas, de tal forma que la función de energía aparte de minimizarse tiene que cumplir estas restricciones.

Este modelo presenta una serie de ventajas dentro de la segmentación de imágenes que le han hecho cobrar importancia:

• El ‘*snake*’ puede ser manejado por un proceso de nivel superior para controlar su funcionamiento. Este proceso superior puede ser un sistema automático o bien la interacción humana.

• Rápida convergencia del modelo, ya que las fuerzas pueden propagarse por todo el contorno.

• Las fases de *extracción de características* y *de representación* están integradas en un sólo proceso, al contrario que en los métodos clásicos de segmentación.

No obstante, como cualquier nuevo modelo que se plantee, presenta también una serie de desventajas y limitaciones que han sido tema de estudio y controversia en este campo:

• Debe ser iniciado en una zona cercana a la región que se desea reconocer. En cualquier otro caso el modelo no llega a una solución óptima. Por esta razón necesita de una alta interactividad humana o la utilización de otro mecanismo superior de presegmentación que sitúe el contorno en la zona deseada.

• Las técnicas originales no ofrecen guías para la elección de los parámetros. Los valores son críticos y deben ser elegidos cuidadosamente.

• El número de puntos considerados en la curva afecta en gran medida al comportamiento del contorno, ya que un número elevado puede conducir a una falsa convergencia (finalización anticipada) y uno demasiado pequeño no conduce a un buen reconocimiento.

- Es necesario diferenciar entre energía de la imagen y restricciones “blandas” (deben satisfacerse sólo en cierto grado) para garantizar la convergencia. Debido a esto no se pueden añadir restricciones duras (en tiempo de desarrollo, estas restricciones deben satisfacerse completamente) en el modelo.

- Los resultados intermedios no son significativos.

- El uso de información global hace que la convergencia no esté garantizada.

- La formulación original no ofrece buenos resultados con formas no convexas.

El replanteo de las energías clásicas podría solucionar este problema.

Tras la exposición de este primer modelo [KASS88], **Amini, Tehrani y Weymouth** puntualizaron algunos fallos del modelo de **Kass et al** en la 2ª *Conferencia Internacional sobre Visión por Ordenador* [AMINI88]. Algunos de estos errores son:

- Inestabilidad computacional.

- Tendencia de los puntos a agruparse en las zonas de bordes pronunciados.

Para mejorar el primer modelo presentado, utilizaron un algoritmo de programación dinámica que permitiera la adición de “restricciones duras”, es decir, restricciones añadidas por el programador y ajenas al comportamiento implícito del snake que ayudan a controlar el movimiento del mismo.

Características más destacables:

- Usa técnicas de Programación Dinámica.

Ventajas de esta corrección sobre el modelo inicial:

- Mayor estabilidad.

- Los puntos se mueven de forma discreta, es decir, dentro de una rejilla definida.

- Permite introducir restricciones más duras, especialmente diseñadas por el programador para controlar el contorno.

Desventajas:

- La convergencia es más lenta y sigue sin estar garantizada.
- Todavía se debe inicializar en una zona cercana a la región que se desea reconocer.
- Tiene grandes requerimientos de memoria.

2.3.1.5. Técnicas de optimización

En este apartado veremos las diferentes técnicas de optimización que se pueden utilizar para optimizar (minimizar) la energía, siguiendo con el modelo de Kass, Witkin y Terzopoulos[KASS88]. En la búsqueda de un mínimo para el snake se hacen múltiples simplificaciones. En primer lugar se discretiza el contorno, de manera que pasa de ser una función continua a estar compuesto de un conjunto n de puntos de control que forman parte de un polígono. En segundo lugar se hacen constantes los coeficientes para las energías internas ($\alpha(s)$ y $\beta(s)$). Finalmente, se transforma la ecuación original de la energía a una formulación iterativa que permite el cálculo de las sucesivas posiciones de los puntos de control del snake en el tiempo $\{v_i(t)\}$, obteniendo finalmente un problema variacional.

La minimización de la energía del snake se puede tratar como un problema de optimización. Fundamentalmente se han utilizado tres técnicas de optimización: descenso del gradiente, programación dinámica y algoritmos voraces.

Para resolver numéricamente la expresión de la energía del snake ($E_{\text{snake}}(V(s))$) se discretiza sobre un conjunto de nodos $v_i = u(ih)$ para $i = 0, \dots, N - 1$ y $h = 1/(N-1)$, los cuales se hacen coincidir con los puntos de control del snake (snaxels). Además, se definen las derivadas en términos de diferencias finitas:

$$\frac{du}{ds} \approx \frac{v_{i+1} - v_i}{h}$$

$$\frac{d^2u}{ds^2} \approx \frac{v_{i+1} - 2v_i + v_{i-1}}{h^2}$$

El problema, definido en su versión discreta, consiste en encontrar la solución de la siguiente expresión:

$$\min \left\{ \sum_{i=0}^N (E_{int}(v_i) + E_{ext}(v_i)) \right\}$$

-Descenso del gradiente.

En el caso del método de descenso del gradiente [KASS88] se calcula una fuerza que desplaza los puntos de control del snake a partir de la condición necesaria de Euler-Lagrange para alcanzar un mínimo en 1:

$$-\frac{\partial(\alpha u)}{\partial s} + \frac{\partial(\beta u)}{\partial s^2} + \frac{\partial E_{ext}}{\partial s} = 0$$

La solución es iterativa y simple de implementar, pero no garantiza el óptimo global y presenta un gran coste computacional.

-Programación dinámica.

Otro método utilizado para la minimización es la programación dinámica [AMINI90] [COUGHLAN98], que se basa en reformular el problema de la minimización de la energía global como un problema de minimizaciones parciales:

$$E(v_1, v_2, \dots, v_N) = E_0(v_0, v_1, v_2) + E_1(v_1, v_2, v_3) + \dots + E_{N-2}(v_{N-2}, v_{N-1}, v_N)$$

donde

$$E_{i-2}(v_{i-2}, v_{i-1}, v_i) = E_{ext}(v_i) + E_{int}(v_{i-2}, v_{i-1}, v_i)$$

$$E_{i-1}(v_{i-1}, v_i, v_{i+1}) = E_{ext}(v_i) + E_{int}(v_{i-1}, v_i, v_{i+1})$$

Es decir, que se tiene en cuenta la energía debida al punto actual, al siguiente y al anterior, por eso se ponen los tres términos en la fórmula.

El método garantiza encontrar la solución óptima, pero es lento y requiere gran cantidad de memoria. Este método ajustará la posición de todos los snaxels en cada

iteración hasta que el snake se estabilice. Cada iteración se planteará como un problema de programación dinámica en el que se trata de encontrar el mínimo de energía global dentro de la vecindad de los puntos considerados.

-Método greedy.

Como alternativa más eficiente se ha definido un método greedy (voraz) William y Shah[WILLIAMS92], el cual es estable, rápido y flexible, aunque no asegura el óptimo global. Este algoritmo se basa en la suposición siguiente:

$$\min \left\{ \sum_{i=0}^N (E_{int}(v_i) + E_{ext}(v_i)) \right\} = \sum_{i=0}^N \min \{ E_{int}(v_i) + E_{ext}(v_i) \}$$

Al no ser cierta esta hipótesis, no se consigue el óptimo global, pero en la práctica se alcanza una solución aceptable, pues factores externos como el ruido, sombras, texturas, etc, hacen que la solución óptima del problema formal planteado no sea necesariamente la solución correcta en la aplicación real. Como ya se ha dicho anteriormente, este es el problema principal de los snakes, que son muy dependientes de la inicialización del contorno. A continuación se define la fórmula del algoritmo voraz.

El funcional de energía a minimizar es una particularización de la expresión de la energía total del snake ($E_{snake}(V(s))$):

$$E_{total} = \int (\alpha(s)E_{continuidad} + \beta(s)E_{curvatura} + \gamma(s)E_{imagen})ds$$

En el algoritmo original [WILLIAMS92] se calcula la función de energía en cada uno de los puntos en la vecindad de la posición actual del punto de control v_i . Cuanto mayor es la ventana definida como vecindad, mayor es la robustez del método ante mínimos locales, pero también es mayor la carga computacional. Los términos que componen la expresión son los siguientes:

- La energía de continuidad se define como:

$$E_{continuidad} = \bar{d} - |v_i - v_{-1}|$$

Favorece configuraciones con puntos equidistantes, siendo d la distancia media.

Esta función fuerza a los snaxels v_i a expandirse uniformemente. El valor d indica la distancia media entre snaxels vecinos. Los puntos de la imagen donde el snaxel quedaría separado más de la media de sus vecinos más próximos, atraen menos al snaxel. La energía viene dada por el valor absoluto de la distancia media entre snaxels menos la distancia entre snaxels vecinos.

- La curvatura se define como:

$$E_{curvatura} = |v_{i-1} - 2v_i + v_{i+1}|^2$$

o también

$$E_{curv} = \left\| \frac{\mu_{i+1}}{\|\mu_{i+1}\|} - \frac{\mu_i}{\|\mu_i\|} \right\|^2$$

$$u_i = v_i - v_{i-1}$$

Esta energía estima la curvatura en un punto. De esta manera mantenemos la restricción de forma (curvatura global).

Obsérvese que se utilizan los puntos de control anterior y posterior al considerado para calcular las restricciones de continuidad y suavidad.

- Y la energía de la imagen se define como:

Esta energía, de signo contrario a la anterior y también denominada energía externa, se define en función de las características hacia las que se pretende atraer el snake (contornos, puntos de intensidad elevada, transformada de distancia).

$$E_{imagen} = (min - mag)/(max-min)$$

donde min y max son el mínimo y máximo gradiente de la vecindad del punto v_i cuyo valor de gradiente es mag . La posición con el menor valor de energía es seleccionada como la nueva posición v_i . Al final de cada iteración se calcula la curvatura en cada punto de control y si el valor es mayor que un cierto umbral, β_i es puesto a cero para la siguiente iteración. El algoritmo termina cuando el número de puntos que se han movido en la última iteración no supera un cierto umbral.

2.3.2. Balloons

Laurent D. Cohen publicó en 1991 una nueva técnica de extracción de contornos que denominó Balloons [COHEN91] y en 1993 publicó, junto con Isaac Cohen [COHEN93], otro trabajo con algunas mejoras.

Para mejorar los resultados de los contornos activos existentes hasta ese momento, se creó una fuerza de inflado que hacía que el contorno activo se moviese de dentro hacia afuera, que se inflase como si fuese un globo.

Una de las mejoras consistió en definir la fuerza de forma continua, pues definirla de forma discreta provocaba oscilaciones entre un punto y el siguiente.

Respecto a la localización inicial del contorno activo, se tiene que inicializar manualmente. Esto tiene muchas consecuencias en la evolución de la curva [COHEN91]. Si la curva no está suficientemente cerca de una frontera, no será atraída por él. Si la curva no está sometida a ninguna fuerza, se encogerá hacia sí misma.

En las primeras versiones del modelo Balloon, si no había fuerza de imagen, el snake se encogía hacia sí mismo. Esto sucedía, por ejemplo, si la curva inicial se situaba en un área constante. Para evitar esto, Cohen et al. [COHEN93] introdujeron una presión interna que hizo que el contorno activo se comportase como un globo que se infla. La fuerza de presión se añade a las fuerzas internas y externas para empujar la curva hacia fuera, como si se estuviera introduciendo aire dentro. Así, la curva se expandía y era atraída hacia los bordes. Esta fuerza de presión interna también tiene un efecto de suavizado sobre el contorno activo.

Una desventaja que presenta este modelo, al igual que muchos otros, es que para que exhiba un correcto funcionamiento, es necesario optimizar los parámetros o coeficientes del modelo. Cohen [COHEN91] obtuvo buenos resultados poniendo los pesos de las fuerzas en función del paso de discretización del espacio.

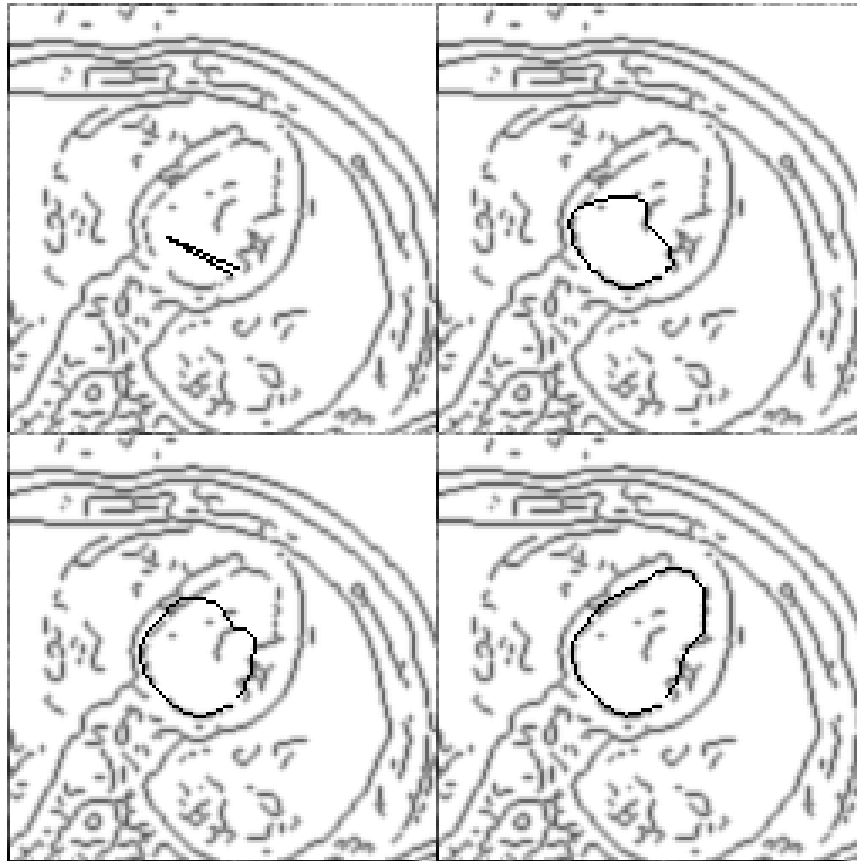


Figura 2.5. Evolución de la curva del balloon para segmentar el ventrículo izquierdo superpuesta a la imagen del potencial [COHEN91].

2.3.3. Snakes eléctricos

Los Snakes Eléctricos fueron propuestos por Marín-Hernández y Ríos-Figueroa en 1999 [MARIN99]. Propusieron asignar una carga eléctrica constante sobre todo el contorno para obtener una fuerza eléctrica entre cada par de puntos del contorno. Esta fuerza se describe como:

$$F = k \frac{q^2}{r^2}$$

siendo k una constante, q la carga eléctrica en cada punto y r la separación entre puntos.

La carga eléctrica se distribuye uniformemente sobre el snake inicial, para hacer que cada punto tenga la misma carga. Estas cargas eléctricas permanecen constantes todo el tiempo, por lo que su magnitud puede ser elegida libremente.

Algunos de los principios físicos que utilizaron Marín-Hernández y Ríos-Figueroa son [WEIDNER71]:

- a) Toda la carga de un conductor, permanecerá en la parte exterior de la superficie del mismo.
- b) El campo eléctrico del conductor son siempre líneas perpendiculares a la parte exterior de la superficie del mismo.
- c) La magnitud del campo eléctrico es exactamente cero dentro de cualquier conductor.
- d) Cuando el conductor haya alcanzado una situación de equilibrio, las cargas libres no se moverán a lo largo la superficie.

De estos principios se puede deducir que la densidad de carga tendrá una magnitud mayor en las regiones que tengan una curvatura mayor (Feynman et al., 1964 [FEYNMAN64]; Price y Crowley, 1985 [PRICE85]).

Así, basándose en las cargas eléctricas distribuidas a lo largo de todo el contorno activo, los snakes eléctricos añaden una nueva energía interna.

El efecto de la carga es el de separar los puntos de control del snake, a menos que lo impidan otras fuerzas de una magnitud mayor que la fuerza eléctrica (fuerzas ya presentes en el modelo clásico de snakes). Así, además de conseguir que haya más puntos de control en las zonas de más curvatura, también se consigue evitar que en el snake se produzcan “nudos” o bucles entre los puntos del contorno, ya que la fuerza eléctrica actúa como una fuerza de repulsión entre puntos de control del snake.

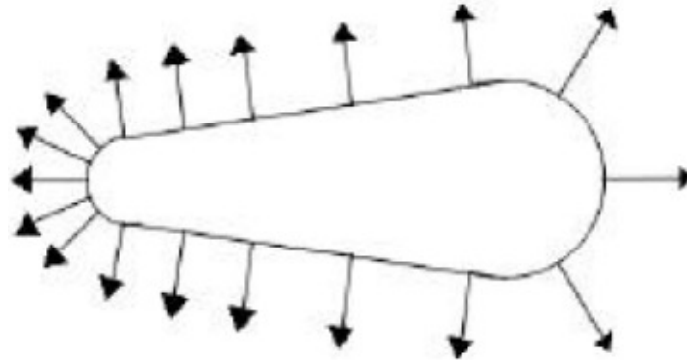


Figura 2.6. En un conductor cargado, la carga eléctrica es más fuerte en las zonas de mayor curvatura [WEIDNER71].

Entonces, cuando se alcance el equilibrio, los puntos de control formarán clusters en las regiones donde la curvatura sea alta y no se van a mover a lo largo de la superficie.

De esta forma, la expresión de la energía total (E_t) será:

$$E_t(v) = \int (w_1(s)|v_s|^2 + w_2(s)|v_{ss}|^2) ds + \int k \frac{q^2}{r^2} ds + \int P(v) ds$$

Usando la ecuación anterior en la minimización de la energía, los puntos de control definirán mejor el contorno, sin incrementar el número de puntos de control ni la complejidad temporal.

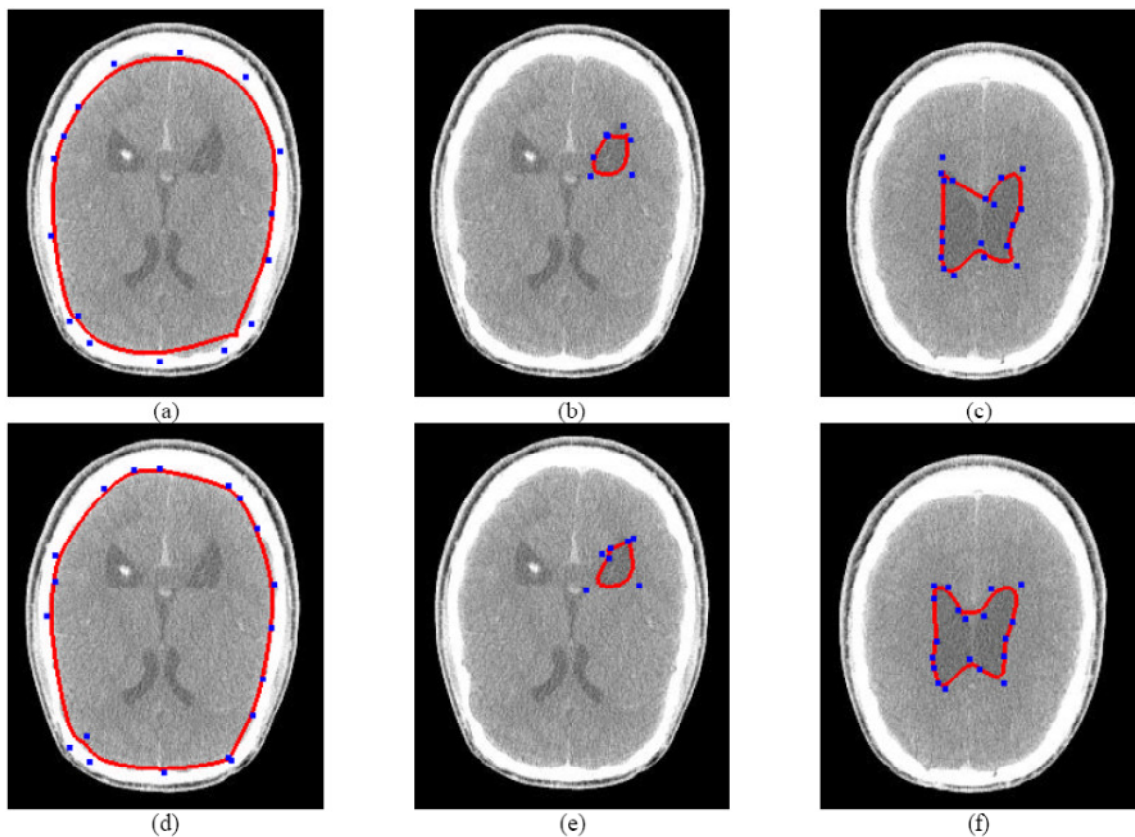


Figura 2.7. Figuras a, b y c muestran estructuras anatómicas de tomografías de una cabeza, identificadas mediante snakes estándar (snakes clásicos). Las mismas estructuras se detectan mejor con los snakes eléctricos en las figuras d, e y f. [MARIN99].

2.3.4. Contornos activos borrosos

En el año 2000 el grupo formado por Frank Höwing, Laurence S. Dooley y Diederich Wermser [HOWING00], investigó los contornos activos desde una nueva perspectiva, la lógica borrosa, naciendo así el modelo desarrollado a continuación.

El concepto de contorno activo borroso es una variación de los contornos activos tradicionales basada en conocimiento, es decir, se usa información de experiencias anteriores para manejar el comportamiento del snake y se introduce una base de reglas lingüísticas, las cuales describen los segmentos del snake borroso. Los

snakes “*borrosos*” permiten especificar de forma intuitiva las características de los bordes de un objeto mediante el uso de funciones de energía “*borrosa*” y una base de reglas lingüísticas que definen cada segmento del contorno.

La longitud de cada contorno se puede aproximar para:

- Mejorar el proceso de segmentación.
- Reducir la complejidad computacional.

Para definir la función de energía del snake y poder tener conocimiento almacenado, se utiliza una descripción gramatical en la que las energías externas se definen mediante terminaciones y las propiedades de los puntos del snake se definen mediante expresiones gramaticales.

En base a esto, se añade un algoritmo de reconocimiento de patrones en el proceso de minimización de energía que restringe los resultados de búsqueda a aquellos contornos que cumplan la expresión gramatical.

Desventajas principales:

- Diferentes funciones de energía no consideran la base de conocimientos como inexacta.
- Computacionalmente caro debido al gran número de posibilidades.
- No se puede especificar la longitud de un contorno exactamente, aunque se puede aproximar.

2.3.5. Contornos activos duales.

Steve R. Gunn y Mark S. Nixon [GUNN94] crearon los Contornos Activos Duales (dual active contour) sobre 1994, mientras trabajaban para el Departamento de Electrónica y Ciencias de la Computación de la Universidad de Southampton.

Esta técnica de contornos activos duales pretende mejorar el modelo tradicional para evitar sus principales problemas.

Utiliza un contorno interno y otro externo como inicializaciones del “snake”, que tienden a expandirse y a contraerse respectivamente para obtener una forma previa. Estos dos contornos son minimizados conforme al intercambio entre las restricciones de la forma previa (obtenida anteriormente) y la imagen.

Elimina parcialmente los problemas de inicialización porque aproxima las características deseadas por ambos lados, para ayudar a obtener un mínimo global. La energía interna se reformula para que sea invariante a la escala y para que rechace mínimos locales pobres.

Presenta un nuevo Modelo de Forma (similar a Lai y Chin [LAI94]) que permite un control local sobre el equilibrio del contorno. El problema de determinar los parámetros se simplifica reduciendo el número de parámetros a un sólo parámetro de ajuste λ con la consiguiente simplificación del modelo, aunque éste sigue siendo consistente con el paradigma original.

En la técnica original las restricciones de continuidad y curvatura forman una fuerza de contracción que hace que las soluciones sean en gran medida dependientes de los parámetros de esa fuerza interna (coeficiente de continuidad y curvatura).

Cuando los dos contornos se convierten en estacionarios se minimiza el de mayor energía hacia el otro de energía menor mediante una fuerza adicional. Esta fuerza se incrementa hasta que el contorno se empieza a mover y se elimina en el momento en que la energía disminuye, para permitir al contorno que vuelva a un estado de equilibrio. Este proceso se repite hasta que ambos contornos encuentran el mismo equilibrio. Esta fuerza adicional se denomina “Fuerza Conductora” y dota al snake con la capacidad de evitar los mínimos locales.

Cada uno de los contornos debe contener información sobre la forma local y debe ser invariante a las transformaciones geométricas (escalado, rotación y traslación) para que contornos de diferentes escalas se puedan comparar.

Un contorno llegará a un estado de equilibrio si es similar al contorno estimado, esto hace que no tiendan a expandirse o a contraerse por defecto, si no que busquen conseguir su forma natural.

La implementación tiene dos restricciones:

- Puntos uniformemente espaciados:

$$|v_{i-1} - v_i| = |v_{i+1} - v_i|$$

- El contorno debe tener una forma específica, factor de forma:

Por ejemplo, en el caso del círculo:

$$\varphi_{i=\frac{(N-2)}{N} \cdot \pi}$$

Estas condiciones son aplicadas por una fuerza que empuja a v_i hacia su posición estimada \tilde{v}_i , esta fuerza e_i se calcula a partir de los puntos del vecindario de v_i mediante:

$$e_i = \tilde{v}_i - v_i = \frac{1}{2}(v_{i-1} + v_{i+1}) - v_i + \theta_i \frac{1}{2}R(v_{i-1} - v_{i+1})$$

$\tilde{v}_i \rightarrow$ posición estimada.

$R \rightarrow$ matriz de rotación de 90° .

$$\theta_i = \tan(\pi/N)$$

Usando esta fuerza las energías interna y externa quedan de la siguiente forma:

$$E_{interna}(v_i) = E_{forma}(v_i) = \frac{1}{2} \left(\frac{|e_i|}{h} \right)^2$$

$$E_{externa}(v_i) = |\nabla I(x, y)|$$

La energía interna se normaliza mediante un valor h .

Esta función de la Energía externa se puede calcular mediante la aplicación de algún filtro, como por ejemplo un filtro laplaciano

Energía Total:

$$E_{snake}(v) = \frac{1}{N} \sum_{i=0}^{N-1} \lambda \cdot E_{interna}(v_i) + (1 - \lambda) \cdot E_{externa}(v_i) \quad \lambda \in [0, 1]$$

se usa en un Espacio Continuo (números reales).

El único parámetro a determinar es el parámetro de ajuste λ , el cual tiene un valor entre 0 y 1. Si λ es 1 el contorno está completamente regularizado y depende sólo de las fuerzas internas para su solución. Si λ es 0 entonces el contorno no está

regularizado y ninguna restricción será aplicada en la forma del contorno o su continuidad.

El proceso de minimización está implementado por el método del descenso del gradiente. Las fuerzas externas F se derivan de una función de imagen basada en el borde $|\nabla I(x,y)|$. Del mismo modo, la normalización de los valores de la energía externa debe quedar en el intervalo $[-1,1]$. Entonces el contorno evoluciona por:

$$v_i^{\tau+1} = v_i^{\tau} + \frac{1}{2} \left(\lambda \frac{e_i}{h} + (1 - \lambda) F_i \right)$$

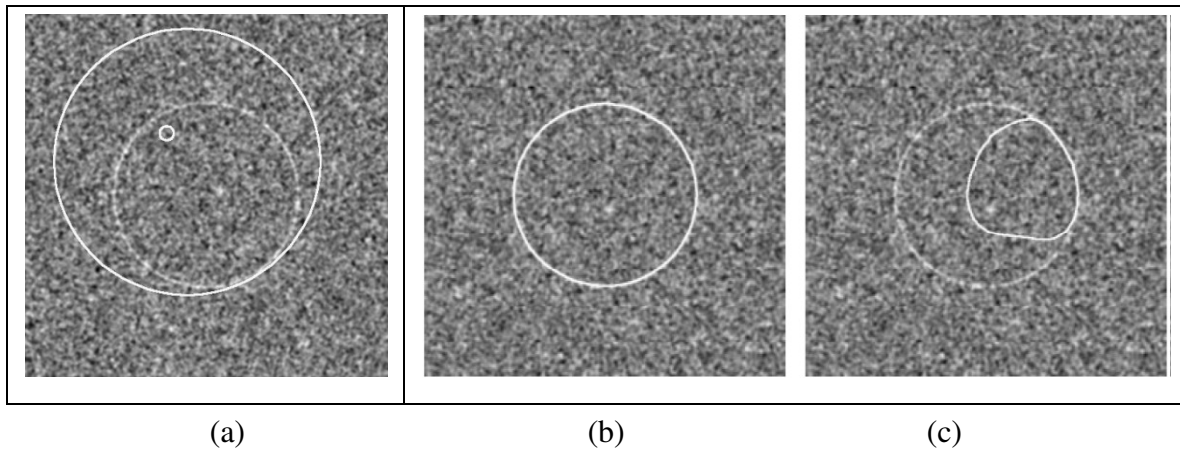


Figura 2.8. a) Ejemplo con snake dual, contornos iniciales; b) vemos que los snakes (en blanco) han formado el contorno del círculo; c) Ejemplo con el método Kass, vemos como el snake (en blanco) no ha definido todo el contorno buscado (en gris). [GUNN94]



(a) (b) (c)

Figura 2.9.a) Ejemplo con snake dual, contornos iniciales; b) con el snake dual se ha definido bien la zona a segmentar; c) Ejemplo con el método Kass.

2.3.6. Snakes topológicamente adaptables.

Este modelo de contornos activos, creado por Tim McInerney y Demetri Terzopoulos, fue presentado en 1995 [MCINERNEY95] en la 5ª Conferencia internacional sobre visión por ordenador.

Definen su Topologically Adaptable Snake (T-snake) como contorno cerrado en 2D consistente en una serie de nodos conectados en serie. Un T-Snake es una aproximación discreta a un modelo convencional de snake paramétrico y conserva muchas de sus propiedades. Las fuerzas internas actúan como una restricción de suavidad y los usuarios pueden interactuar con el modelo usando resortes y otras restricciones. Una fuerza de inflado se usa para empujar el modelo hacia los bordes de la imagen hasta que sea frenada por las fuerzas externas. La deformación del modelo está gobernada por ecuaciones del movimiento discretas de Lagrange.

A diferencia de los modelos tradicionales, el conjunto de nodos y los elementos de interconexión de un T-snake no permanecen constantes durante la evolución. Descomponemos el dominio de la imagen en una rejilla de celdas discretas. A medida que el modelo se mueve bajo la influencia de fuerzas internas y externas, se reparametriza el modelo con un nuevo conjunto de nodos y elementos calculando los puntos de intersección del modelo con esta rejilla ACID (Affine Cell Image Decomposition). Esto permitirá al modelo ser relativamente insensible a la posición inicial y fluir hacia formas complejas con topologías complejas de una manera estable.

El movimiento en general de un T-snake es análogo al movimiento de un frente de propagación. Sin embargo, hay dos fases distintas para moverse. En la primera fase, entre fases de reparametrización, un T-snake se comporta como un snake paramétrico estándar y evoluciona de acuerdo a ecuaciones de Lagrange. Esta fase de

formulación Lagrangiana permite que cualquier fuerza derivada de datos o definida por el usuario guíe al snake. Durante la segunda fase o fase de reparametrización, el snake se reparametriza en términos de la rejilla ACID y los puntos de rejilla fijados se usan para rastrear el interior del modelo de contorno cerrado, creando un particionamiento de espacio similar al de una función implícita. De esta forma, el contorno puede reconocer formas complejas y cambiar topológicamente de manera dinámica, dividiéndose o uniéndose según las necesidades. Si se desea convertir el modelo para usarlo como un “snake” tradicional sólo es necesario eliminar la rejilla.

El comportamiento de un T-snake está gobernado por una ecuación del movimiento que es una simplificación de la de Lagrange:

$$\mu \frac{\partial^2 v}{t^2} + \gamma \frac{\partial v}{t} - \frac{\partial}{s} \left(w_1 \frac{\partial v}{s} \right) + \frac{\partial^2}{s^2} \left(w_2 \frac{\partial^2 v}{s^2} \right) = -\nabla P(v(s, t))$$

y trabaja con:

$$\gamma_i \dot{x}_i + a \alpha_i + b \beta_i = \rho_i + f_i$$

donde $\dot{x}_i(t)$ es la velocidad del nodo i y γ es un coeficiente de enfriamiento.

Las fuerzas de tensión interna

$$\alpha_i(t) = 2x_i(t) - x_{i-1}(t) - x_{i+1}(t)$$

son una aproximación discreta de las derivadas con respecto a s de la ecuación de Lagrange y actúan para mantener un espaciado uniforme entre nodos del modelo. El parámetro a controla la resistencia del contorno a deformaciones por estiramiento.

Las fuerzas internas de flexión

$$\beta_i(t) = 2\alpha_i(t) - \alpha_{i-1}(t) - \alpha_{i+1}(t)$$

son una aproximación discreta a la cuarta derivada con respecto a s . El parámetro b controla la resistencia del contorno a deformaciones por curvamiento.

La fuerza de inflado es:

$$p_i(t) = q F(I(x_i(t))) n_i(t)$$

donde n_i es el vector unitario normal al contorno en el nodo i y q es la amplitud de esta fuerza.

La función binaria

$$F(I(x, y)) = \begin{cases} +1 & \text{Si } I(x, y) \geq T \\ -1 & \text{En otro caso} \end{cases}$$

une la fuerza de inflado a los datos de la imagen $I(x, y)$, donde T es un umbral de intensidad de la imagen. La función F hace contraer al T -snake cuando $I(x, y) < T$ y se usa para prevenir al T -snake de filtrarse hacia el fondo.

Para parar el contorno en bordes importantes incluimos la fuerza externa:

$$f_i(t) = p \nabla P(x_i(t))$$

donde el peso p controla el valor de la fuerza y P es el potencial

$$P(x, y) = -c \|\nabla [G_\sigma * I(x, y)]\|$$

p debe ser ligeramente mayor que q para que el contorno se pare en un borde importante pero con q lo suficientemente grande como para atravesar los bordes débiles o espurios.

Para actualizar la posición de los nodos del modelo desde el tiempo t al $t + \Delta t$ usamos la siguiente fórmula:

$$x_i^{(t+\Delta t)} = x_i^t - \frac{\Delta t}{\gamma} (a\alpha_i^{(t)} + b\beta_i^{(t)} - \rho_i^{(t)} - f_i^{(t)})$$

-Fase 1: en la fase 1 del algoritmo de reparametrización, llevamos a cabo una búsqueda local eficiente y un test de intersección de cada elemento del t -snake con los bordes de celda de la rejilla. Si encontramos un punto de intersección para un borde de celda de la rejilla, se almacena en una estructura de datos asociada con este borde. Este punto de intersección puede llegar a ser un nodo del t -snake actualizado (fig. 29b,c). el punto de intersección puede no ser usado y por tanto descartado si después de la segunda fase del proceso ambos vértices de rejilla del borde de celda de la rejilla están “on”. Esto significa que ambos vértices de la rejilla deben estar dentro del t -snake y el borde que los une no es por tanto un borde de frontera.

Se pueden encontrar varios puntos de intersección para un borde de celda de la rejilla particular. Esta situación ocurre cuando, durante un paso de deformación, un

t-snake interseca a sí mismo o cuando múltiples t-snake intersecan. En estos casos, tomamos el vértice con numeración más baja de la rejilla y determinamos en que cara de la línea formada por el elemento t-snake yace (fig. 26).

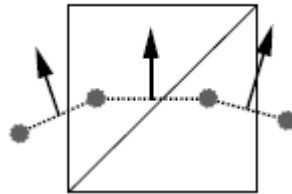


Figura 2.10. Ejemplo 2D que ilustra el cálculo del signo de un borde de celda de la rejilla y un punto de intersección. Se usa el elemento t-snake normal para determinar qué vértice del borde de la rejilla está en la mitad interior del elemento t-snake y cuál está en la mitad exterior. [MCINERNEY00].

Consecuentemente, a cada punto de intersección se le da un signo, o dentro o fuera (usando el elemento normal apuntador como dirección de referencia). Así, cada punto de intersección se compara con el punto de intersección existente (si hubiese) de un borde de celda de la rejilla. Si los puntos son de signo opuesto, se cancelan mutuamente. Si son del mismo signo, el nuevo punto de intersección reemplaza al punto existente. Si no existe punto de intersección para el borde de la rejilla, el nuevo punto de intersección se almacena en la estructura de datos de bordes de rejilla.

Finalmente, si el borde de rejilla de la mitad interior del elemento t-snake está “off”, también lo almacenamos en una cola para procesarlo en la fase 2.

-Fase 2: durante un paso de deformación, a medida que el t-snake se expande, algunos vértices de rejilla que estaban fuera del t-snake cerrado se meterán dentro ahora. Entonces actualizamos el estado de los vértices de la rejilla de “off” a “on”. De esta manera, somos capaces de determinar y seguir la región interior del t-snake. El conjunto total de “on” o vértices de rejilla interiores definen sin ambigüedad la frontera del t-snake; se usan para seguir las celdas de la frontera a lo largo de la evolución del t-snake y por tanto determina qué puntos de intersección deberían ser usados para formar los nodos y elementos del nuevo t-snake.

Como se mencionó en la fase 1, durante un paso de deformación, para cada elemento del t-snake se calculan sus puntos de intersección del borde de celda y metemos en cola los vértices para ser etiquetados “on” o dentro para el proceso de la fase 2. En la fase 2, desencolamos estos vértices y comprobamos sus estructuras de datos correspondientes. Si hay almacenado un punto de intersección en una estructura de datos de borde de un vértice de rejilla (indicando que el t-snake se ha movido de tal forma que el vértice de rejilla está ahora dentro) y el vértice de rejilla está “off”, lo ponemos a “on”. Para todos los vértices de rejilla que se pusieron a “on”, los usamos como vértices semillas en un algoritmo de relleno de región estándar para poner a “on” cualesquiera vértices vecinos que estén en el interior del t-snake y estén todavía a “off”.

Algoritmo T-snake [MCINERNEY00].

Para cada paso de deformación (M medidas de tiempo)

1. Para M medidas de tiempo:
 - a) Calcular las fuerzas internas y externas que actúan sobre el T-snake y sus elementos.
 - b) Actualizar las posiciones de los nodos usando la fórmula anterior.
2. Llevar a cabo la primera fase de reparametrización.
3. Llevar a cabo la segunda fase de reparametrización.
4. Para todos los elementos actuales del T-snake, determinar si el elemento es todavía válido. Un elemento de T-snake es válido si su correspondiente rejilla es todavía una celda frontera. Descartar elementos del T-snake inválidos y nodos no usados.
5. Usar los vértices de la rejilla activados en la fase 2 (si hay alguno) para determinar las nuevas células frontera y por tanto los nuevos nodos y elementos del T-snake.

Descomposición simple.

El modelo utilizado para partir el dominio de trabajo es la triangulación más simple del espacio Euclídeo, llamada triangulación “Coxeter-Freudenthal” [COXETER34] [FREUDENTHAL42]. Este es un modelo que evita la ambigüedad, dividiendo el espacio en rejillas cúbicas uniformes, que se dividen en $n!$ figuras simples, obteniendo la triangulación requerida.

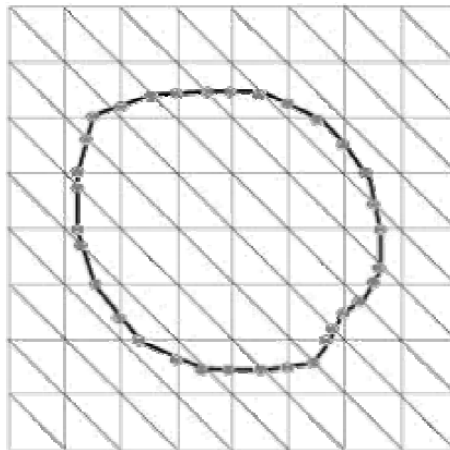


Figura 2.11. Aproximación simple de un modelo de contorno usando triangulación Freudenthal [MCINERNEY95].

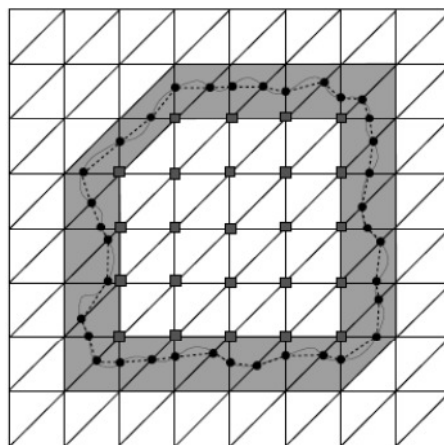


Figura 2.12. Aproximación simplificada (línea punteada) del contorno de un objeto (línea sólida) usando triangulación Freudenthal. Los nodos modelo (puntos de intersección) están marcados y los triángulos frontera están sombreados [MCINERNEY00].

Según la bibliografía, los vértices pertenecientes al contorno toman valores positivos y los pertenecientes al fondo valores negativos.

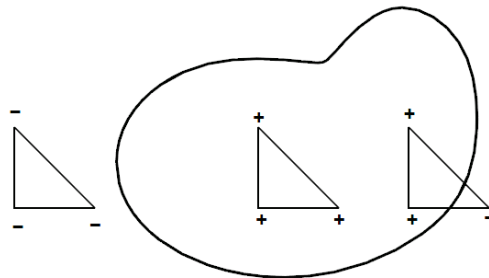


Figura 2.13. Clasificación de las celdas de la rejilla según si están localizadas fuera, dentro o sobre el contorno[MCINERNEY00].

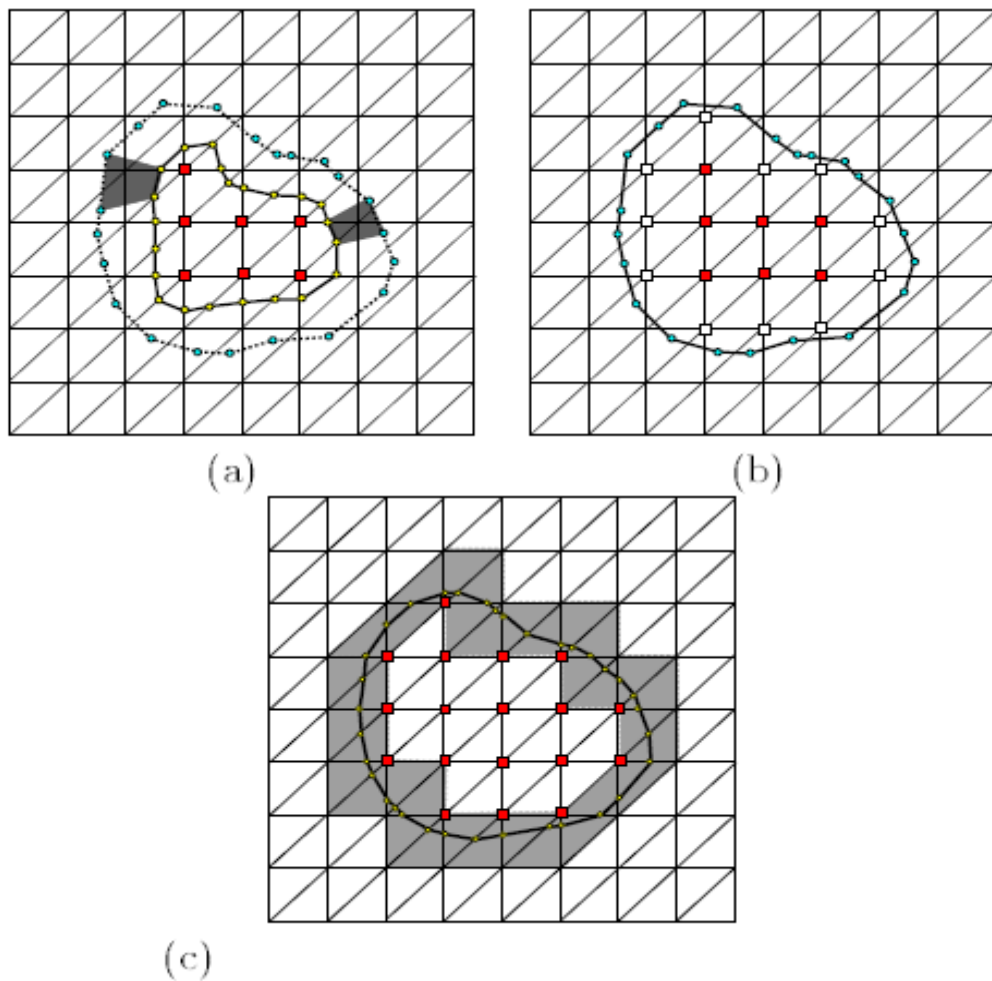


Figura 2.14. Fase dos de la reparametrización del T-snake. a) Las regiones sombreadas muestran ejemplos de vértices de rejilla que se activan debido al contorno en expansión, b) nuevos nodos de rejilla interiores (blancos) añadidos a los actuales nodos interiores (rojos), c) nuevo contorno después de un paso de deformación mostrando las nuevas intersecciones de rejilla, vértices de rejilla interiores y celdas de rejilla frontera (gris sombreado)[MCINERNEY95].

Transformaciones topológicas.

Las transformaciones topológicas consisten en la conexión o desconexión entre los nodos del “snake”.

Las decisiones para conectar o desconectar dos nodos se basan en la información proporcionada por la rejilla superpuesta, que como se explicó anteriormente ofrece un marco de trabajo sin ambigüedad. Una colisión entre contornos se manifiesta con un triángulo que contiene más de una línea como se puede ver en la figura 35, el triángulo A ofrece una colisión puesto que por el pasan un trozo de cada contorno. Para realizar la transformación topológica se unen los dos extremos finales de cada una de las líneas del triángulo en cuestión, dichos extremos finales se obtienen utilizando la información proporcionada por los signos de los vértices del triángulo.

Todo esto queda expuesto cuidadosamente en la figura 35.

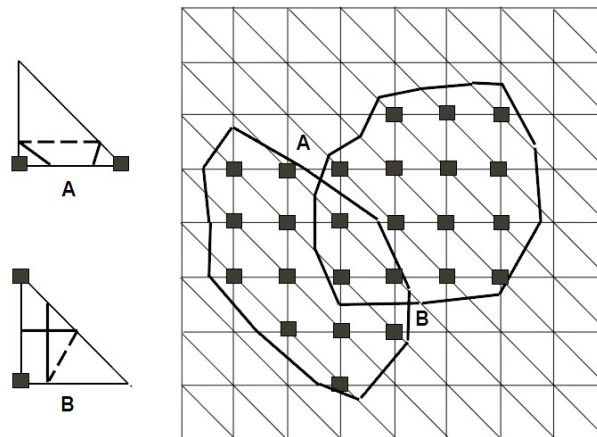


Figura 2.15. Las casillas marcadas de la rejilla representan puntos interiores del contorno. Las líneas discontinuas de los triángulos A y B son las nuevas conexiones de los nodos del snake en esta situación[MCINERNEY95].

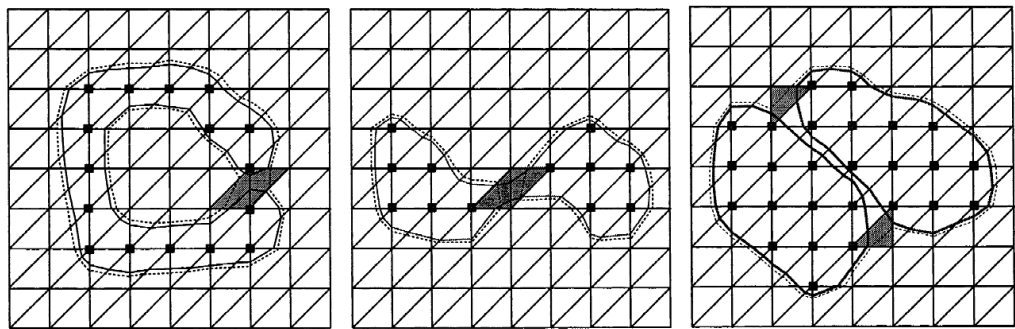


Figura 2.16. Ejemplos de transformaciones topológicas T-snake. Izquierda: intersección consigo mismo. Centro: encogimiento y separación. Derecha: mezcla. Los T-snakes resultantes después de las transformaciones se muestran como líneas punteadas. Las reconexiones de nodos ocurren en los triángulos sombreados automáticamente[MCINERNEY00].

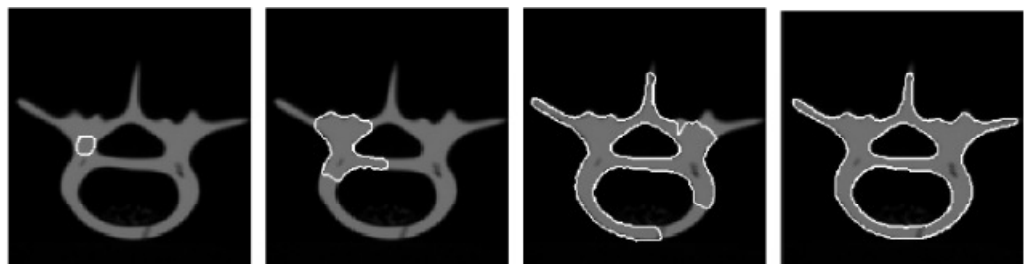


Figura 2.17. Ejemplo de utilización sobre una imagen real de un corte axial de una vértebra[MCINERNEY95].

2.3.7. Superficies deformables adaptables (T-surfaces).

Esta nueva técnica desarrollada por Tim Macinerney y Demetri Terzopoulos [MCINERNEY99] utiliza superficies deformables en términos de una descomposición afín de celdas (ACID). En particular, usando ACID se consigue un mecanismo de reparametrización eficiente que habilita a las superficies paramétricas deformables para evolucionar como geometrías complejas, incluso modificando su topología si fuese necesario.

Como se ha visto en el apartado anterior, los autores introdujeron un nuevo modelo de contorno deformable llamado snake topológicamente adaptable (T-snake) [MCINERNEY95]. Los T-snake explotan la descomposición afín de celdas, un marco de trabajo que extiende significativamente las habilidades de los snakes paramétricos estándar. El T-snake puede adaptar dinámicamente su topología a la del objeto objetivo, fluir alrededor de objetos y/o mezclarse automáticamente con otros modelos introducidos interactivamente por el usuario. Así, los snakes paramétricos inmersos en un ACID son capaces de segmentar y reconstruir incluso las estructuras biológicas con forma más compleja con un alto grado de eficiencia.

En este trabajo se presenta una extensión natural del marco de trabajo ACID. En particular, desarrollan los T-surfaces para usarlos en imágenes con volumen como si fuese una ampliación de los T-snakes para tres dimensiones.

2.3.7.1 Formulación de las T-surfaces.

Los componentes principales de la formulación tridimensional T-surfaces son análogos a los usados en los T-snakes bidimensionales.

Una T-surface es una malla cerrada, elástica y triangular que consiste en un conjunto de nodos y elementos triangulares. Asociamos con cada nodo i una posición

variante con el tiempo $\mathbf{x}_i(t) = [x_i(t), y_i(t), z_i(t)]$ junto con fuerzas internas (flexión) $\alpha_i(t)$ y fuerzas de rigidez $\beta_i(t)$, fuerzas de inflado $p_i(t)$ y fuerzas externas $f_i(t)$. El comportamiento de una T-surface está gobernado por una versión simplificada de las ecuaciones de Lagrange del movimiento:

$$\gamma_i \dot{x}_i - a\alpha_i + b\beta_i = p_i + f_i$$

Donde \dot{x}_i es la velocidad del nodo i y γ_i es un coeficiente que controla la tasa de disipación de la energía cinética de los nodos, p_i y f_i son fuerzas externas. Puesto que el modelo no tiene inercia, llega a un estado de descanso (por ejemplo $\dot{x}_i = 0$) tan pronto como las fuerzas aplicadas igualan a las fuerzas internas. Se usa una fuerza de inflado para empujar la T-surface hacia los bordes de intensidad de la imagen $I(x, y, z)$, hasta que las fuerzas de la imagen se oponen. La fuerza de inflado es:

$$p_i(t) = qF(I(x_i(t)))n_i(t)$$

Donde n_i es el vector unitario normal al modelo en el nodo i , y q es la amplitud de esta fuerza. La función binaria

$$F(I(x)) = \begin{cases} +1 & \text{si } I(x) \geq T \\ -1 & \text{en otro caso} \end{cases}$$

une la fuerza de inflado a los datos de la imagen $I(x)$, donde T es un umbral de intensidad de imagen.

Aproximamos la fuerza interna por la siguiente función:

$$\alpha_i(t) = L(x_i(t)) = \frac{1}{m} \sum_{j \in N(i)} x_j(t) - x_i(t)$$

$$\beta_i(t) = \alpha_i(t) - \alpha_{i-1}(t) - \alpha_{i+1}(t)$$

Donde x_j son los vecinos del nodo x_i , y $m = \#N(i)$ es el número de estos vecinos. El parámetro a se usa para controlar la intensidad de esta fuerza.

Para parar el modelo cuando llegue a bordes significantes, incluimos la fuerza externa:

$$f_i(t) = p \nabla P(x_i(t))$$

Donde el peso p controla la intensidad de la fuerza y el potencial P se define como:

$$P(x) = -c \|\nabla [G_\sigma * I(x)]\|$$

$G_\sigma * I$ representa un filtro Gaussiano de suavizado de desviación estándar σ y c escalas de potencial. Los pesos p y q normalmente se eligen para que sean del mismo orden, con p un poco más grande que q para que un borde significativo pare la inflamación pero con q lo suficientemente grande como para que el modelo pase a través de bordes débiles o espurios.

Se actualizan las posiciones de los nodos del modelo desde el tiempo t al tiempo $t + \Delta t$ de acuerdo a la siguiente fórmula:

$$x_i^{(t+\Delta t)} = x_i^{(t)} - (\Delta t/\gamma)(-a\alpha_i^{(t)} + b\beta_i^{(t)} - p_i^{(t)} - f_i^{(t)})$$

2.3.7.2. Descomposición de la imagen por celdas afines en tres dimensiones.

El segundo componente de las T-surfaces es la extensión del marco de trabajo ACID (Affine Cell Image Decomposition) para tres dimensiones usando celdas tetraédricas o hexaédricas. Para implementar las T-surfaces se ha usado la aproximación tetraédrica. En este esquema el espacio se particiona en celdas tetraédricas usando la triangulación Coxeter-Freudenthal [COXETER34] [FREUDENTHAL42]. Construimos la rejilla dividiendo el volumen de la imagen con una rejilla cúbica uniforme y subdividiendo cada cubo en 6 tetraedros (figura 38(a)). La aproximación poligonal se construye a partir de la intersección de la superficie del objeto con los bordes de cada celda frontera. Los puntos de la intersección forman un triángulo o un cuadrilátero (que puede ser dividido en dos triángulos) (figura 38(b)(c)). El conjunto de todos estos triángulos constituye la aproximación poligonal de la superficie del objeto.

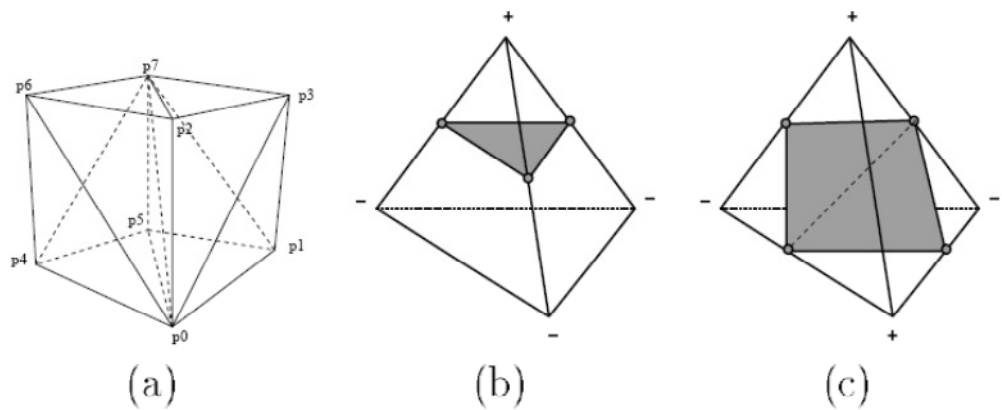


Figura 2.18. (a) Cubo dividido en 6 tetraedros; (b) Intersección de objetos de frontera con celdas de rejilla.

La versión en tres dimensiones es exactamente análoga al caso de dos dimensiones excepto en que estamos calculando intersecciones entre elementos triangulares y bordes de celdas de rejilla en R^3 en vez de elementos de borde y bordes de celdas de rejilla en R^2 .

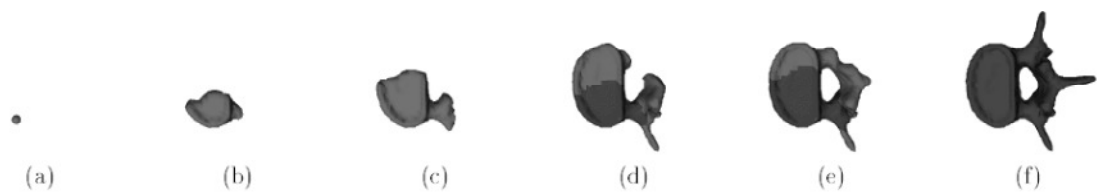


Figura 2.19. Ejemplo de progreso en una segmentación T-surface.

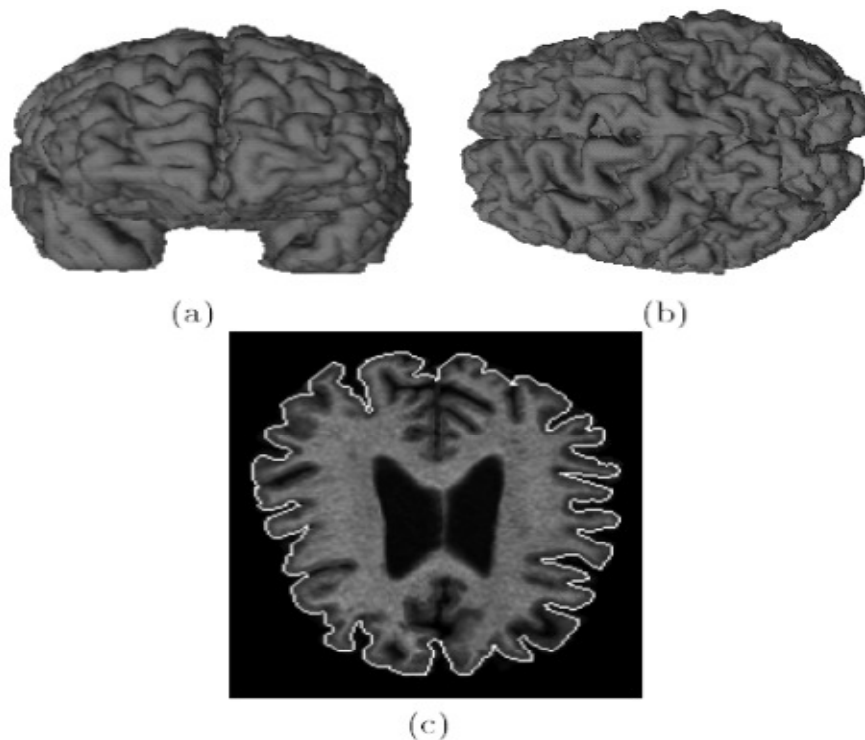


Figura 2.20. Segmentación T-surface de la corteza cerebral. a) Vista frontal; b) Vista superior; c) Sección cruzada de la T-surface sobre un corte.

2.3.8. Snakes estadísticos

Ivins y Porrill [IVINS94a] presentaron los “snakes” estadísticos basados en distintas fuerzas de presión. Este trabajo se llevó a cabo en colaboración con el Centro Científico Británico de IBM como parte del proyecto AIM A2003: Visión por Ordenador en la Radiología (COVIRA, COmputer VIsion in RAdiology), mientras Ivins y Porrill pertenecían a la Unidad de Investigación sobre Visión en Inteligencia Artificial de la Universidad de Sheffield en Inglaterra.

Es una técnica de crecimiento de regiones que usa snakes cerrados conducidos por una fuerza de presión que está en función de las características estadísticas de los datos de la imagen, estas características suelen ser normalmente la media y la desviación estándar.

El “snake” estadístico se expande hasta que los puntos del contorno sobrepasan los límites definidos por el usuario, relativos a la región sembrada. En ese momento la fuerza de presión se invierte para hacer contraer el modelo. Las fuerzas de tensión y rigidez mantienen el borde de la región liso y una fuerza de repulsión previene la auto intersección.

La fuerza de presión no siempre permite al contorno alcanzar el equilibrio, ya que a veces la imagen no tiene los bordes suficientemente definidos. En esos casos los “snakes” estadísticos pueden modelar las características de la imagen que no puedan extraerse por los “snakes” tradicionales.

Un “snake” estadístico es un modelo de regiones activas conducido por un término de energía que une la fuerza de presión a los datos de la imagen.

Un modelo basado en una fuerza de presión clásica utilizaría la siguiente función de energía:

$$E = \underbrace{\frac{\alpha}{2} \oint \left| \frac{\partial u}{\partial \lambda} \right|^2 \cdot d\lambda}_{\text{Tensión}} + \underbrace{\frac{\beta}{2} \cdot \oint \left| \frac{\partial^2 u}{\partial \lambda^2} \right|^2 \cdot d\lambda}_{\text{Rigidez}} - \underbrace{\frac{\rho}{2} \cdot \oint \frac{\partial u}{\partial \lambda} x u \cdot d\lambda}_{\text{Presión}} + \underbrace{\oint P(I(U)) \cdot d\lambda}_{\text{Potencial}}$$

El potencial P se genera normalmente procesando la imagen I para hallar los bordes destacados. De acuerdo con la ecuación del movimiento, la fuerza de presión f surgiendo de la energía de la imagen es simplemente la normal a la frontera del snake ponderada por la función de bondad G.

Siendo la definición del contorno:

$$u(\lambda) = (x(\lambda), y(\lambda))$$

El modelo que nos atañe en este momento utiliza una función muy similar que integra la Energía Potencial en la Fuerza de Presión:

$$\frac{\partial u}{\partial t} = \underbrace{\alpha \cdot \frac{\partial^2 u}{\partial \lambda^2}}_{\text{Tensión}} - \underbrace{\beta \cdot \frac{\partial^4 u}{\partial \lambda^4}}_{\text{Rigidez}} + \underbrace{\frac{\rho}{2} \cdot G(I(u)) \cdot \left(\frac{\partial u}{\partial \lambda} \right)^\perp}_{\text{Presión}}$$

Todos los pixels se consideran lo bastante buenos para ser incluidos en el modelo, y la frontera se expande hasta que o es atrapada por bordes fuertes o las fuerzas de suavizado se convierten en prohibitivas.

Siendo α , β y ρ constantes, se pueden usar varios tipos de función de presión:

• **Presión binaria:** que une la fuerza de presión con los datos de la imagen, de forma que cuando el límite estadístico definido por el usuario es sobrepasado se hace contraer el modelo, este límite viene en función de las estadísticas obtenidas de la región semilla tales como media y desviación estándar.

$$G(I(x, y)) = +1 \quad (|I(x, y) - \mu| \leq k \cdot \sigma) \quad f(u) = +\frac{\rho}{2} \cdot \left(\frac{\partial u}{\partial \lambda}\right)^{\perp}$$

$$G(I(x, y)) = -1 \quad (|I(x, y) - \mu| > k \cdot \sigma) \quad f(u) = -\frac{\rho}{2} \cdot \left(\frac{\partial u}{\partial \lambda}\right)^{\perp}$$

La intensidad μ y la desviación estándar σ se calculan de una región sembrada, k es una constante definida por el usuario (normalmente 2 ó 3).

• **Presión lineal:** la dirección de la fuerza de presión está continuamente cambiando a medida que los elementos de la frontera se mueven entre los píxels a cada lado del umbral estadístico para la inclusión en el modelo. La estabilidad puede ser mejorada además usando un funcional de bondad normalizada:

$$G(I(x, y)) = 1 - \frac{|I(x, y) - \mu|}{k\sigma} \quad f(u) = \rho \left(\frac{\partial u}{\partial \lambda}\right)^{\perp} \left(1 - \frac{|I(u) - \mu|}{k\sigma}\right)$$

Si $I(u) - \mu = 0 \rightarrow$ Presión Lineal Máxima \rightarrow Expansión Rápida.

Si $I(u) - \mu \gg k \cdot \sigma \rightarrow$ Presión Lineal Muy Negativa \rightarrow Contracción Rápida.

Si $I(u) - \mu = k \cdot \sigma \rightarrow$ Presión Lineal Nula \rightarrow Estabilidad.

$\mu \rightarrow$ Intensidad media.

$\sigma \rightarrow$ Desviación estándar.

$k \rightarrow$ Constante definida por el usuario normalmente 2 ó 3.

• **Presión Mahalanobis:** que extiende los términos de presión binaria y lineal para tratar características multidimensionales, es decir, podría considerar

información de varias rodajas bidimensionales para calcular la presión que se debe aplicar en cada punto. Si tenemos el vector multidimensional $I(u)$ en vez del escalar $I(u)$

$$G(I(x, y)) = 1 - \frac{1}{k} \sqrt{x^2} \quad f(u) = \rho \left(\frac{\partial u}{\partial \lambda} \right)^\perp \left(1 - \frac{1}{k} \sqrt{x^2} \right)$$

$$x^2 = (I(u) - \mu_I)^T S_I^{-1} (I(u) - \mu_I)$$

El límite k se elige de las tablas χ^2 estándar de acuerdo al número de imágenes. El vector de la media μ_I y la matriz de covarianza S_I se calculan según una región de siembra en cada imagen S .

$$s = \frac{1}{M-1} \sum_S (I(x, y) - \mu_I)(I(x, y) - \mu_I)^T$$

Usando la presión Mahalanobis un snake estadístico puede combinar información de múltiples imágenes. Por ejemplo para desarrollar un snake sensible al color que se ejecute a la vez en imágenes en rojo, verde y azul.

- (a) Región de siembra (b) Expansión (c) Equilibrio

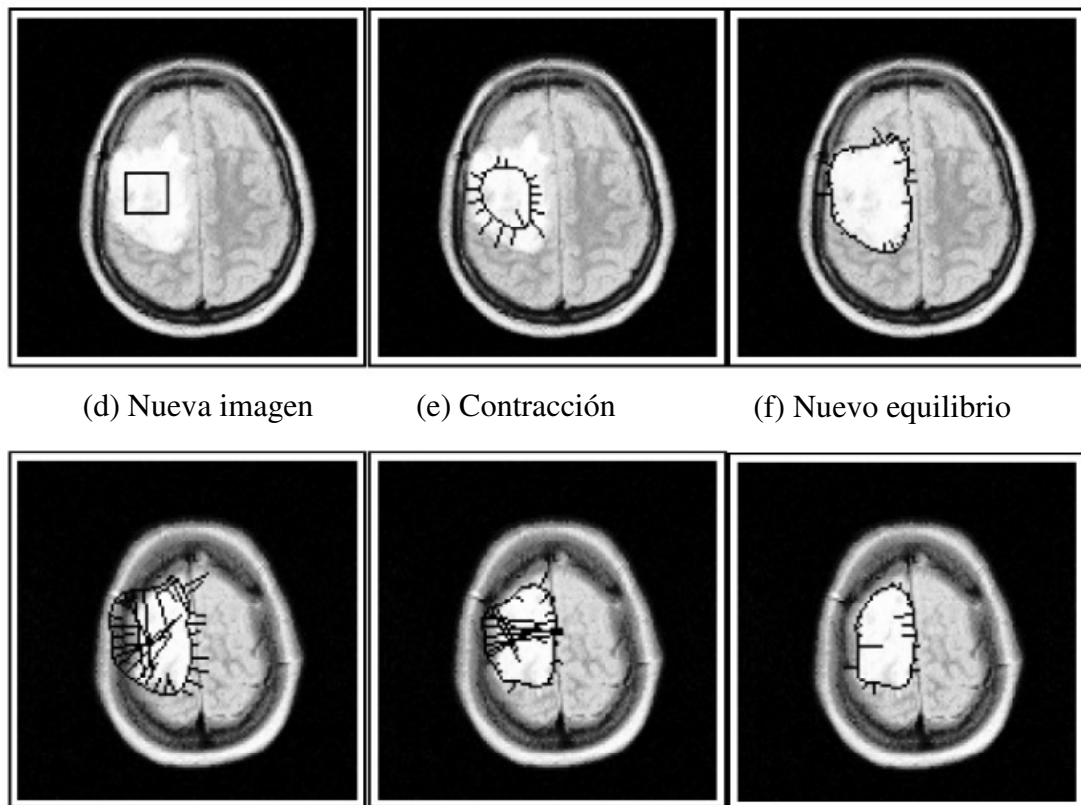


Figura 2.21. Muestra dos cortes diferentes de datos NMR. a) el usuario define una región de siembra (dentro del área pálida de tejido inflamado) de la cual los pixels son automáticamente analizados para su media y desviación estándar; b) la frontera de la región sembrada se convierte a un snake estadístico que se expande hasta que sus elementos encuentran pixels que caen fuera de los límites predefinidos relativos a las estadísticas de siembra; c) en equilibrio el modelo oscila ligeramente, como se muestra con las fuerzas de presión lineales normales a la frontera; d) el modelo se copia a otra imagen donde se expone a nuevas fuerzas de presión; e) el modelo se deforma para ajustarse a los nuevos datos –secciones de frontera que caen fuera de la región pálida; f) en 50 iteraciones el modelo se ajusta a los nuevos datos ($\alpha = \beta = p = 1$; $k=2$).

2.3.9. Modelo de contorno activo adaptable a la textura

En el año 2001 Thomas Lehmann, Jörg Bredno y Klaus Splitzer [LEHMANN01], trabajando para el Instituto de Informática aplicada a la Medicina de la Universidad Tecnológica de Aachen, crearon los modelos de contornos activos adaptables a la textura.

Este método combina el modelo de los “balloons” con un análisis a priori de la textura que determina la similitud entre la imagen a procesar e imágenes anteriores de entrenamiento. Este análisis permite adaptar la parametrización del procedimiento de segmentación. El análisis global de texturas permite configurar automáticamente los parámetros para imágenes heterogéneas sin la intervención de ningún experto que determine el ajuste de dichos parámetros de segmentación. Para ello es necesario el uso de diferentes conjuntos de parámetros asociados a distintos grupos de texturas adaptables según las imágenes.

-Método del balloon para la segmentación.

Para la segmentación de imágenes médicas, se aplica un modelo generalizado de balloon que se basa en mallas de elementos finitos. Los elementos del borde de las mallas se mueven bajo influencias mecánicas hasta que contactan con bordes significativos de objetos de la imagen. Los bordes de un polígono se mueven iterativamente por fuerzas que resultan de la presión del balloon, una fuerza deformable que reduce la derivada de segundo orden, e influencias de la imagen. Los niveles de gris o de color son interpretados como potenciales de imagen, resultado de fuerzas externas basadas en regiones y ofrecen una resistencia local al movimiento del contorno. Basándose en este modelo, el algoritmo localiza las distintas estructuras formadas sin necesidad de definir un contorno inicial. Este modelo al igual que todos los basados en “balloons” requiere una cuidadosa parametrización, en concreto se deben ajustar las siguientes consideraciones sobre los elementos finitos:

- Las distancias máxima y mínima entre los puntos de control del contorno.

- La escala de gradientes de los bordes de la imagen o de los objetos relevantes.
- La intensidad de los gradientes que se deben codificar como valores de potencial de la imagen.
- El valor de la fuerza de deformación.
- El valor de la fuerza de presión.

Los parámetros de segmentación se pueden agrupar para determinados tipos de imágenes de entrada. Si se desea una segmentación totalmente automática para imágenes de distinto tipo, es necesaria una fase de entrenamiento del sistema para adecuar los parámetros a cada tipo de imagen.

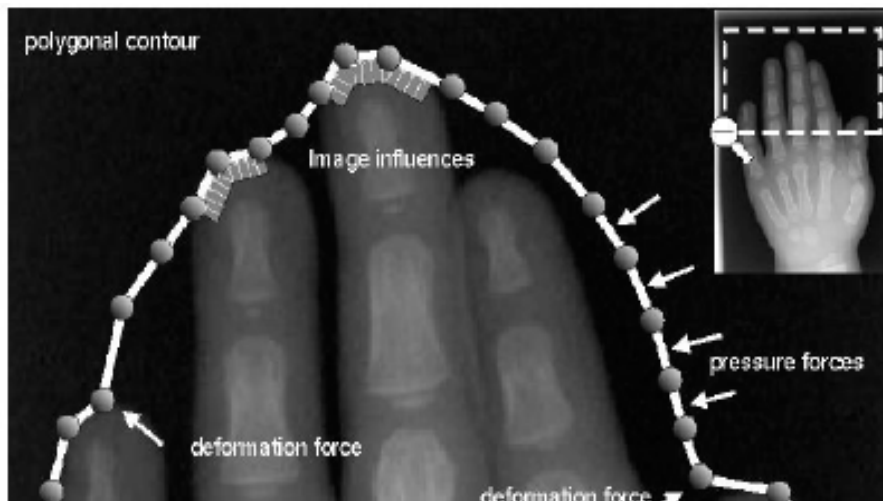


Figura 2.22. Ejemplo de las características influyentes sobre el modelo.

-Entrenamiento de la textura.

Para la segmentación no supervisada de conjuntos de imágenes heterogéneas, los parámetros deben ser ajustados sin un conocimiento a priori de los contenidos de la imagen. Por eso, se usan las estadísticas globales de la textura para determinar la similitud en apariencia entre imágenes. En una fase de entrenamiento, buscamos un subconjunto representativo de imágenes para identificar diferencias significantes en la apariencia. Para cada clase de apariencia, se escoge una imagen prototipo arbitrariamente. Los parámetros para este prototipo se establecen usando un método

automatizado que emplea una segmentación manual ejemplar. Entonces, la apariencia de estos prototipos queda descrita por su matriz de co-ocurrencia [HARALICK73]. Estos estadísticos son usados frecuentemente para clasificación de texturas.

Todos los prototipos se escalan a 256x256 pixels y la escala de grises se reduce a 64. Las matrices de co-ocurrencia se extraen para estas imágenes con un desplazamiento de 5. Los parámetros de segmentación deben ser ajustados de tal forma que se permita al balloon vencer estructuras irrelevantes de la imagen pero pararse en el borde de objetos considerables, donde ocurren cambios notables en la apariencia del tejido. La matriz resultante se comporta como un vector de características de gran dimensión. Para estas características usamos el clasificador sinérgico propuesto en [HAKEN91] para alcanzar similitudes significativas. Las características de co-ocurrencia \vec{c}_j^+ se construyen como una combinación lineal de todas las n características de los prototipos \vec{c}_j :

$$\vec{c}_j^+ = \sum_{k=1}^n a_{jk} \vec{c}_k$$

Multiplicando esto con los vectores prototipo \vec{c}_l obtenemos la matriz identidad I:

$$I = \begin{pmatrix} 1 & \vec{c}_1 \cdot \vec{c}_2 & \cdots & \vec{c}_1 \cdot \vec{c}_n \\ \vec{c}_2 \cdot \vec{c}_1 & 1 & & \vec{c}_2 \cdot \vec{c}_n \\ \vdots & & \ddots & \vdots \\ \vec{c}_n \cdot \vec{c}_1 & \vec{c}_n \cdot \vec{c}_2 & \cdots & 1 \end{pmatrix} \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & & a_{2n} \\ \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix}$$

Los a_{jk} pueden ser obtenidos invirtiendo la matriz de los productos escalares de todos los vectores de características de prototipos.

$$\begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & & a_{2n} \\ \vdots & & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{pmatrix} = \begin{pmatrix} 1 & \vec{c}_1 \cdot \vec{c}_2 & \cdots & \vec{c}_1 \cdot \vec{c}_n \\ \vec{c}_2 \cdot \vec{c}_1 & 1 & & \vec{c}_2 \cdot \vec{c}_n \\ \vdots & & \ddots & \vdots \\ \vec{c}_n \cdot \vec{c}_1 & \vec{c}_n \cdot \vec{c}_2 & \cdots & 1 \end{pmatrix}^{-1}$$

-Parametrización adaptable a la textura.

Los conjuntos de parámetros para las imágenes prototipo que resultan del entrenamiento así como los vectores de características se usan para la segmentación de imágenes heterogéneas. Durante la segmentación, un vector de características de coocurrencia es calculado para cada imagen. Los productos escalares de este vector con los vectores de todos los prototipos representan similitudes. Por esto, estos productos escalares se usan para crear automáticamente un conjunto individual de parámetros para la imagen.

Sea \vec{c}_{image} la matriz de coocurrencia de una imagen que tiene que ser segmentada y \vec{P}_j un vector que contiene los parámetros que se usan para la segmentación de la imagen prototipo j . Entonces, los parámetros requeridos \vec{P}_{image} se calculan de la similitud de la característica de coocurrencia \vec{c}_{image} y las características del prototipo \vec{c}_j^+

$$\vec{P}_{image} = \frac{\sum_{j=1}^n w_j \vec{P}_j}{\sum_{j=1}^n w_j} \quad \text{con}$$

$$w_j = \begin{cases} c_{image}^{\vec{}} \cdot c_j^{\vec{+}} & \forall \quad 0 \leq c_{image}^{\vec{}} \cdot c_j^{\vec{+}} \leq 1 \\ 1 - c_{image}^{\vec{}} \cdot c_j^{\vec{+}} & \forall \quad 1 < c_{image}^{\vec{}} \cdot c_j^{\vec{+}} \leq 2 \\ 0 & \text{en caso contrario} \end{cases}$$

Cuando se segmenta una imagen, los parámetros del modelo del balloon se interpolan linealmente desde todos los conjuntos de parámetros de los prototipos usando los pesos w_j , los cuales dependen de los productos escalares de las respectivas características. El modelo de balloon detecta contornos significantes en la imagen por un equilibrio de todas las fuerzas que actúan sobre un borde.

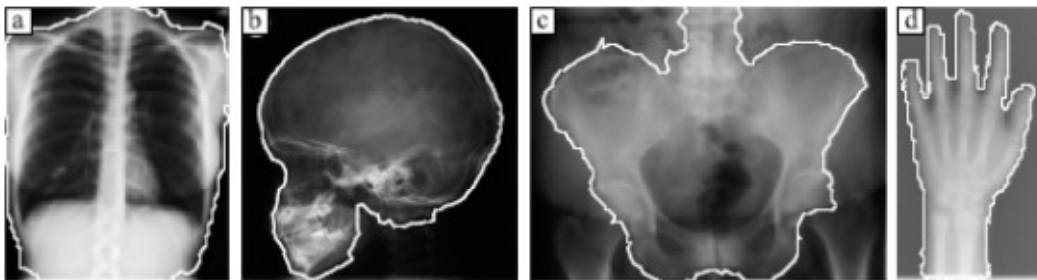


Figura 2.23. Ejemplos de segmentación sobre a) Pecho; b) cabeza; c) pelvis; d) mano.

2.3.10. Snakes Virtuales

B.Galvin, B. McCane y K. Novins [GALVIN99] presentan en 1999 su modelo de “snakes” virtuales, en la Universidad de Otago, Nueva Zelanda.

Los “snakes” virtuales se usan para detectar objetos en movimiento que se ocultan tras otros, es decir, tratan oclusiones de objetos en movimiento averiguando quien es el ocluido y quien el ocultador y así seguir la pista al objeto incluso si la oclusión es grande. Este método es simple y no requiere entrenamiento del sistema.

Cuando dos snakes se cortan, cuatro snakes virtuales son generados uno en el fondo y otro en primer plano para cada uno de los dos originales. A los dos de primer plano se les permite una cierta relajación, mientras que los dos del fondo se mueven de acuerdo con su movimiento previo. Las energías combinadas de los snakes en los dos objetos colisionantes son examinadas después de la colisión para determinar la relación de oclusión.

Definen la energía interna como:

$$E_{int} = w_{int} * \sum_{i=1}^{N-1} |v_i - v_{i-1} - p_i|$$

Que pretende preservar la forma original del snake, N es el número de puntos de control, v_i es la posición del i-ésimo punto de control, p_i es el valor medio, descendiente con el tiempo, de la cantidad $v_i - v_{i-1}$. El conjunto de vectores p_i describe la forma “media” de los contornos activos sobre los últimos frames. E_{int} calcula entonces la diferencia entre la forma actual y esta “media”.

Y la energía externa es de la siguiente forma:

$$E_{ext}(v) = w_{edge} * E_{edge}(v) + w_{texture} * E_{texture}(v)$$

La energía externa permite a los snakes llegar al lado oculto de la frontera de oclusión, tiene dos términos: uno de límite que utiliza un valor de gradiente (como el modelo tradicional) y otro de textura que usa una media ponderada de la correlación de la textura. Donde E_{edge} es la clásica $-\nabla I(x,y)$ para atraer los snakes a bordes cercanos, el término E_{texture} permite al contorno activo acercarse así mismo a la textura de la imagen. Si el contorno activo está situado en una frontera de oclusión aprenderá a acercarse a la textura del objeto en el lado oculto e ignorar la textura poco fiable del lado ocluido.

Definen E_{texture} como una media de la correlación de textura en el lado izquierdo del snake, C_{left} , y la correlación de textura en el lado derecho del snake, C_{right} , para controlar el peso que tendrán se usa la variable γ

$$E_{\text{texture}}(v) = \sum_{i=1}^{N-1} \begin{cases} C_{\text{left}}, & \gamma < 0.4 \\ C_{\text{right}}, & \gamma \geq 0.6 \\ \frac{1}{2}(C_{\text{left}} + C_{\text{right}}), & \text{en cualquier otro caso} \end{cases}$$

La correlación de textura se calcula sumando la diferencia entre el valor actual de los píxeles y el esperado. Para imágenes en color, la diferencia entre dos colores se define como la suma de las diferencias de cada canal de color. Así tendríamos:

$$C_{\text{left}} = \sum_{j=0}^{M-1} |I(V_{i-1} - n * l + d * j) - color_j|$$

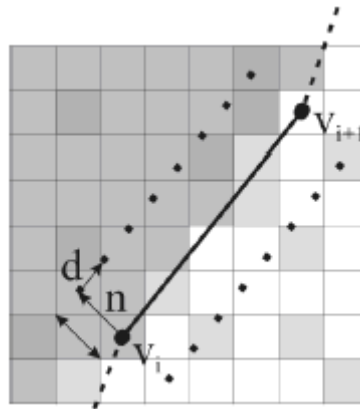


Figura 2.24. Variables usadas para definir $E_{texture}$

donde $I(u)$ es el valor del nivel de gris interpolado en u , M es el número de muestras por segundo,

$$d = (v_i - v_{i-1}) / (M - 1)$$

n es un vector unitario perpendicular a $v_i - v_{i-1}$, l es una constante pequeña, normalmente 3 ó 4 y $color_j$ es una media de degradación del color de las muestras.

Inicialmente γ tiene el valor 0.5, después de cada iteración se actualiza usando una media del valor previo y el actual de la correlación de izquierda y derecha:

$$\gamma = \gamma * k + \frac{C_{left}}{(C_{left} + C_{right})} * (1 - k)$$

Donde k es una constante.

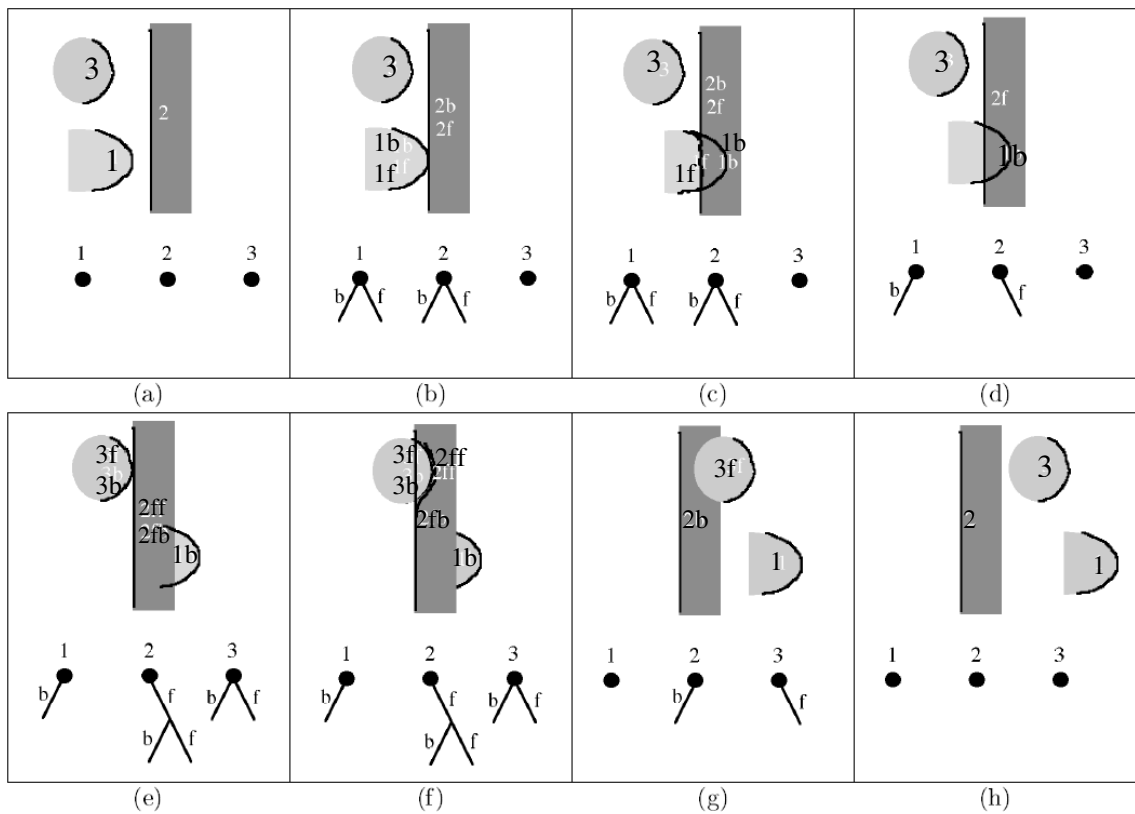


Figura 2.25. (a) el primer frame de la secuencia, los snakes se indican con la curva negra, uno para cada objeto. (b) snakes 1 y 2 se cortan generando un par de snakes virtuales cada uno. (c) el snake 1f (asociado con el primer plano) es incapaz de seguir su objeto puesto que está oculto. (d) el objeto 1 es identificado como ocultado a causa del incremento en la energía del snake 1f, resultando en el borrado de 1f y 2b. (e) el número 3 corta al virtual 2f generando un árbol de snakes virtuales. (f) el snake 2ff es incapaz de seguir su objeto puesto que está siendo ocultado e intenta curvarse alrededor del objeto ocultador. (g) los objetos 1 y 2 se separan y se borran los snakes virtuales asociados. El objeto 2ff se identifica como inválido y se borra junto con el snake 3b. (h) los objetos 3 y 2 se separan y los snakes virtuales asociados se borran.

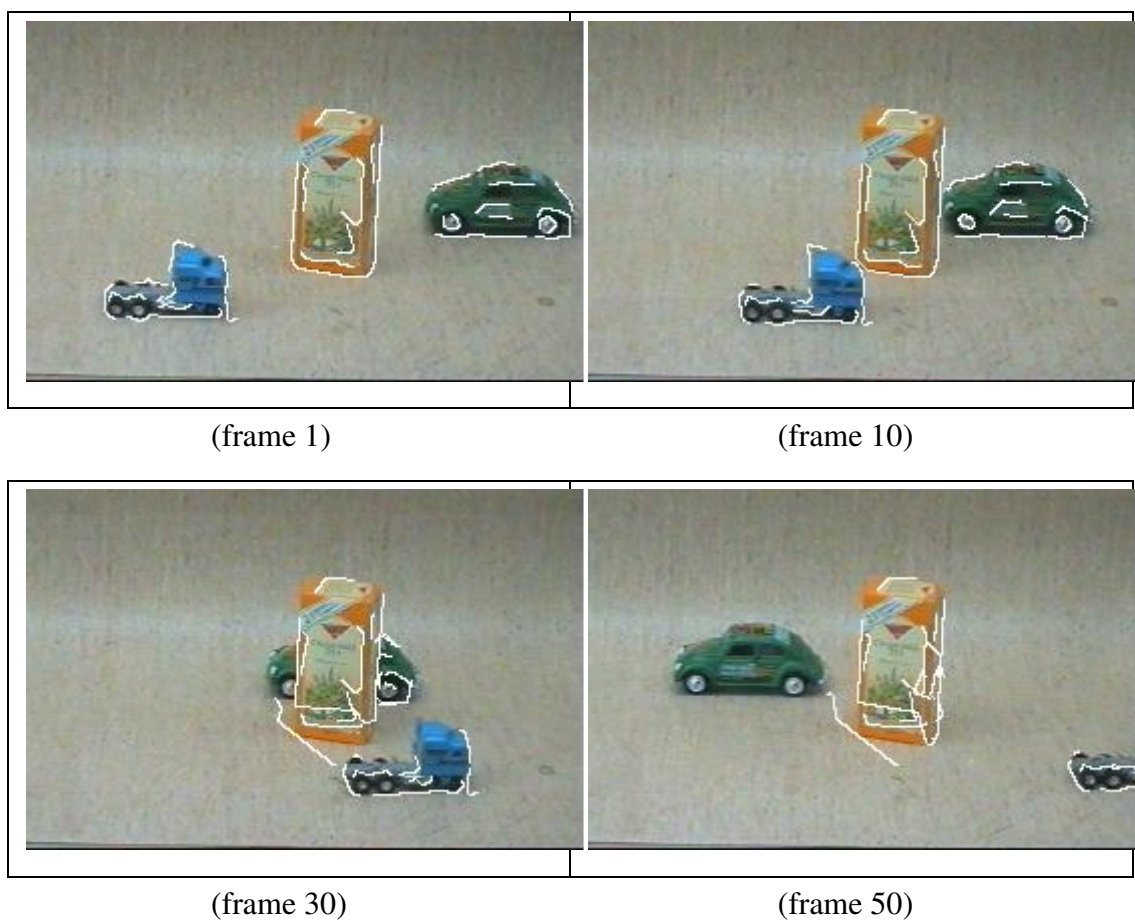
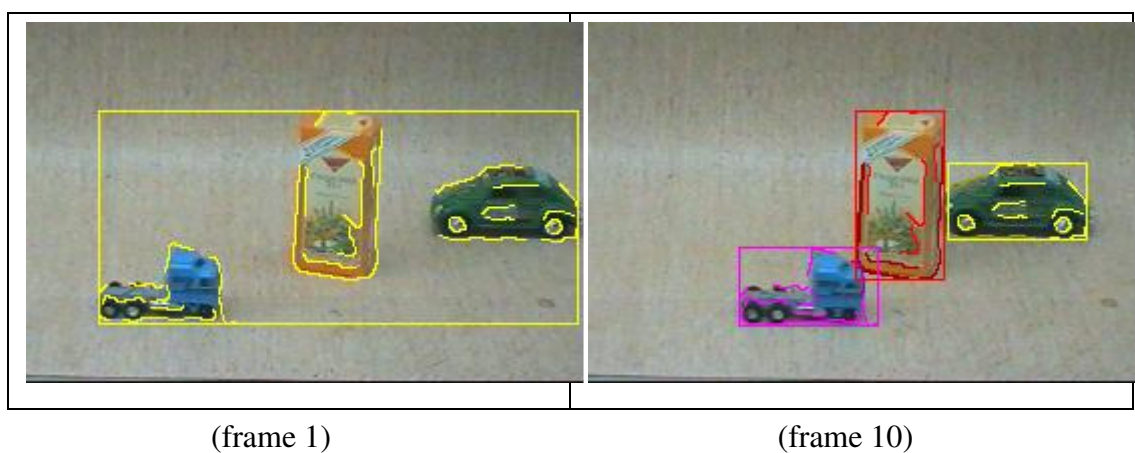


Figura 2.26. Ejemplo de seguimiento de objetos con un snake tradicional, se aprecia que el snake del coche verde está completamente desmontado a medida que el coche es ocultado por la caja. Un snake en lo alto del camión también se acerca a la caja cuando el camión pasa delante de ella.



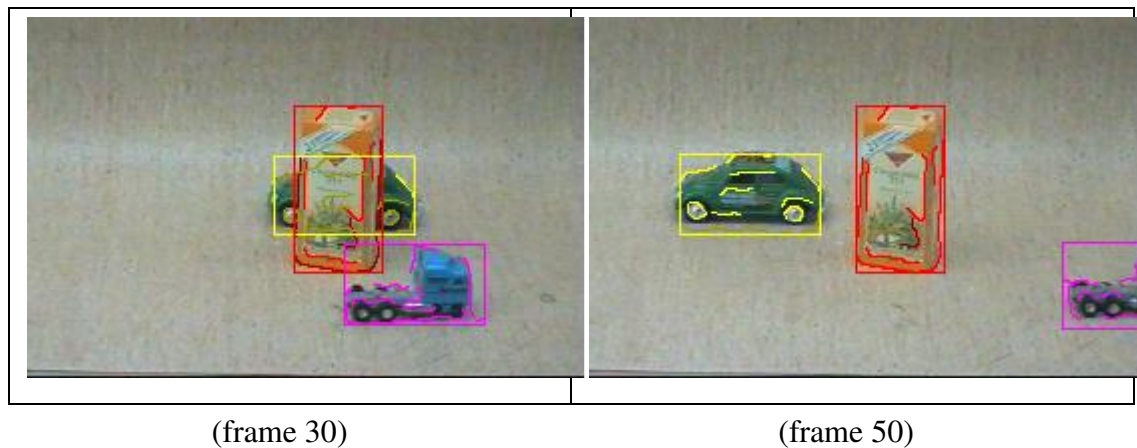


Figura 2.27. Ejemplo de evolución en snake virtual. El sistema actúa mucho mejor que la técnica estándar, siguiendo los snakes del coche incluso cuando está totalmente oculto.

2.3.11. B-Splines snakes

En 1998 Patrick Brigger, Robert Engel y Michael Unser [BRIGGER98] desarrollan sobre la plataforma JAVA[®] los “B-Splines Snakes”.

Propone una nueva formulación llamada “B-Splines Snake” (contornos activos basados en “BSplines”) donde la escala intrínseca del modelo de la curva se establece a priori. Esto minimiza el número de parámetros en las energías internas, acelerando la convergencia y facilitando el ajuste del peso de los coeficientes.

Un “B-Snake” está principalmente caracterizado por:

- Parámetro de escala que controla implícitamente el alisamiento.
- Pocos parámetros.
- Control del alisamiento implícito en el modelo.
- Eliminación de las energías internas, que son sustituidas por un nuevo concepto denominado “puntos nudo”.

Además el método de “B-Snake” permite, de forma natural, un control local sobre la curva manejando los puntos de control individualmente.

La formulación bidimensional para el B-Spline snake es la siguiente:

$$s(t) = (s_x(t), s_y(t)) = \sum_{h \in Z} c(k) \cdot \beta^n(t - k) \\ (0 \leq t \leq t_{\max} = N-1)$$

donde $s_x(t)$ y $s_y(t)$ son los componentes x e y del spline, respectivamente, ambos parametrizados por la variable curvilínea t. N denota el número de puntos de control. Mientras que la fórmula permite una representación continua de la curva, todavía está sujeta a algunas de las limitaciones de la aproximación tradicional. La idea que se presenta aquí es eliminar el término correspondiente a la energía interna e introducir un espaciado de grupo variable entre los grupos de puntos del spline.

Siendo $c(k) = (c_x(k), c_y(k))$ Coeficientes primarios del B-Spline

Introduciendo el concepto de “punto nudo”, se considera la separación entre cada uno de ellos como h, siendo éste el nuevo parámetro de alisamiento:

$$s(t) = \sum_{h \in Z} c_h(k) \cdot \beta^n\left(\frac{t}{h} - k\right) \\ (0 \leq t \leq t_{\max} = hN-1)$$

Según esta fórmula h debe ser un número entero que reduzca los grados de libertad del B-Snake, es decir, que reduzca los coeficientes del B-Spline en la misma proporción.

$$h = \frac{M}{N}$$

siendo:

M → Resolución de la curva.

N → Número de puntos de control del snake.

El final de la curva se da en $t=t_{\max}$, en el caso discreto asociaremos $M = t_{\max} + 1$.

Incrementando el número N de puntos de control se reducirá el espaciado de grupos y en consecuencia se reducirá el efecto de suavizado de la curva. El término de energía puede ser ahora formulado de la siguiente manera:

$$\xi(c(k)) = \sum_{i=0}^M g(s_x(i), s_y(i))$$

Donde $g(x,y) = L[f(x,y)]$. L es un operador de procesamiento de imagen (por ejemplo, magnitud del gradiente) que realza los contornos de interés de la imagen $f(x,y)$. El operador puede incluir un componente de suavizado para reducir la probabilidad de que el snake se quede atrapado en un mínimo local.

En la figura 53, demostramos la optimización del snake propuesto sin energías internas, contra el snake tradicional con energías internas. La comparación está basada en una imagen de test binaria consistente en una línea vertical, de la cual una pequeña parte ha sido desplazada a la izquierda. Para obtener una función de fuerza de suavizado, la imagen binaria se suaviza mediante un gaussiano bidimensional con $\sigma = 5$. La optimización se formula como un problema de minimización, y por tanto como la posición óptima del snake sobre la línea. Once puntos de control que han sido colocados manualmente a intervalos de longitud distintos caracterizan a la curva inicial. Dependiendo de las necesidades de suavizado de la curva final, se pueden anticipar dos resultados diferentes:

a) La curva resultante está verticalmente centrada sobre la línea más larga, no siendo afectada por la pequeña parte desplazada. Tal resultado corresponde a una restricción importante de suavidad.

b) La curva resultante tiene un “bache” y es atraída hacia el pequeño desplazamiento de la izquierda. Este resultado refleja un requerimiento de suavizado menos severo. Primero el snake tradicional se calcula con varios pesos para las energías internas. Cada punto de la curva está independientemente optimizado y atraído al mínimo más cercano colocando pesos para la tensión y la curvatura iguales a cero (figura 53a). Un peso de $\alpha_{stretch}=0.1$ y $\beta_{stretch}=0.1$ intenta empujar el bache hacia la derecha (figura 53b), sin embargo, no produce una curva recta todavía. Un peso de $\alpha_{stretch}=0.2$ y $\beta_{stretch}=0.2$ produce una curva casi plana (figura 53c).

Este tipo de snake es muy flexible ya que el usuario puede escoger entre un gran número de suavizados simplemente ajustando $\alpha_{stretch}$ y $\beta_{stretch}$.

El experimento de la figura 53 corresponde a un b-spline de grado uno con cero pesos, y el resultado es idéntico (figura 54a). Usando un b-spline de mayor grado, los puntos de control no son totalmente independientes. El resultado optimizado con $h=2$ se muestra en la figura 54b. Nótese que hay un punto interpolado entre dos puntos de control. El punto ayuda a atraer a la curva hacia la línea más larga. Un espaciado de grupos incrementado con $h=3$ usa dos valores interpolados entre dos puntos de control para la computación de la energía (figura 54c). En esta configuración, estos puntos controlan para atraer totalmente la curva hacia la línea más larga, y el bache desaparece.

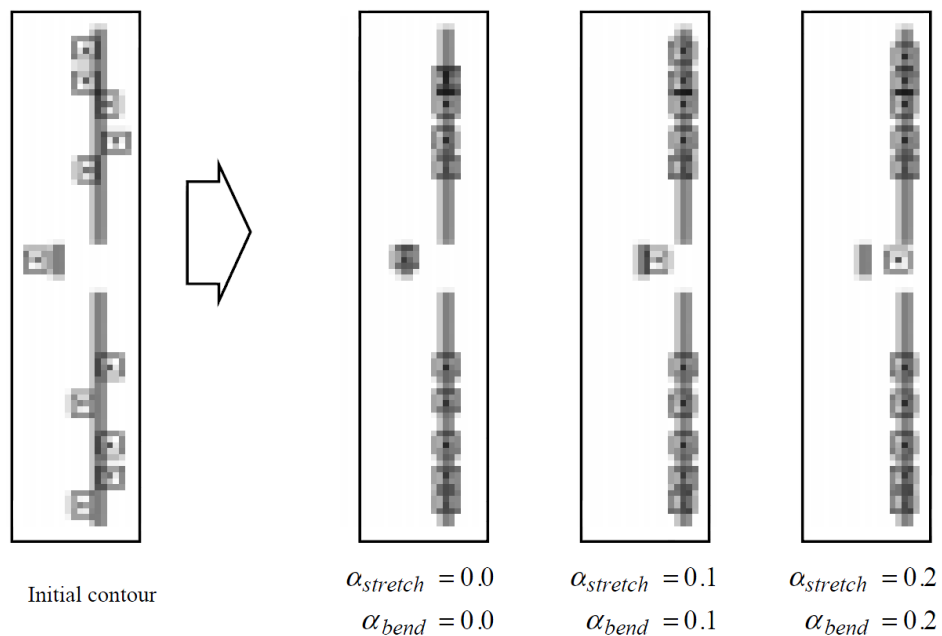


Figura 2.28. Ejemplo de snake tradicional. Contorno inicial y distintas aproximaciones de la energía [BRIGGER98]

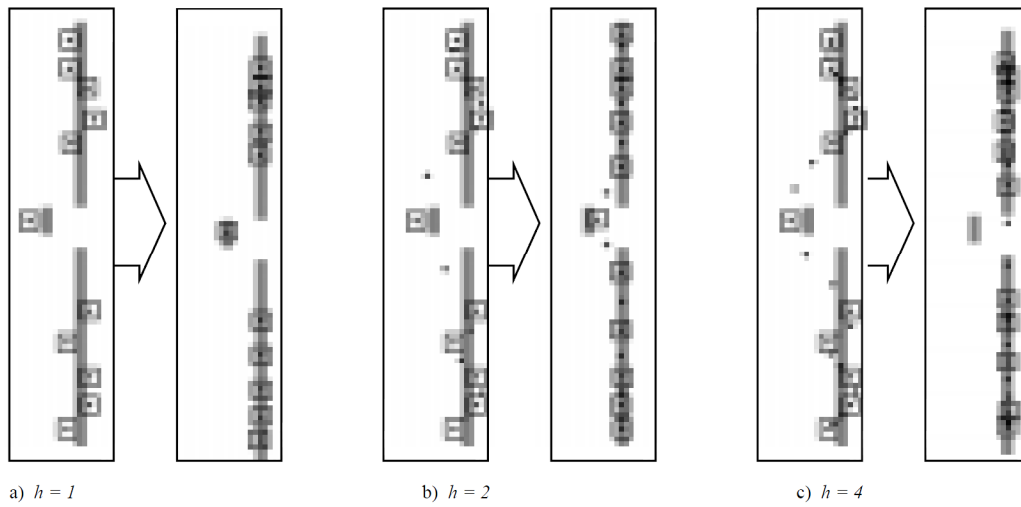


Figura 2.29. Ejemplo de B-snake. Contorno inicial y distintas aproximaciones con espaciados de nodos diferentes [BRIGGER98].

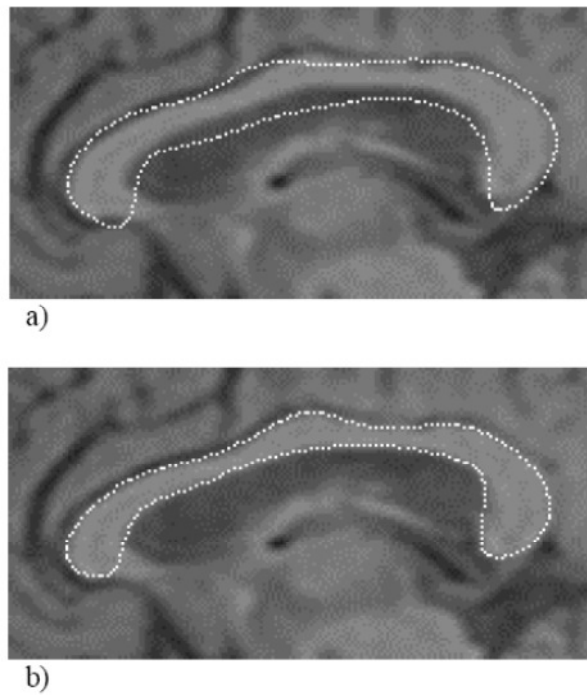


Figura 2.30. Contorno de un corpus callosum; a) Curva inicial; b) Curva optimizada automáticamente [BRIGGER98].

2.3.12. Contornos activos paramétricos

Los contornos activos a menudo se usan para eliminar el vacío existente entre el procesamiento de imágenes a bajo nivel (extracción de características) y la representación geométrica de alto nivel. Kass y col. en su artículo de 1988 [KASS88] eligieron discretizar la ley del movimiento utilizando una representación de contorno paramétrica con un esquema de integración semi-implícito. Después de Kass y col., otros autores han propuesto otras representaciones como el uso de modelos analíticos [METAXAS91], elementos finitos [COHEN92] o curvas de subdivisión [HUG99].

Osher y col. [SETHIAN96] introdujeron los contornos activos implícitos que analizaremos más adelante cuando estudiemos los Level Set. Este enfoque sirvió de base para otros autores como, por ejemplo, Caselles y col. [CASELLES97] cuyo trabajo será estudiado en profundidad durante este capítulo y los posteriores.

La principal diferencia entre la representación de contorno paramétrico e implícito corresponde a las diferencias entre esquemas de trabajo Lagrangiano y Euleriano. En los marcos de trabajo Lagrangianos, la deformación de un cuerpo se describe con la parametrización propia de la posición de un cuerpo en reposo. En los marcos de trabajo Eulerianos, esta deformación se describe con la parametrización propia de la posición actual de un cuerpo.

Los marcos Eulerianos son apropiados para la simulación y cálculo de mecánica de fluidos mientras que los Lagrangianos son apropiados para la simulación y cálculo de mecánica de sólidos.

Delingette y Montagnat [DELINGETTE01] resumieron las ventajas de uno y otro enfoque en la siguiente tabla:

	Contornos implícitos	Contornos paramétricos
Eficiencia	+	+++
Facilidad de implementación	++	+++
Cambio topológico	Sí	No
Contornos abiertos	No	Sí
Interactividad	Pobre	Buena

Tabla. Propiedades de las representaciones de contornos activos paramétricos e implícitos [DELINGETTE01]

Debido a la gran variedad de algoritmos que existen en la literatura es difícil comparar la eficiencia y la implementación de estos dos marcos de trabajo. Aunque a causa de que la actualización de un contorno implícito requiere la actualización de al menos una banda estrecha alrededor de cada contorno, las representaciones implícitas suelen ser menos eficientes que las paramétricas. Por otro lado, en los contornos implícitos la resolución está restringida por la resolución de una rejilla regular, en cambio, en los contornos paramétricos el muestreo de vértices puede no ser constante. Sin duda, la capacidad para cambiar automáticamente la topología del contorno durante su evolución, es la principal ventaja de la representación implícita.

Delingette y Montagnat estudiaron a fondo los contornos activos paramétricos y aportaron conclusiones relevantes en los tres niveles de modelado de este tipo de contornos activos [MONTAGNAT98] [DELINGETTE00] [MONTAGNAT00] [DELINGETTE01]:

-Discretización: propusieron dos algoritmos para el control del número de vértices y la distancia entre los vértices. El primer algoritmo introdujo la novedad de actualizar periódicamente el número total de vértices del contorno, que demostró ser

más eficiente que una reparametrización global. El otro algoritmo controlaba el espaciado de vértices mediante la aplicación a cada vértice del componente tangencial de la fuerza interna.

-Forma: introdujeron una fuerza interna que regularizaba la curvatura del contorno sin producir ninguna contracción del mismo. Esta fuerza interna no dependía de la parametrización del contorno si no de dos fuerzas que llamaron fuerza tangencial y fuerza normal.

-Topología: aportaron un nuevo algoritmo para conseguir cambios topológicos en sus contornos activos. El enfoque que utilizaron se puede estructurar en tres pasos. El primero, creaba una estructura de datos para hacer eficiente la detección de colisiones entre varias componentes conectadas del contorno. El segundo calculaba la intersección geométrica entre las componentes del contorno y el tercero realiza los cambios topológicos.

2.3.13. Sectored snakes

Fenster y Kender [FENSTER01] en su artículo "*Sectored Snakes: Evaluating Learned-Energy Segmentations*" indicaron como los contornos activos anteriores fallaban en la detección del borde de una estructura anatómica si había cerca otras estructuras anatómicas con bordes más fuertes. En este artículo indicaban que se puede enseñar a un contorno activo para que sea atraído a ciertas características de la imagen. Para poder encontrar un objeto dentro de una imagen, el modelo debe saber a qué se parece. Para esto, debemos entrenar el modelo para que aprenda los valores de similitud de las cualidades de la imagen seleccionada asociadas con el borde de esa clase de objeto. Pero puede que las características del borde buscado no sean uniformes en toda su longitud, de forma que pueden variar con la posición. Puesto que estas variaciones espaciales de frontera son comunes a las imágenes de un dominio particular, tuvo sentido crear un contorno activo que aprendiera a buscar por separado en fragmentos separados de su longitud. El objetivo final es obtener una

frontera precisa en toda su longitud, mejor que sólo en unas pocas porciones del contorno activo.

Fenster y Kender [FENSTER01] dividieron el contorno en un número fijo de sectores de igual longitud (este es el origen del nombre “sectored snakes”), cada uno con un entrenamiento separado (figura 2._64).

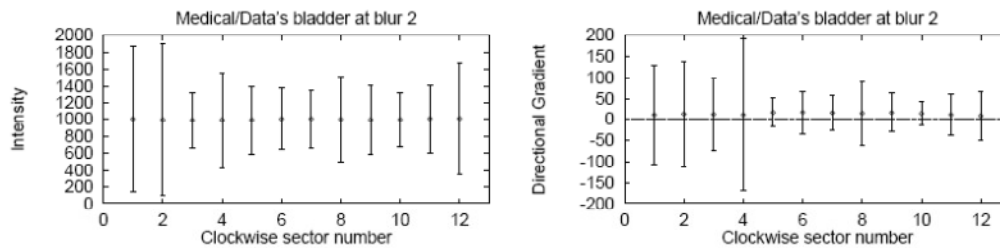


Figura 2.31. Cada sector de un sectored snake ha aprendido por separado una intensidad de imagen y un gradiente en la normal de la dirección del snake. Las barras de intensidad (izquierda) son más altas en las regiones 12, 1 y 2, correspondientes a las variaciones de intensidad más grandes. [FENSTER01]

2.3.14. Watersnakes

Los Watersnakes fueron creados por Nguyen, Worring y Boomgaard [NGUYEN03], partiendo del algoritmo *watershed* [VINCENT91][BEUCHER92] que había obtenido buenos resultados en la fase de segmentación. Sin embargo, el algoritmo watershed no permite la incorporación de una información a priori tal y como hacen los métodos de segmentación basados en minimización de energía. En particular no hay control sobre el suavizado de los resultados de la segmentación. Estos autores presentaron el algoritmo de segmentación watershed como un problema de minimización de energía usando la definición basada en la distancia de la línea divisoria del watershed. De esta manera, se pueden tener consideraciones a priori sobre suavidad añadiendo la longitud del contorno a la función de energía. Esto nos lleva a un nuevo método llamado Watersnakes, integrando las propiedades de la segmentación watershed y de la segmentación basada en energía.

Nguyen y col. contemplaron[NGUYEN03] dos enfoques principales en la segmentación: basada en energía y basada en watershed. En el primer enfoque, la segmentación se obtiene como resultado de la minimización de una función de energía. El segundo enfoque está basado principalmente en el algoritmo watershed de morfología matemática.

En el primer enfoque, la energía se compone de un término conducido por datos y un término de ajuste para equilibrar la proximidad a los bordes y la suavidad. El propósito del segundo término es imponer un conocimiento a priori sobre el resultado de la segmentación, normalmente la suavidad del contorno de la región. Los términos conducidos por datos usados en la literatura pueden ser clasificados en dos clases: basados en contornos (como los snakes) y basados en regiones.

Los métodos basados en snakes (como el definido originalmente por Kass y col. [KASS88]) usan como datos de entrada información de gradiente. Los métodos basados en regiones usan una función de energía global, calculada para toda el área de la región, no sólo para sus fronteras.

El algoritmo watershed de morfología matemática toma un enfoque diferente, comparado con los métodos basados en energía. La entrada es una función de relieve que muestra la presencia de vértices, donde el gradiente morfológico es una elección habitual para calcular tal relieve. Mirando esta función como un paisaje de montaña, las fronteras de los objetos están determinadas como líneas watershed (divisorias de agua). En [BEUCHER92] y [VINCENT91] el algoritmo watershed está implementado como un crecimiento de regiones, donde las semillas son los mínimos regionales del relieve.

Para poder utilizar la idea de la segmentación watershed como un contorno activo, necesitamos presentarla como una función de energía que tenemos que minimizar. Por esto, para diseñar los watersnakes, se le impuso información a priori añadiéndole a la energía los términos de regularización apropiados.

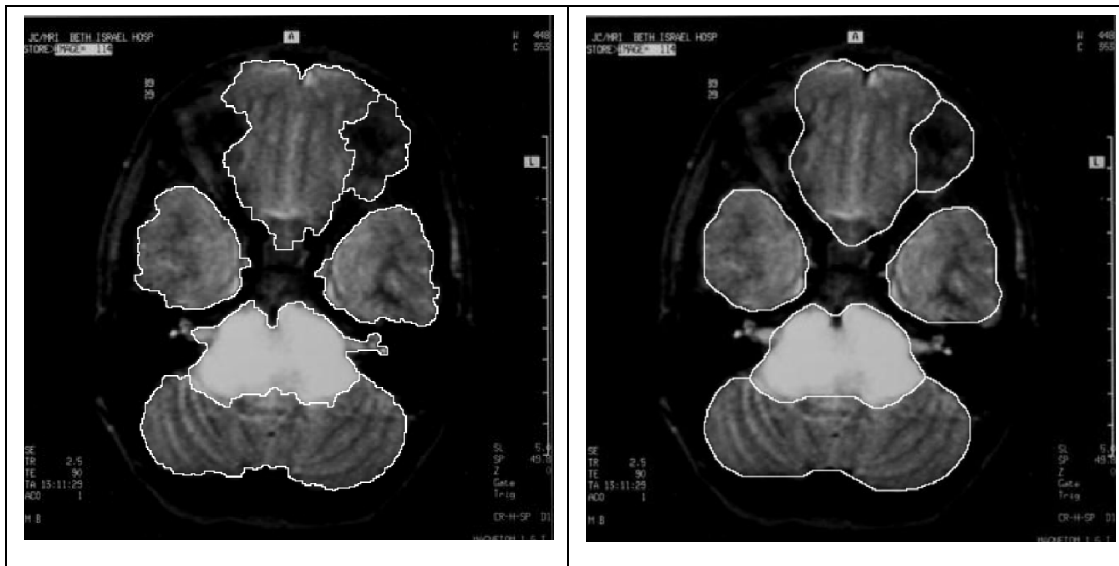


Figura 2.32. Izquierda: resultado de la segmentación original watershed sobre una imagen cerebral. Derecha: resultado de la segmentación watersnake basada en discretización de la energía. Se aprecia como en comparación con la segmentación original watershed de la figura de la izquierda, los resultados de la figura son más suaves, pero todavía identifica los objetos principales[NGUYEN03].

2.3.15. Level Set

Los Level Set fueron introducidos en 1995 por Malladi, Sethian y Vemuri [MALLADI95] partiendo de un trabajo previo realizado por Osher y Sethian [OSHER88]. Si representamos una frontera como una curva como la de la figura siguiente, la expansión de la curva hacia el punto (x,y) se produce según una función de velocidad F que se mueve en dirección normal a la propia curva.

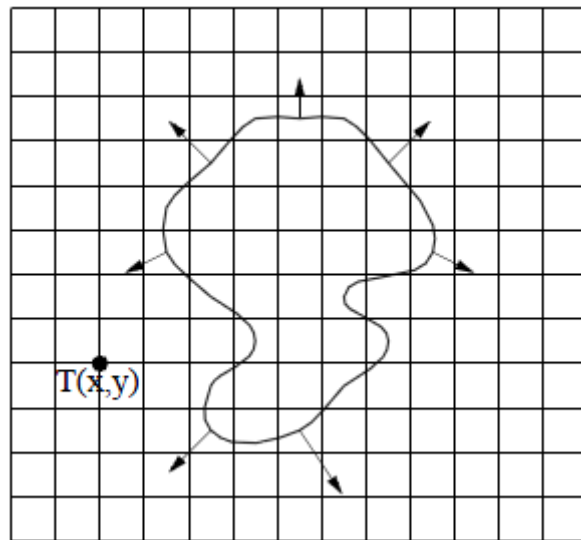


Figura Nueva-2.33. Expansión de la curva hacia (x,y) con $F>0$

Según la teoría de los Level Set se hace coincidir la curva con un conjunto de nivel cuando $t=0$. Posteriormente, la curva irá evolucionando según la función de velocidad F , de forma que en diferentes tiempos (diferente valores de t), la curva irá cambiando de forma. En general, el movimiento de la curva está asociado a un level set de nivel 0 de una función distancia $\psi(x, y, t=0) = \pm d$, donde d tendrá valores positivos en puntos que estén fuera de la curva y valores negativos en puntos que estén dentro de la curva.

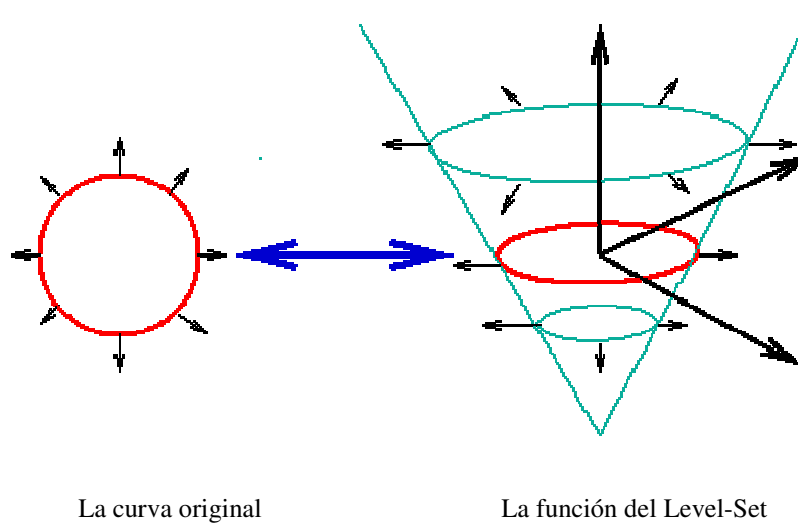


Figura 2.34. La curva es una intersección entre la superficie y el plano xy, basado en [SETHIAN06].

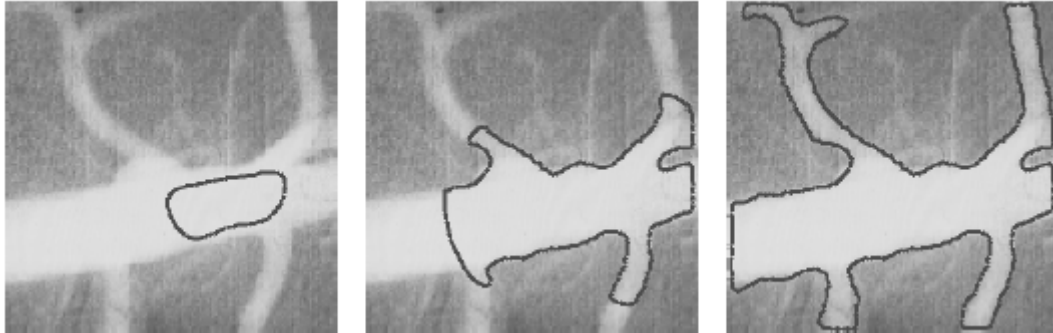


Figura 2.35. Ejemplo de aplicación de los Level Set a una imagen médica [SETHIAN96b]

Siguiendo esta misma filosofía, surgieron posteriormente otros modelos, como por ejemplo, los contornos activos geodésicos [CASELLES97].

2.3.16. Contornos activos geodésicos

Caselles, Kimmel y Sapiro [CASELLES97] introdujeron un nuevo modelo de contornos activos llamados Contornos Activos Geodésicos basados en la misma filosofía que el modelo desarrollado previamente por Osher y Sethian [OSHER88]. Este modelo está inspirado en las curvas geodésicas. Una curva geodésica es el camino de mínima distancia entre dos puntos. Esto es especialmente útil en imágenes con grandes diferencias en su gradiente.

Para explicar este modelo, podríamos partir de un modelo clásico de snake

$$E(C) = \alpha \int_0^1 |C'(q)|^2 dq + \beta \int_0^1 |C''(q)|^2 dq - \lambda \int_0^1 |\nabla I(C(q))| dq$$

donde C es la curva y los dos primeros términos constituyen la energía interna y controlan la suavidad del contorno, el tercer término es la energía externa y atrae el contorno al objeto en la imagen. En este modelo se considera el caso particular de

que β valga 0, ya que esto permite hallar la relación entre los snakes y los contornos basados en la evolución de curva geométrica. Así, la expresión anterior quedaría

$$E(C) = \alpha \int_0^1 |C'(q)|^2 dq - \lambda \int_0^1 |\nabla I(C(q))| dq$$

Y teniendo en cuenta el clásico principio de Maupertuis [DUBROVIN84], se llega finalmente a la conclusión de que minimizar la energía es equivalente a encontrar una curva geodésica. Y recurriendo al principio de Fermat¹, la expresión final a minimizar sería:

$$\int_0^1 g(|\nabla I(C(q))|) |C'(q)| dq$$

que sería equivalente a tener un nivel de energía fijo $E_0=0$ (ley de conservación de la energía), o dicho de otro modo, la energía interna sería igual a la energía externa en la ecuación

$$E(C) = \alpha \int_0^1 |C'(q)|^2 dq - \lambda \int_0^1 |\nabla I(C(q))| dq$$

ya que en un borde ideal se espera que la energía externa sea 0 y serán atraídos a los bordes en mayor medida.

2.3.17. Contornos activos sin bordes

Chan y Vese [CHAN01] definieron un modelo al que llamaron contorno activo sin bordes. Este modelo define una función de energía que al minimizarla detecta el borde de un Objeto de Interés (OI) en la imagen.

$$F_1(C) + F_2(C) = \int_{interior(c)} |I(x) - v|^2 dx + \int_{exterior(c)} |I(x) - u|^2 dx$$

¹ El principio de Fermat fue enunciado en 1657. Caselles, en [CASELLES97] lo reprodujo como "en un medio isotrópico, el camino tomado por un rayo de luz para ir de un punto A a un punto B son los extremos correspondientes al tiempo transversal. Estos caminos son geodésicos"

Esta función de energía está compuesta de dos términos $F_1(C)$ y $F_2(C)$ distinguiendo, básicamente, cuatro situaciones diferentes:

- Si la curva está fuera del OI: $F_1(C) > 0$, $F_2(C) \approx 0$, $\text{ajuste} > 0$,
- Si la curva está dentro del OI: $F_1(C) \approx 0$, $F_2(C) > 0$, $\text{ajuste} > 0$,
- Si la curva está parcialmente dentro y parcialmente fuera del OI: $F_1(C) > 0$, $F_2(C) > 0$, $\text{ajuste} > 0$ y
- Si la curva está encima del contorno, $F_1(C) \approx 0$, $F_2(C) \approx 0$, $\text{ajuste} = 0$.

En la figura siguiente podemos ver, de izquierda a derecha y de arriba a abajo, estas cuatro posibilidades. El término de ajuste se minimiza sólo cuando la curva está en el borde del objeto.

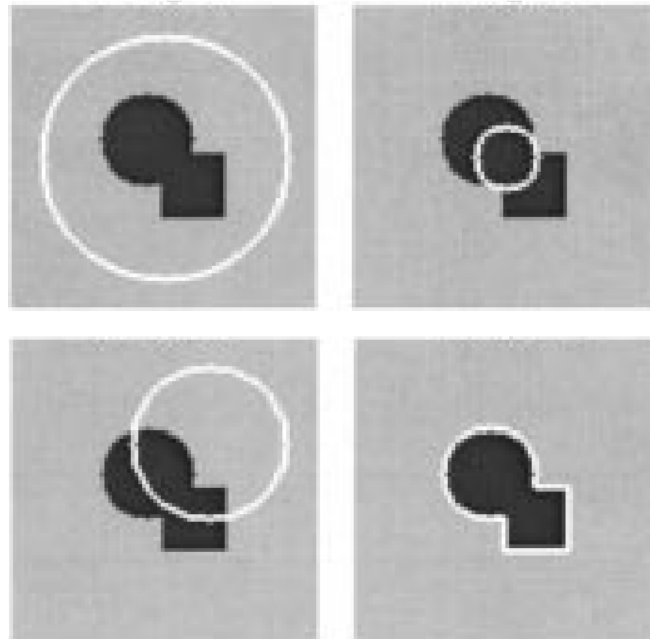


Figura 2.36. Todos los posibles casos en la posición de la curva respecto al objeto de interés [CHAN01].

Además de minimizar el término de ajuste, también se incluyen en la formulación otros términos reguladores como la longitud de la curva C y el área de la región que contiene la curva. Chan y Vese, para minimizar la función, se basaron en el funcional de Mumford y Shah [MUMFORD89]. Desarrollaron la formulación utilizando una función lipschitziana, una función de Heaviside y la función delta de

Dirac. Con todo ello, la solución al problema se reduce a un problema de partición mínima. Para más información sobre el desarrollo de la formulación de este modelo, se aconseja leer [CHAN01].

2.3.18. Método de Li

Este método fue ideado por Li, Kao, Gore y Ding [LI08] para solucionar los

$$\begin{aligned}
 E(\phi) = & \lambda_1 \int_{\Omega} \int_{\Omega} K_{\sigma}(x-y) |I(y) - f_1(x)|^2 H(\phi)(x) dy dx \\
 & + \lambda_2 \int_{\Omega} \int_{\Omega} K_{\sigma}(x-y) |I(y) - f_2(x)|^2 (1 - H(\phi(x))) dy dx \\
 & + \nu \int_{\Omega} \delta(\phi(x)) \|\nabla \phi(x)\| dx + \mu \int_{\Omega} \frac{1}{2} (\|\nabla \phi(x)\| - 1)^2 dx,
 \end{aligned}$$

problemas que encontraban otros métodos debido a la heterogeneidad en la intensidad de algunas imágenes. Esta falta de homogeneidad se encuentra muy habitualmente en las imágenes radiológicas.

El criterio de Energía que caracteriza a este método es el siguiente: donde $I(x)$ es la intensidad en el pixel x , H es la función de Heaviside y K_{σ} es un kernel gaussiano definido como:

$$K_{\sigma}(U) = \frac{1}{2\pi^{n/2}\sigma^n} e^{-\|u\|^2/2\sigma^2}$$

con un parámetro de escala $\sigma > 0$. f_1 y f_2 son dos funciones c

$$\begin{aligned}
 f_1(x) &= \frac{k_{\sigma} * \left((1 - H(\phi(X))) I(X) \right)}{K_{\sigma} * H(1 - \phi(x))} \\
 f_2(x) &= \frac{k_{\sigma} * \left((1 - H(\phi(X))) I(X) \right)}{K_{\sigma} * H(1 - \phi(x))}
 \end{aligned}$$

centradas en el pixel x y calculadas en cada iteración como: donde λ_1 y λ_2 son dos constantes a las que algunos autores suelen asignarles el valor 1 [DIETENBECK10].

Las dos primeras integrales de $E(\phi)$ sirven para lograr el ajuste gracias al kernel gaussiano K_{σ} , que controla el tamaño de la región local centrada en x . La tercera

integral corresponde al término de regularización que permite suavizar la curva durante su evolución. El último término es un término de regularización que fuerza al level set a mantener las propiedades de la distancia con signo durante el proceso de evolución.

Este algoritmo es capaz de segmentar objetos no homogéneos debido a la localización introducida por f_1, f_2 y K_σ .

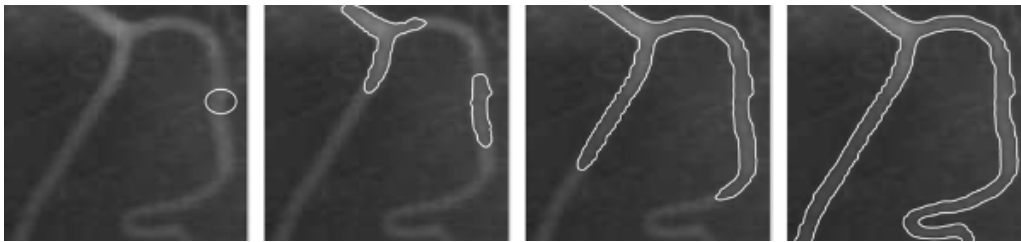


Figura 2.37. Evolución de la curva desde el contorno inicial (primera columna) al contorno final (cuarta columna). [LI08]

2.3.19. Método de Bernard

Bernard, Friboulet, Thevenaz y Unser [BERNARD09] definieron su método como una función paramétrica continua basada en B-splines. La función $\phi(x)$ la definieron como una combinación lineal de B-splines

$$\phi(x) = \sum_{k \in \mathbb{Z}^d} c[k] \beta^n \left(\frac{x}{h} - k \right)$$

donde β^n es un B-spline simétrico uniforme de grado n y d -dimensional. h es un factor de escala que permite configurar el grado de suavidad durante la evolución del contorno.

Así, el criterio de energía es

$$E(\phi) = \int_{\Omega} F(I(x), \phi(x)) dx$$

siendo Ω un subconjunto abierto de \mathbb{R}^d , siendo d el número de dimensiones de la imagen.

Este método está basado en regiones e intenta separar la imagen en dos regiones homogéneas según su valor medio. Otra característica de este método es que calcula la evolución del level set de toda la imagen, de forma que los contornos pueden llegar a aparecer en zonas de la imagen lejanas al contorno con el que se inicializaron.

2.3.20. Método de Shi

Shi y Karl [SHI08] definieron un algoritmo rápido basado en una versión de la evolución de la curva de los level set. Una de las características principales de este método es que no necesita resolver ecuaciones diferenciales parciales para calcular la evolución de la curva. La función implícita se define como una función a trozos que toma sólo cuatro valores diferentes (-3, -1, 1 y 3) correspondiendo a los puntos interiores, los puntos interiores adyacentes a la curva, los puntos exteriores adyacentes a la curva y los puntos exteriores, respectivamente.

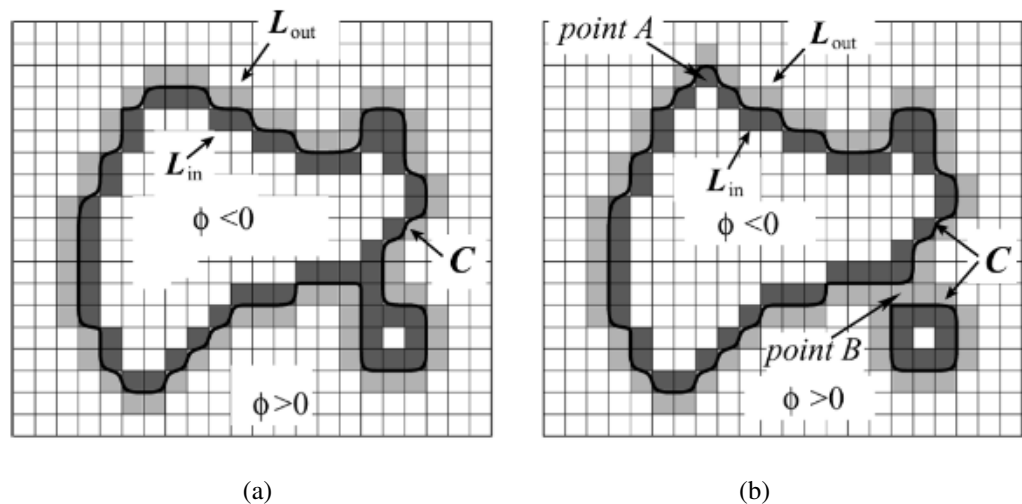


Figura 2.38. a) Representación implícita de la curva C . En esta rejilla se pueden definir dos listas de puntos vecinos L_{in} y L_{out} . b) El movimiento de la curva representada implícitamente puede conseguirse intercambiando puntos entre L_{in} y L_{out} . [SHI08]

El proceso de evolución de la curva se consigue mediante un algoritmo que funciona de acuerdo a la fórmula de la velocidad general $F=F_d + F_{int}$, y se consigue mediante un algoritmo que funciona en dos pasos bien diferenciados:

-durante un número preestablecido de iteraciones la curva evoluciona según la información contenida en F_d ,

-entonces la curva se suaviza durante otro número preestablecido de iteraciones. El término de regularización se calcula para cada punto de L_{in} y L_{out} usando un filtro gaussiano. El número de iteraciones en el ciclo de regularización puede ser configurado con diferentes valores [SHI08]. Dietenbeck, Alessandrini, Friboulet, y Bernard le asignaron, por defecto, el valor 3 [DIETENBECK10].

2.3.21. Método de Lankton

Lankton y Tannenbaum [LANKTON08] crearon este método para solucionar problemas que surgían en objetos con perfil heterogéneo como en el que se muestra en la siguiente figura.

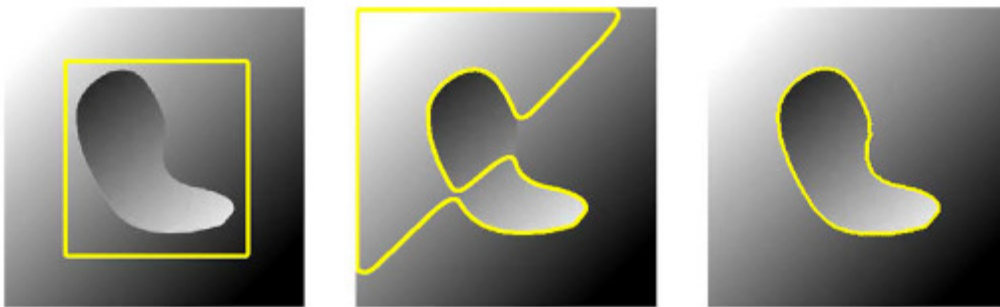


Figura 2.39. Imagen sintética de una mancha con intensidad heterogénea con un fondo de intensidad heterogénea similar. (a) Contorno inicial. (b) Resultado de una segmentación no exitosa basada en regiones [CHAN01]. (c) Resultado exitoso de la segmentación con una técnica basada en fronteras [CASELLES97]. Figura procedente de [LANKTON08].

El método de Lankton y Tannenbaum crea varios tipos de energías locales en cada punto de la curva. Para calcular estas energías locales, los píxeles vecinos se clasifican como interior o exterior a la curva.

Sea I una imagen definida en el dominio Ω y sea C un contorno cerrado que se corresponde con el level set cero y una función de distancia con signo ϕ . El interior de la curva C se puede definir como una función de Heaviside $H\phi(x)$ y el exterior de C como $1 - H\phi(x)$.

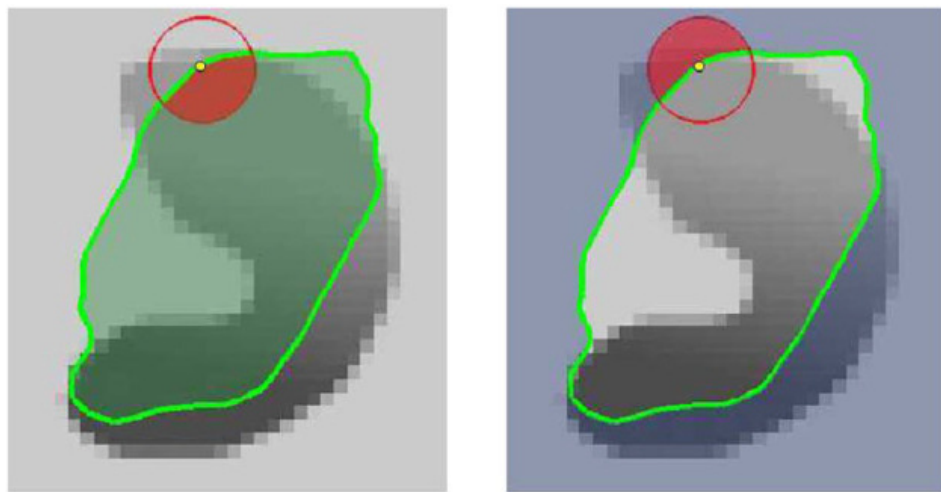


Figura 2.40. En cada punto de la curva se considera un círculo. Este círculo es separado por el contorno en regiones locales interiores y regiones locales exteriores. En ambas imágenes, el punto x se representa mediante un pequeño punto. La vecindad $B(x, y)$ se representa con un círculo rojo más grande. En (a) la parte sombreada del círculo indica el interior local es y en (b) la parte sombreada del círculo indica el exterior local [LANKTON08].

Con $B(x, y)$ se define la función de energía y, añadiéndole un término de regularización para mantener la uniformidad de la curva (al que se le asigna un peso λ) finalmente queda:

$$E(\phi) = \int_{\Omega} \delta(\phi(x)) \int_{\Omega} B(x, y) \cdot F(I(y), \phi(y)) dy dx + \lambda \int_{\Omega} \delta(\phi(x)) \|\nabla \phi(x)\| dx$$

La función F es una medida de energía interna genérica usada para representar la adherencia local al modelo dado en cada punto del contorno. En [LANKTON08]

se muestra como cualquier energía basada en regiones puede modificarse y reescribirse como una fuerza F para ser incluida en este modelo. Veamos varias formas en las que Lankton y col. definieron esta fuerza F :

-Como la Energía utilizada por Chan [CHAN01] (y que Lankton y col. llaman Energía de Modelización Uniforme, E_{UM}) y su fuerza correspondiente F_{UM} . En este caso, la energía modela el fondo y el primer plano como intensidades constantes, representados por sus valores medios

$$F_{UM} = \mathcal{H}\phi(y)(I(y) - u_x)^2 + (1 - H)(I(y) - v_x)^2$$

-Como la Energía de Separación Media (E_{MS}) y su fuerza correspondiente F_{MS} , basada en el modelo de Yezzi, Tsai y Willsky [YEZZI02], que asume que las intensidades medias del primer plano y del fondo son tan diferentes como sea posible:

$$F_{MS} = (u_x - v_x)^2$$

-Como la Energía de Separación de Histogramas, que es una energía más compleja y que compara los histogramas completos del primer plano y del fondo.

Resumiendo, podemos concluir que funciona bien para objetos con un perfil heterogéneo y para los que sería difícil obtener buenos resultados utilizando un método global, aunque tiene el inconveniente de que, al ser un método local, es muy sensible a la inicialización.

2.3.22. Snakes a Nivel de Píxel.

Los Snakes a Nivel de Pixel, también conocidos como Pixel Level Snakes (PLS) fueron creados por Vilariño, Brea, Cabello y Pardo [VILARIÑO00]. Este tipo de contorno activo se basa principalmente en dos procesos que se van ejecutando iterativamente, estos dos procesos realizan operaciones morfológicas binarias y locales que llevan a cabo un proceso de expansión y otro proceso de adelgazamiento. Estos procesos se realizan en las direcciones norte, sur, este y oeste, y son

implementados mediante redes neuronales celulares masivamente paralelas lo que consigue que sean procesos muy rápidos. En este caso se asocia a cada pixel de la imagen un procesador local que se relaciona con los procesadores "asociados" a los píxeles vecinos, de ahí su nombre (snakes a nivel de pixel, Pixel Level Snakes).

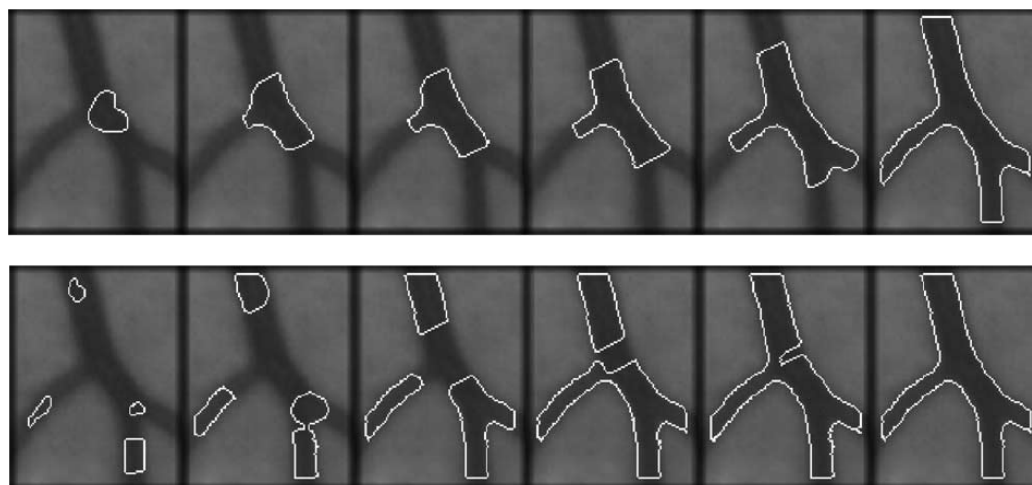


Figura 2.41. Segmentación de un vaso sanguíneo en un angiograma de retina. En la primera fila se muestra un contorno inicial y en la segunda se muestran múltiples contornos iniciales. El contorno final coincide en ambos casos [VILARIÑO03].

2.4. Conclusiones

Después de la revisión bibliográfica que hemos realizado, podemos concluir que el éxito de un método de contornos activos depende bastante de los parámetros de definición del modelo y de la aplicación particular.

Básicamente, podemos clasificar los contornos activos en modelos paramétricos y modelos implícitos. Los primeros permiten un gran control de la movilidad debido a las energías de guía y también a la posibilidad de establecer un control manual sobre las características locales. Además permiten un mejor control de la deformación del contorno en términos de rigidez y elasticidad permitiendo también penalizar las deformaciones que van más allá de una forma establecida a priori. Estas características hacen a los modelos paramétricos particularmente

convenientes para tareas apoyadas en un conocimiento previo del dominio de aplicación y la forma aproximada del objeto de la escena.

Por otro parte, los modelos paramétricos presentan baja flexibilidad tal que no pueden obtener buenos resultados en la segmentación de objetos altamente irregulares con concavidades profundas y estrechas o protuberancias. Asimismo, la mayoría no tiene la capacidad de adaptarse topológicamente al número de objetos de la imagen. Existen alternativas como los contornos activos topológicamente adaptables (también llamados T-snakes) derivados de los modelos paramétricos que permiten transformaciones topológicas y que se puedan alcanzar formas de contorno irregulares.

De todo lo revisado, también podemos llegar a la conclusión de que, dado un modelo paramétrico, el éxito del proceso de segmentación dependerá fuertemente de los términos potenciales externos considerados y el peso de estos; de la inicialización que se haga del contorno activo, especialmente la proximidad al objeto de interés; del número de vértices de discretización y las técnicas de simulación numérica utilizadas; y, por último, de la técnica usada para la minimización de energía (ecuaciones de Lagrange, programación dinámica, algoritmos greedy, etc.)

Los modelos implícitos gozan de la ventaja de poseer una mayor flexibilidad que los modelos paramétricos, lo que les permite llegar a cambios topológicos fácilmente. Aunque tienen la desventaja de que son más difíciles de controlar y de aplicar restricciones.

Capítulo 3

Metodología

3.1. Introducción.

En este capítulo comenzaremos exponiendo los problemas encontrados en el uso de los contornos activos para la segmentación de imágenes radiológicas y continuaremos explicando los métodos desarrollados para solucionar los problemas encontrados.

3.2. Segmentación con contornos activos.

En este apartado vamos a presentar en primer lugar los problemas encontrados en los contornos activos existentes previamente y que se han estudiado y probado en profundidad. De los ensayos realizados y los problemas encontrados en estos contornos activos previos, se ha extraído una experiencia y un conocimiento que ha servido para desarrollar

posteriormente los nuevos métodos de segmentación creados para este trabajo y que también se basan en contornos activos. Más concretamente, se han desarrollado varios métodos que se enmarcan dentro de los Contornos Activos Auto-Reconfigurables, que se han creado en el marco de este trabajo.

3.2.1. Problemas encontrados en contornos activos actuales.

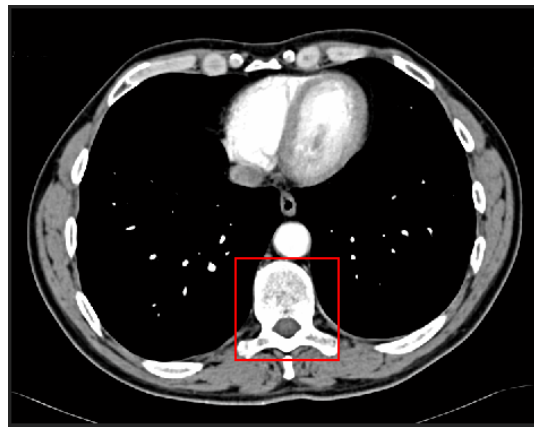
Como hemos visto en el capítulo de *Antecedentes*, los algoritmos de contornos activos suelen fallar en muy diversas circunstancias, dependiendo principalmente del tipo de imagen y del tipo de contorno activo utilizado. En el capítulo de resultados veremos también algunos casos en los que fallaron algunos de los contornos activos que se probaron.

Vamos a exponer a continuación algunos de los problemas encontrados durante los experimentos realizados con diversos algoritmos de contornos activos. Para ello, vamos a evidenciar ahora casos en los que falla cada algoritmo analizado para varias imágenes. Posteriormente, en el capítulo 4 de *Resultados*, cuando se haga una comparativa de los algoritmos desarrollados con estos otros que analizamos a continuación, siempre escogeremos el mejor resultado que hayamos sido capaces de encontrar para cada algoritmo y para cada imagen, para comparar el mejor resultado que es capaz de dar cada algoritmo para cada imagen. Ahora, en este apartado, mostraremos los mejores resultados, pero también revelaremos los casos en los que fallan, para intentar presentar en que aspectos pueden ser mejorados estos algoritmos.

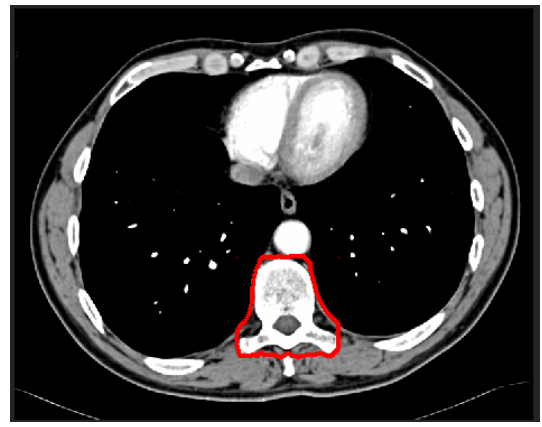
Todos los algoritmos se evaluaron con el mismo conjunto de imágenes. Iremos exponiendo, para cada algoritmo, los resultados obtenidos para cada imagen y, en una fase posterior de análisis de esos experimentos, indicaremos los problemas encontrados.

3.2.1.1. Problemas encontrados en algoritmo de Caselles

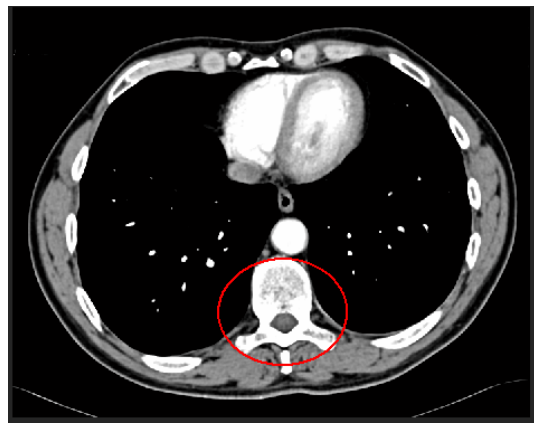
Observemos en primer lugar los resultados obtenidos en los ensayos realizados con el algoritmo de Caselles [CASELLES97] y posteriormente extraeremos algunas conclusiones. Empezamos con algunos experimentos realizados para la imagen 1:



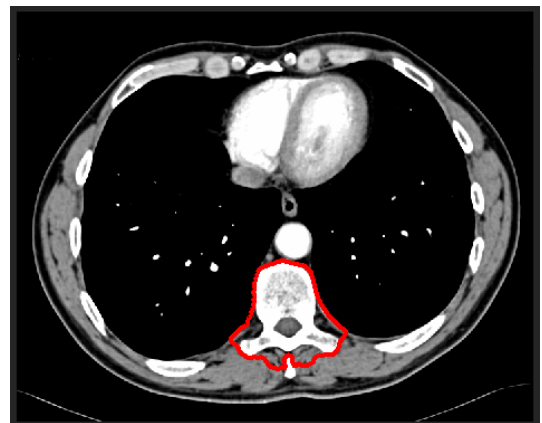
Prueba 1002-Contorno inicial



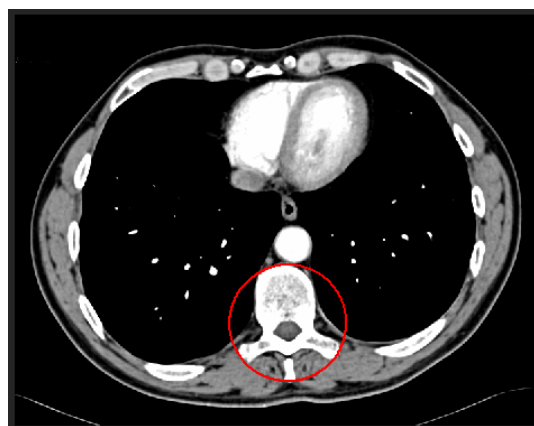
Prueba 1002-Contorno final



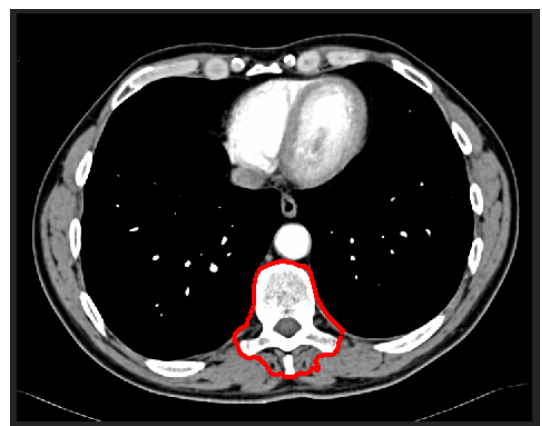
Prueba 1003-Contorno inicial



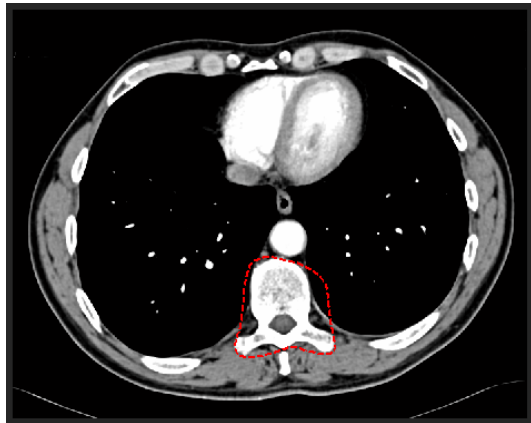
Prueba 1003-Contorno final



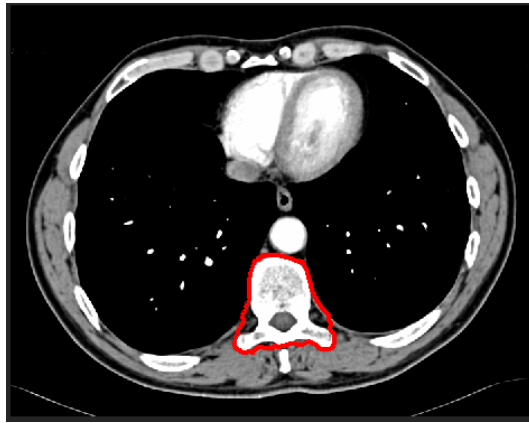
Prueba 1004-Contorno inicial



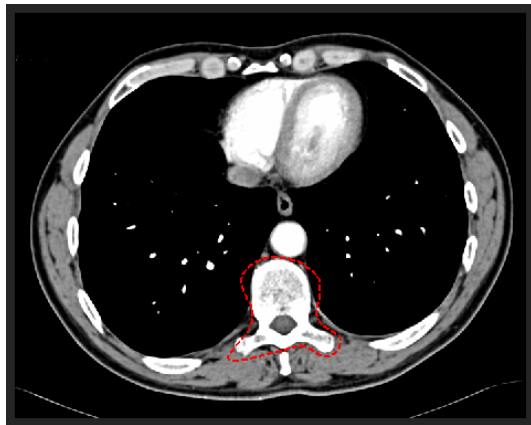
Prueba 1004-Contorno final



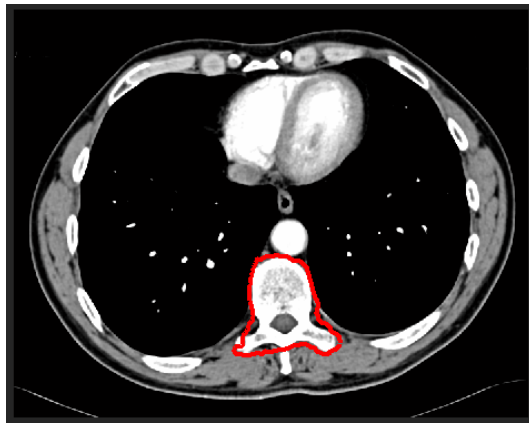
Prueba 1005-Contorno inicial



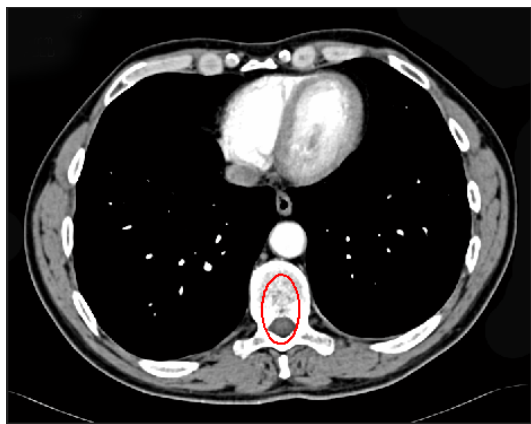
Prueba 1005-Contorno final



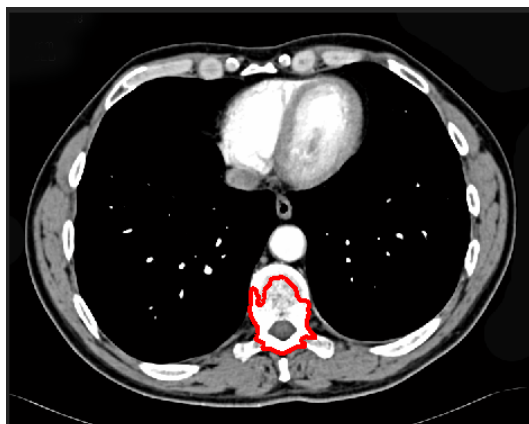
Prueba 1006-Contorno inicial



Prueba 1006-Contorno final



Prueba 1033-Contorno inicial



Prueba 1033-Contorno final

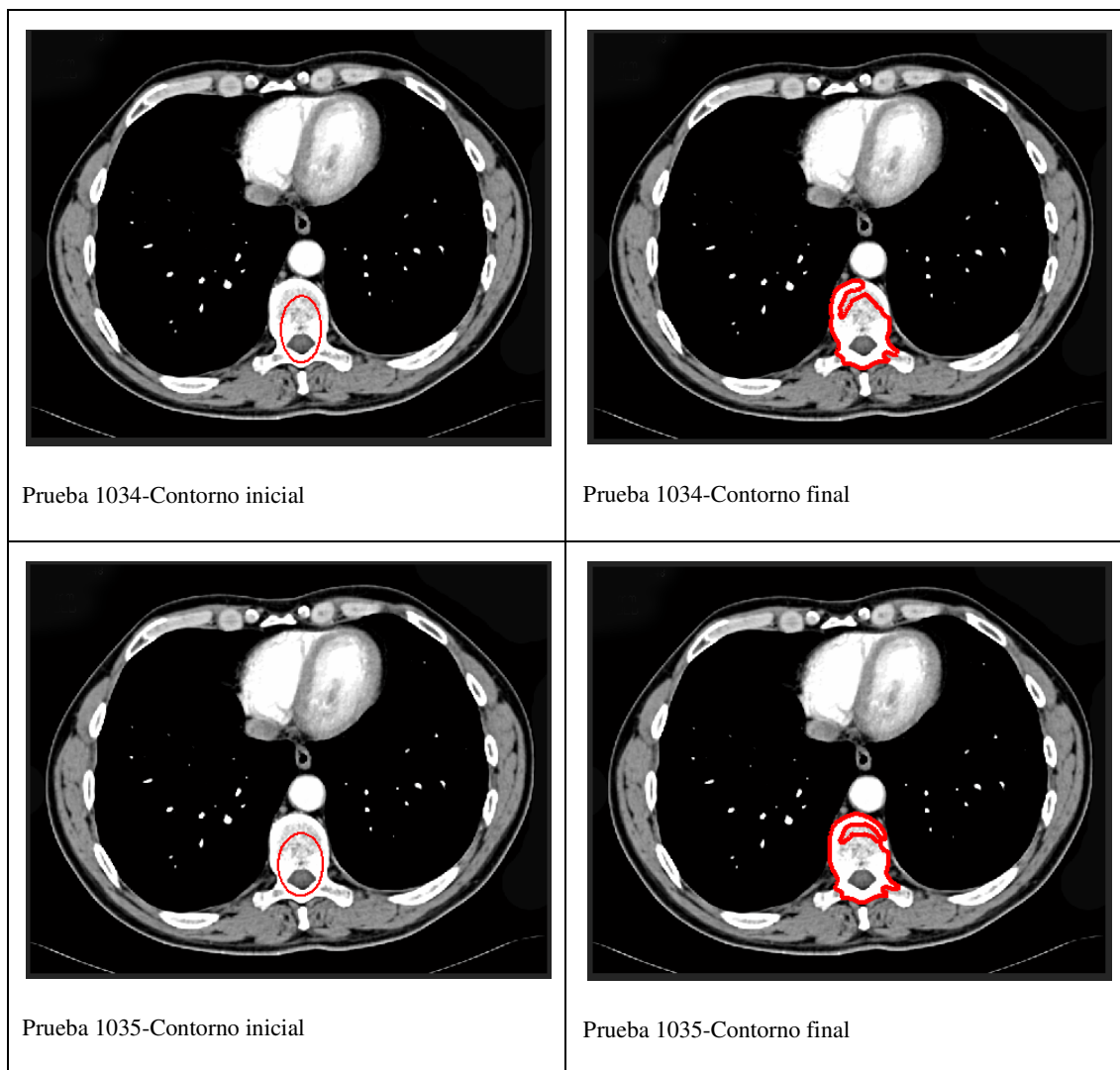


Figura 3.1. Pruebas realizadas con el método de Caselles para la imagen 1

Estudiamos algunos ejemplos de experimentos realizados con el algoritmo de Caselles para la imagen 2:

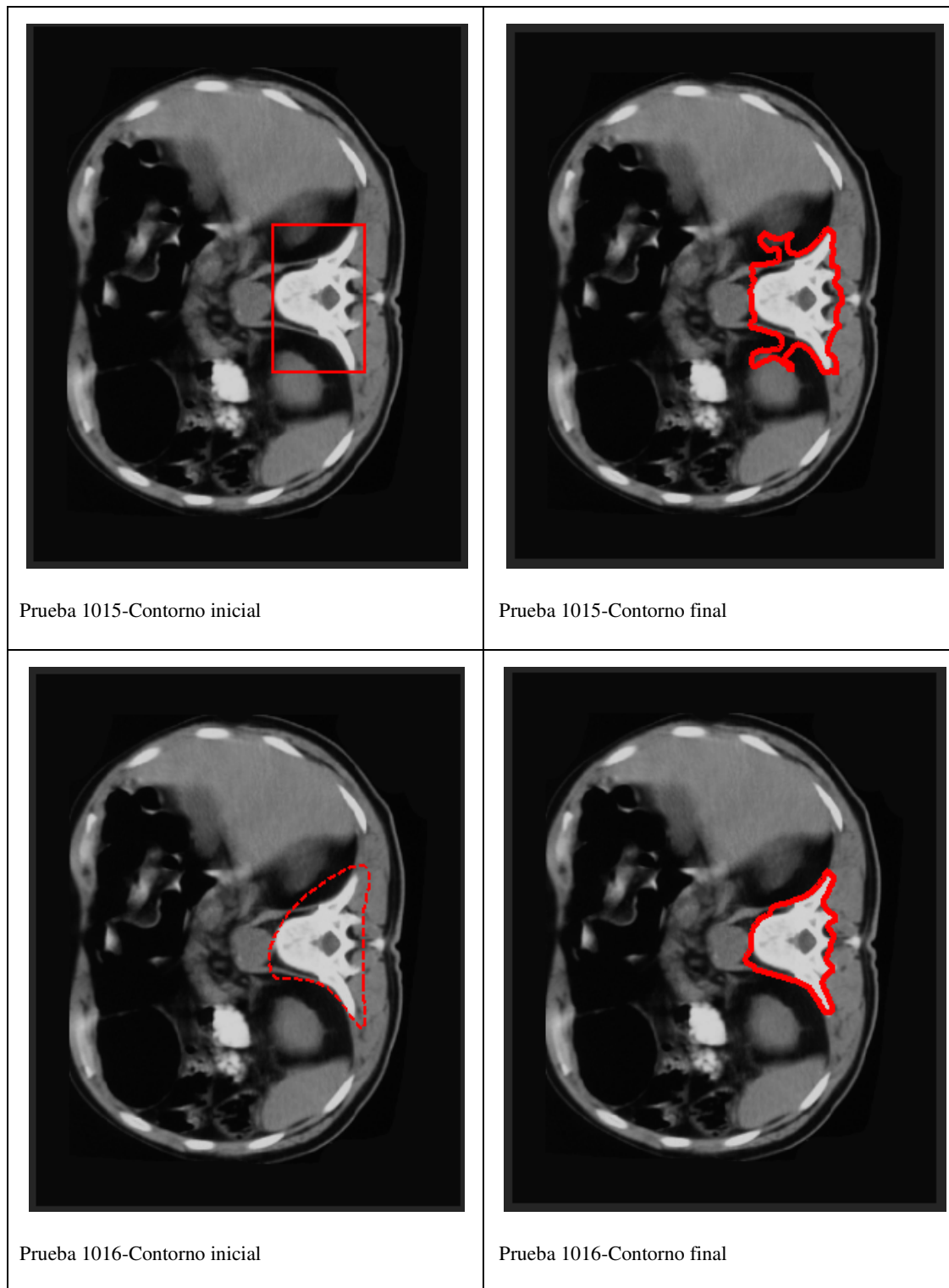


Figura 3.2. Pruebas realizadas con el método de Caselles para la imagen 2

Consideremos algunos ejemplos de ensayos realizados con el algoritmo de Caselles para la imagen 3:

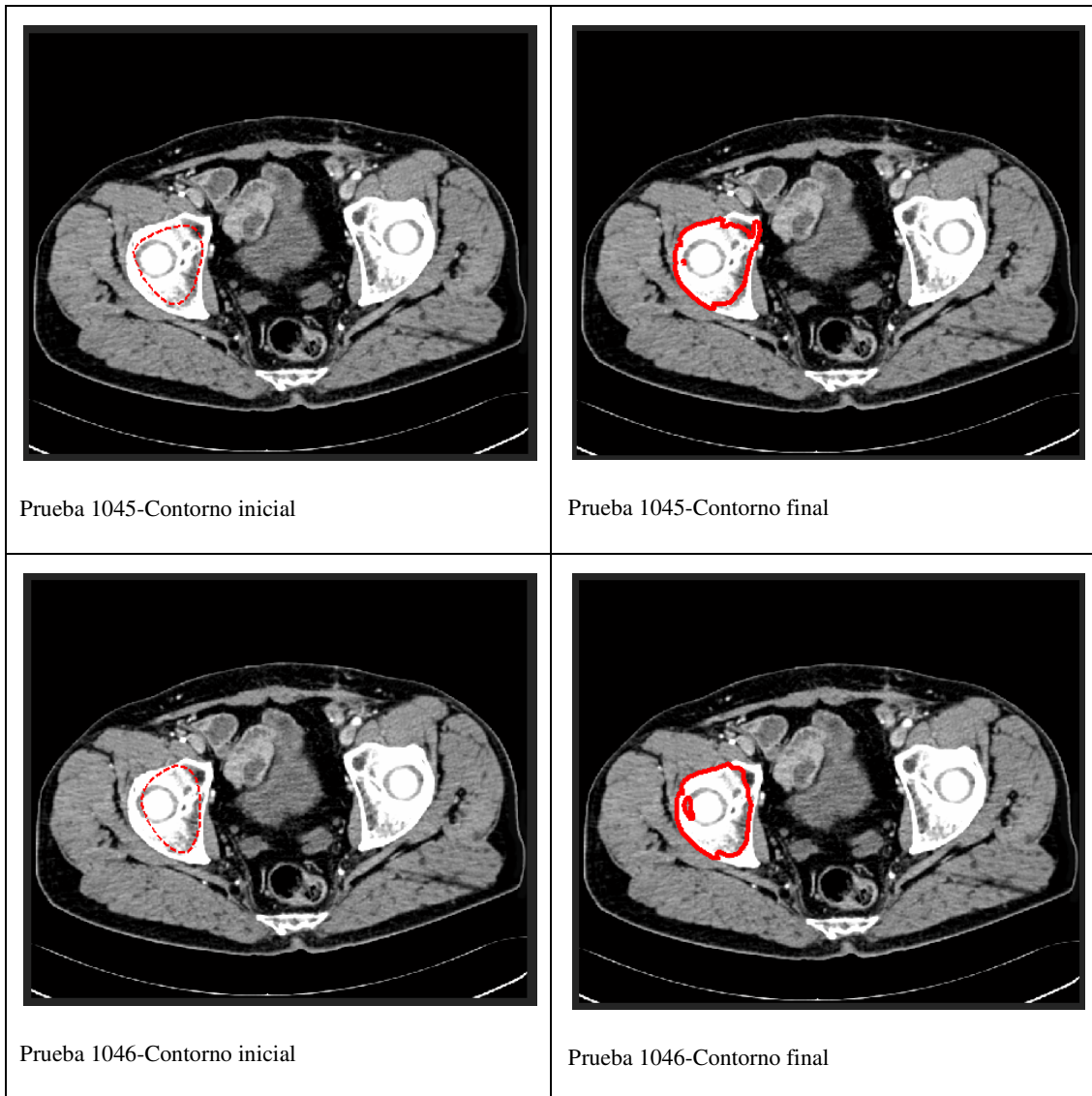


Figura 3.3. Resultados de pruebas realizadas con el método de Caselles para la imagen 3

Observemos algunos ejemplos de experimentos realizados con el algoritmo de Caselles para la imagen 7:

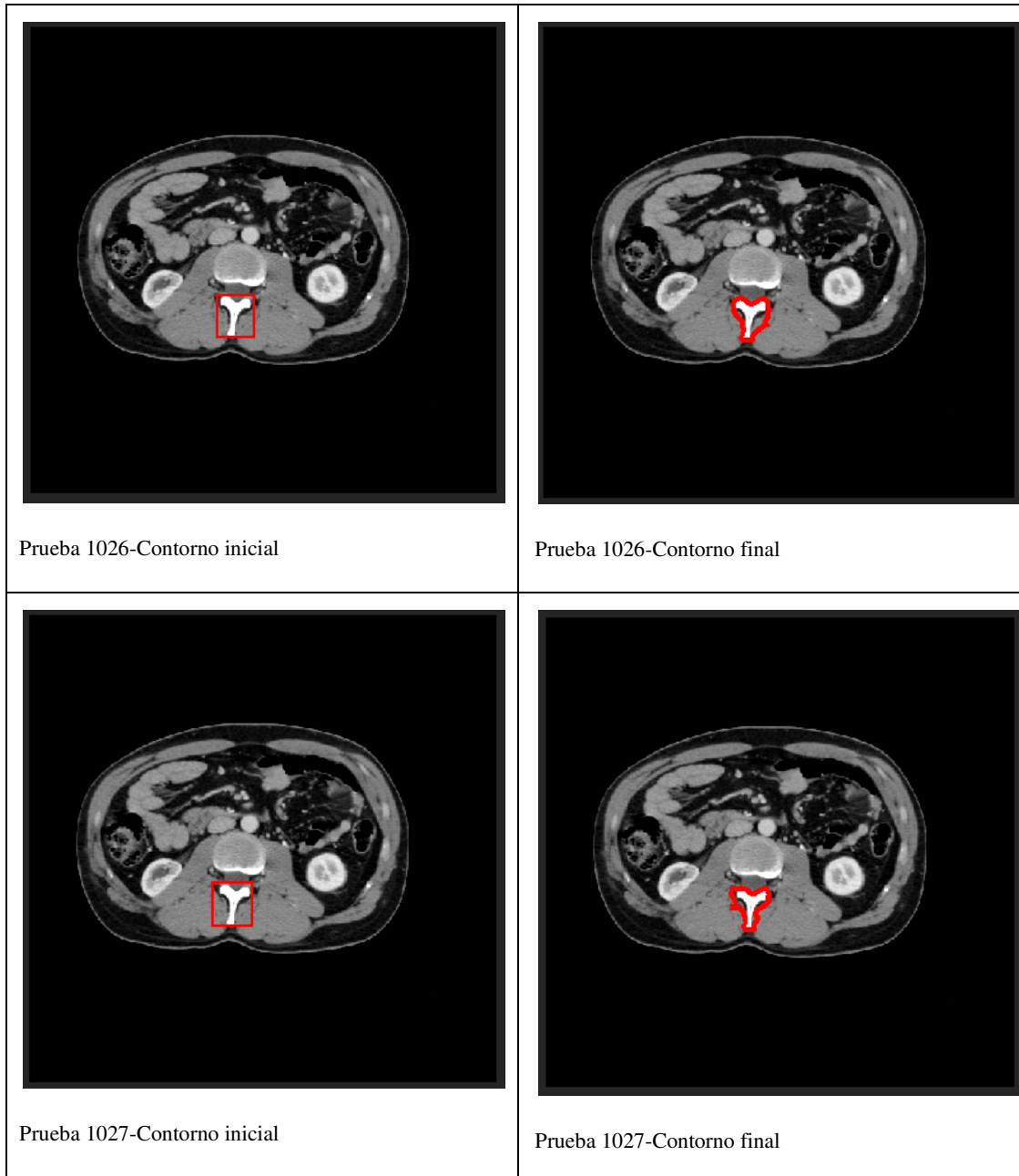


Figura 3.4. Pruebas realizadas con el método de Caselles para la imagen 7

Veamos ahora los resultados de ejecutar el algoritmo sobre otro tipo de imagen (Test-BN) donde se ha preprocesado la imagen hasta el punto de dejar sólo el objeto de interés:

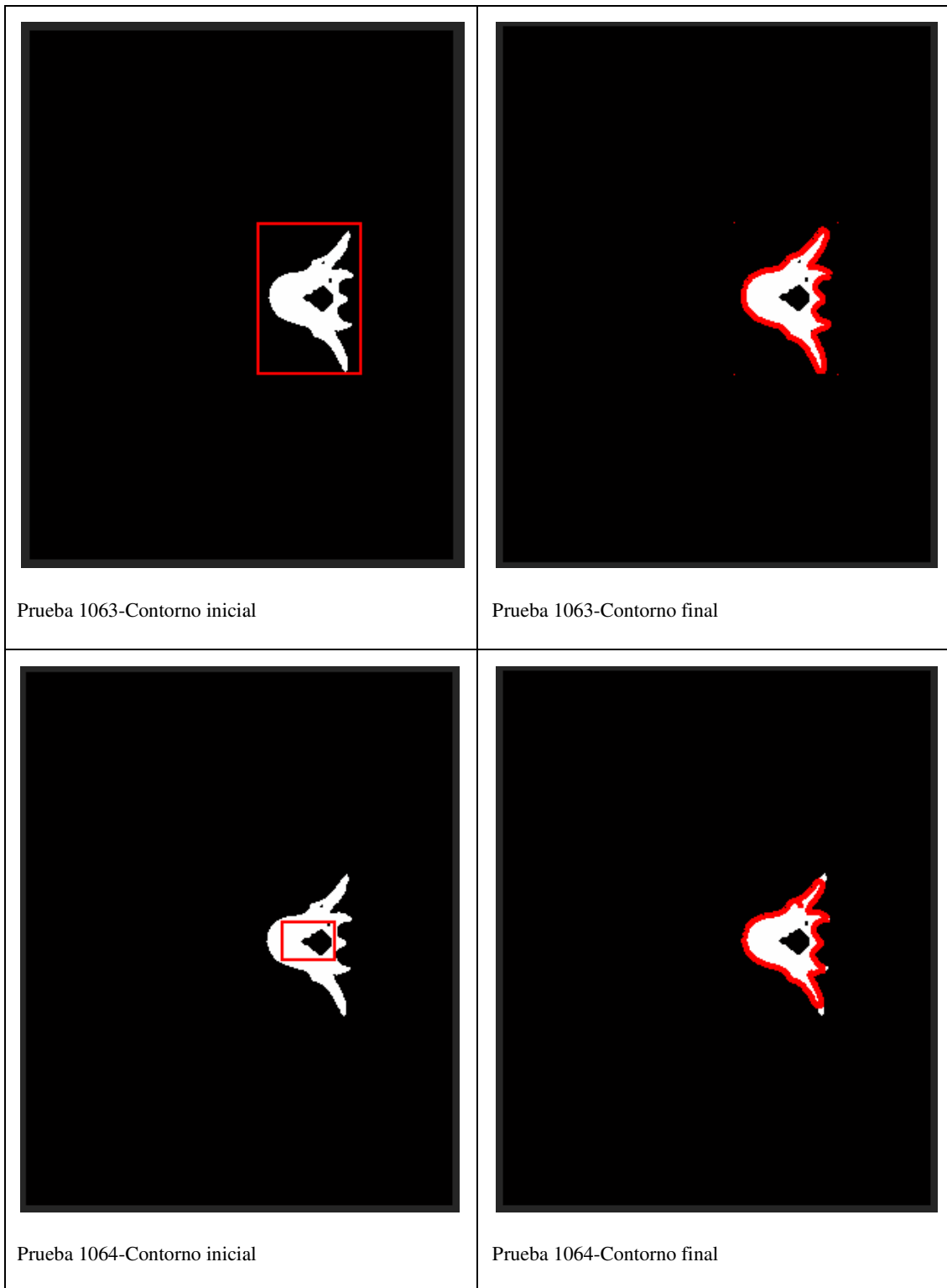


Figura 3.5. Ensayos realizados con el método de Caselles para la imagen Test-BN

A la vista de los experimentos realizados con el algoritmo de Caselles[CASELLES97] podemos concluir:

-en la mayoría de los experimentos llevados a cabo el contorno final no se acercaba mucho al contorno ideal.

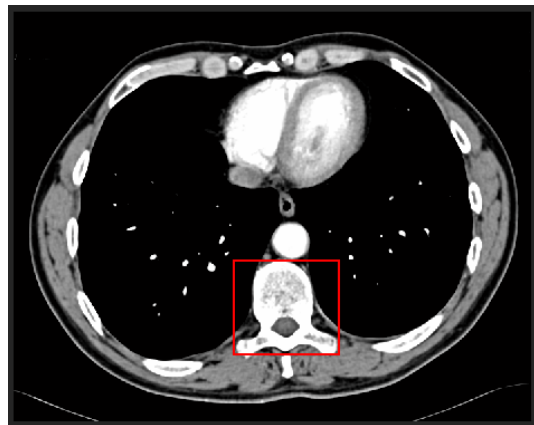
-este algoritmo está preparado tanto para funcionar desde fuera hacia adentro, como desde dentro hacia afuera, para esto, basta con configurar su término de propagación a -1 cuando queramos que evolucione hacia adentro o a +1 cuando queramos que evolucione hacia afuera. El algoritmo funciona mejor estableciendo el contorno inicial fuera del objeto de interés. Cuando inicializamos el contorno dentro del objeto de interés se comporta peor y los errores son mucho mayores.

-se han obtenido mejores resultados cuando se ha ejecutado sobre una imagen binaria, aunque en el caso de inicializar el contorno dentro del objeto los resultados tampoco han sido del todo satisfactorios.

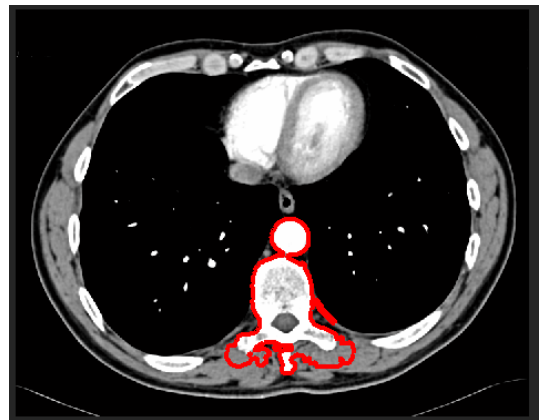
-en otros experimentos realizados en imágenes con ruido se ha observado que empeoran los resultados y, si no tiene mucho ruido, a veces llega a acercarse al contorno buscado, pero necesita muchas más iteraciones para poder acercarse al contorno óptimo.

3.2.1.2. Problemas encontrados en algoritmo de Chan

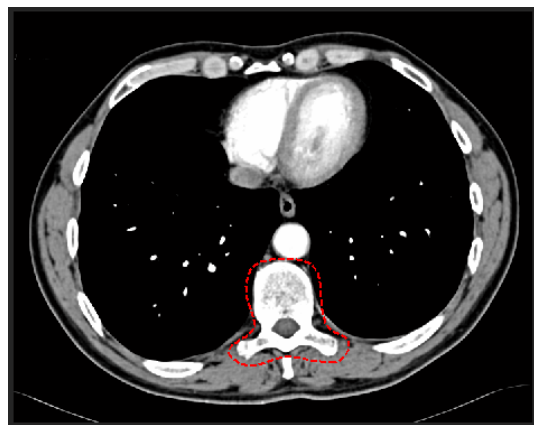
Examinemos en primer lugar algunos ejemplos de experimentos realizados con el algoritmo de Chan [CHAN01] y posteriormente extraeremos algunas conclusiones. Empezamos con algunos ensayos efectuados para la imagen 1:



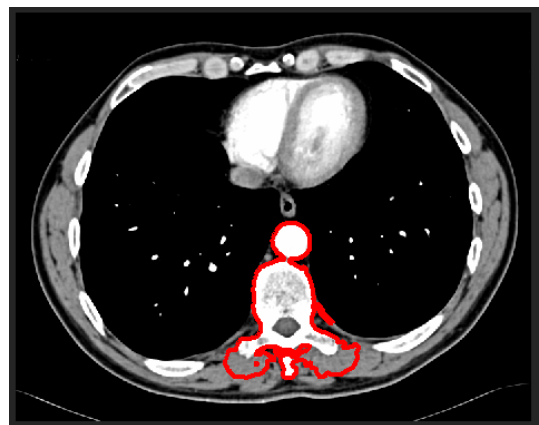
Prueba 1007-Contorno inicial



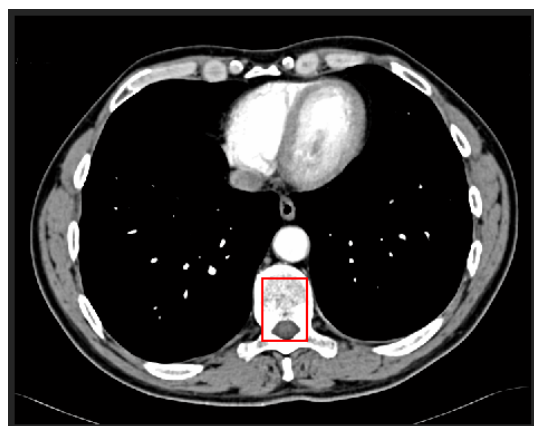
Prueba 1007-Contorno final



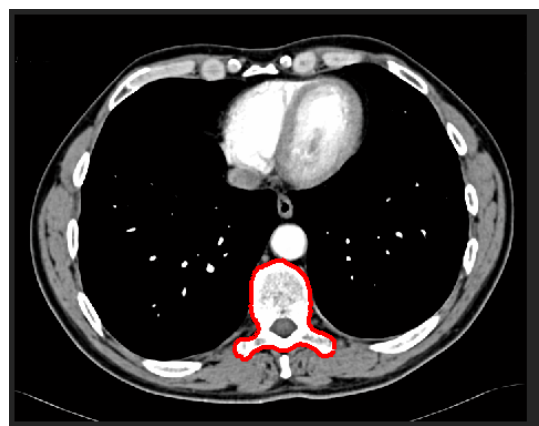
Prueba 1008-Contorno inicial



Prueba 1008-Contorno final



Prueba 1009-Contorno inicial



Prueba 1009-Contorno final

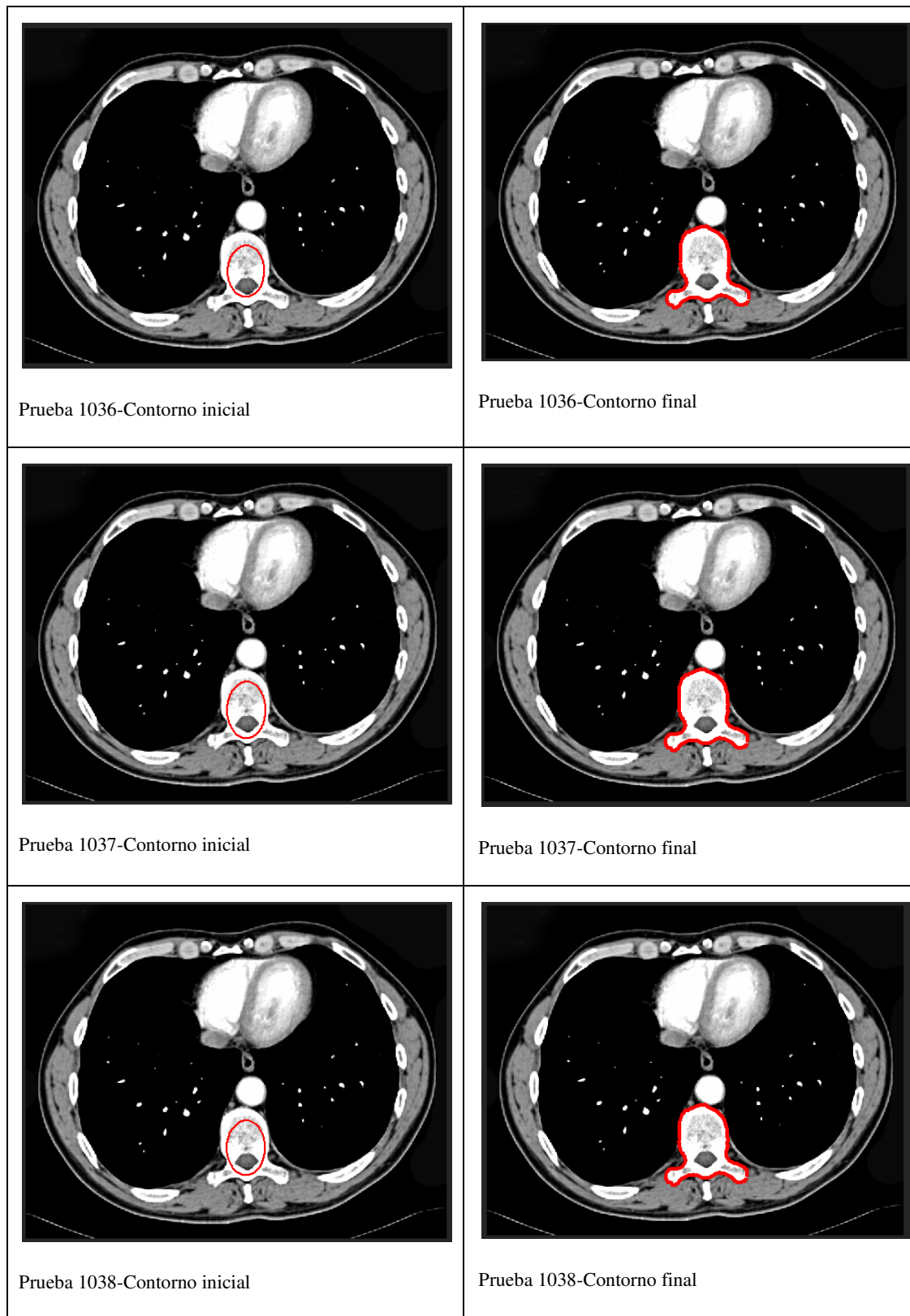


Figura 3.6. Pruebas realizadas con el método de Chan para la imagen 1

Veamos algunos ejemplos de experimentos ejecutados con el método de Chan para la imagen 2:

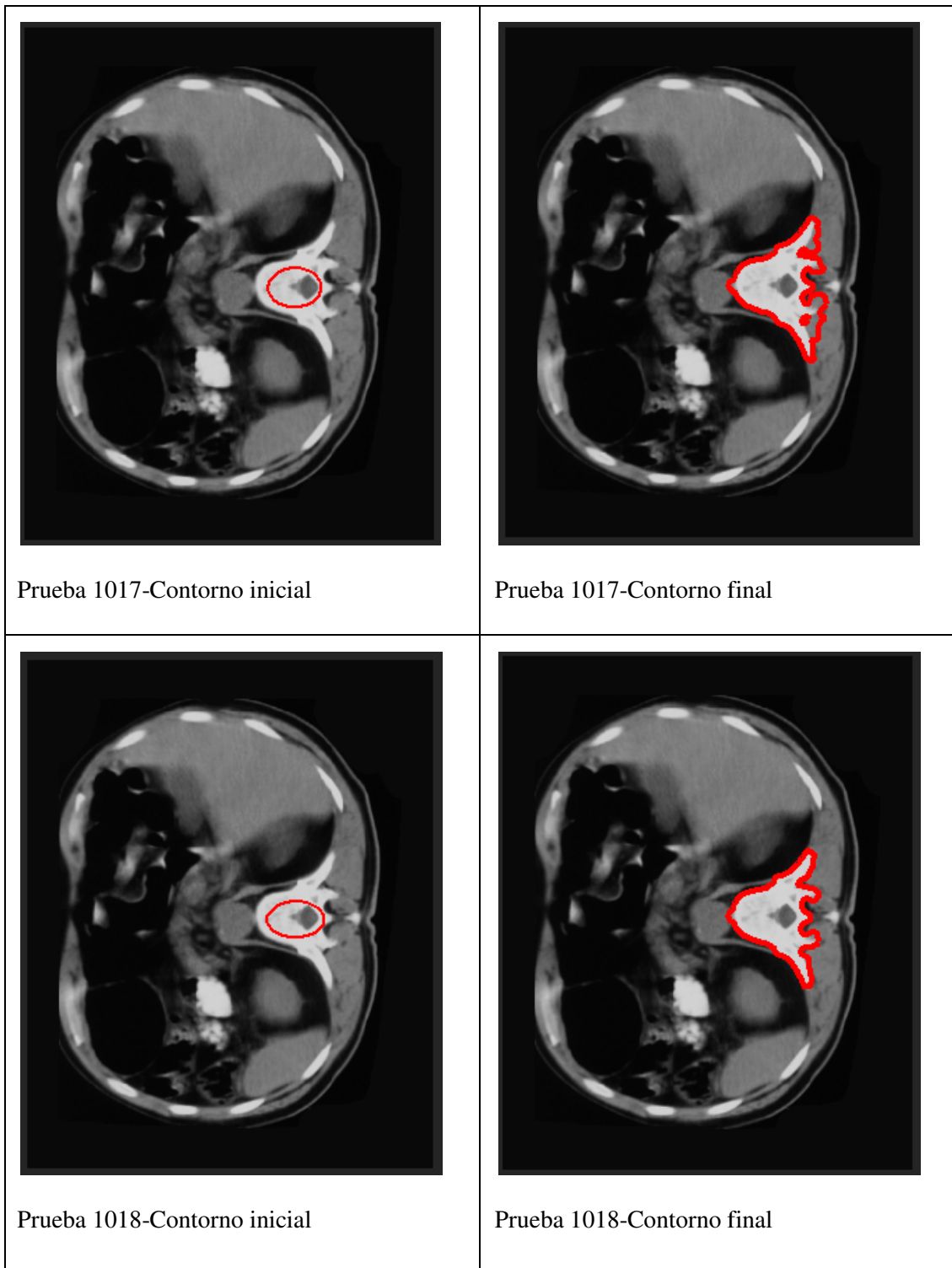
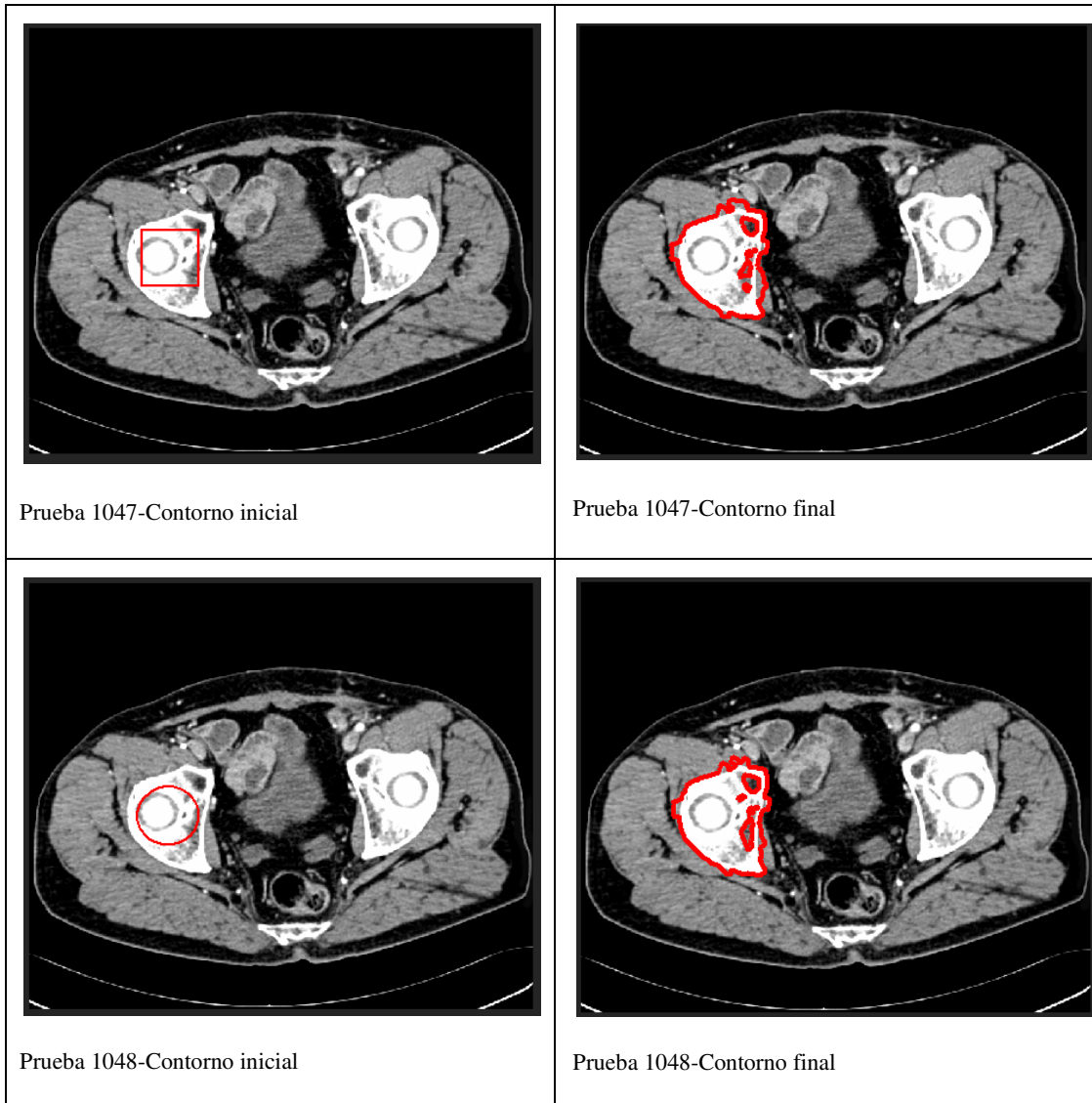


Figura 3.7. Pruebas realizadas con el método de Chan para la imagen 2

Estudiamos algunos ejemplos de experimentos efectuados con el método de Chan para la imagen 3:



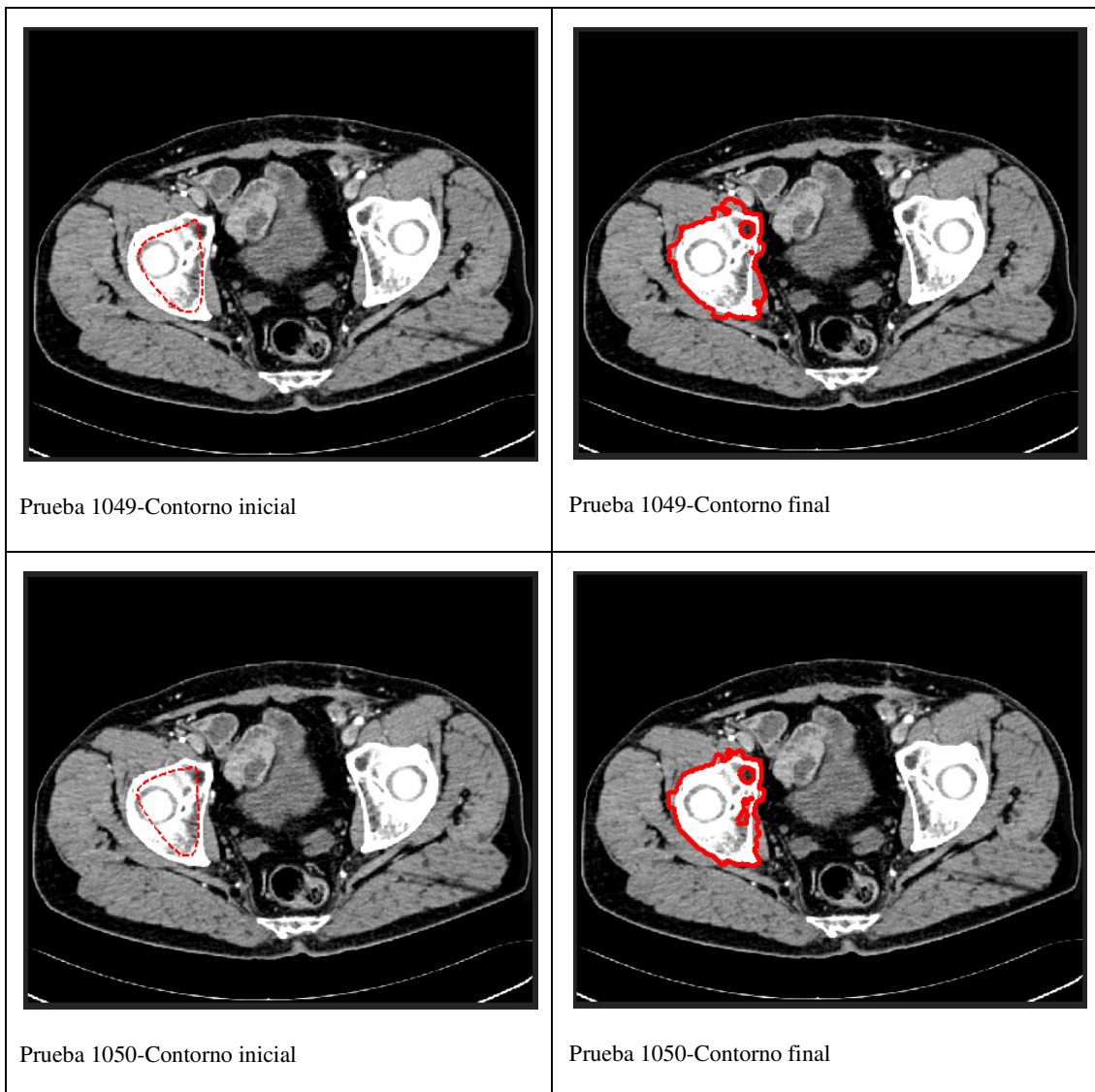
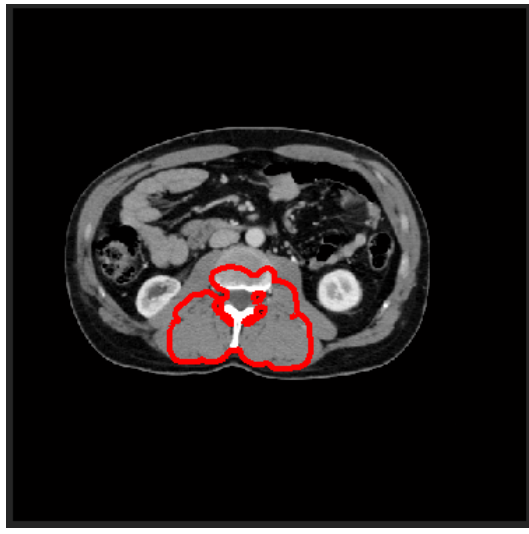


Figura 3.8. Pruebas realizadas con el método de Chan para la imagen 3

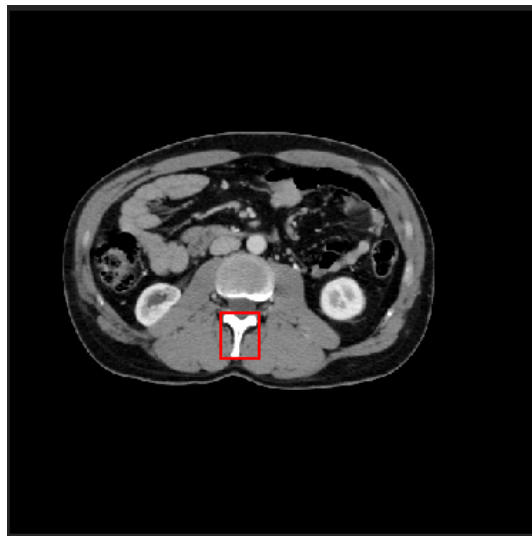
Observemos varios experimentos ejecutados con el método de Chan para la imagen 7:



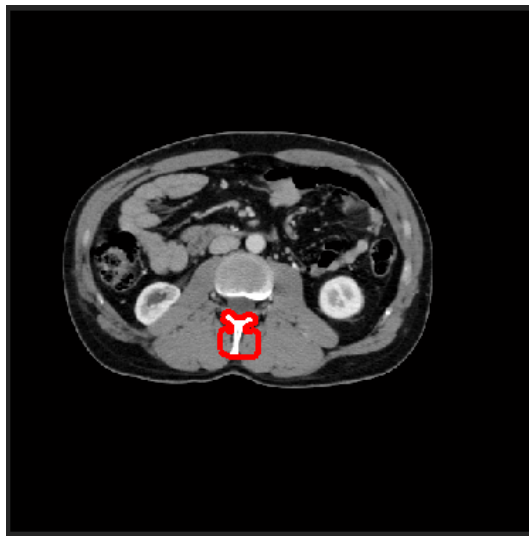
Prueba 1028-Contorno inicial



Prueba 1028-Contorno final



Prueba 1029-Contorno inicial



Prueba 1029-Contorno final

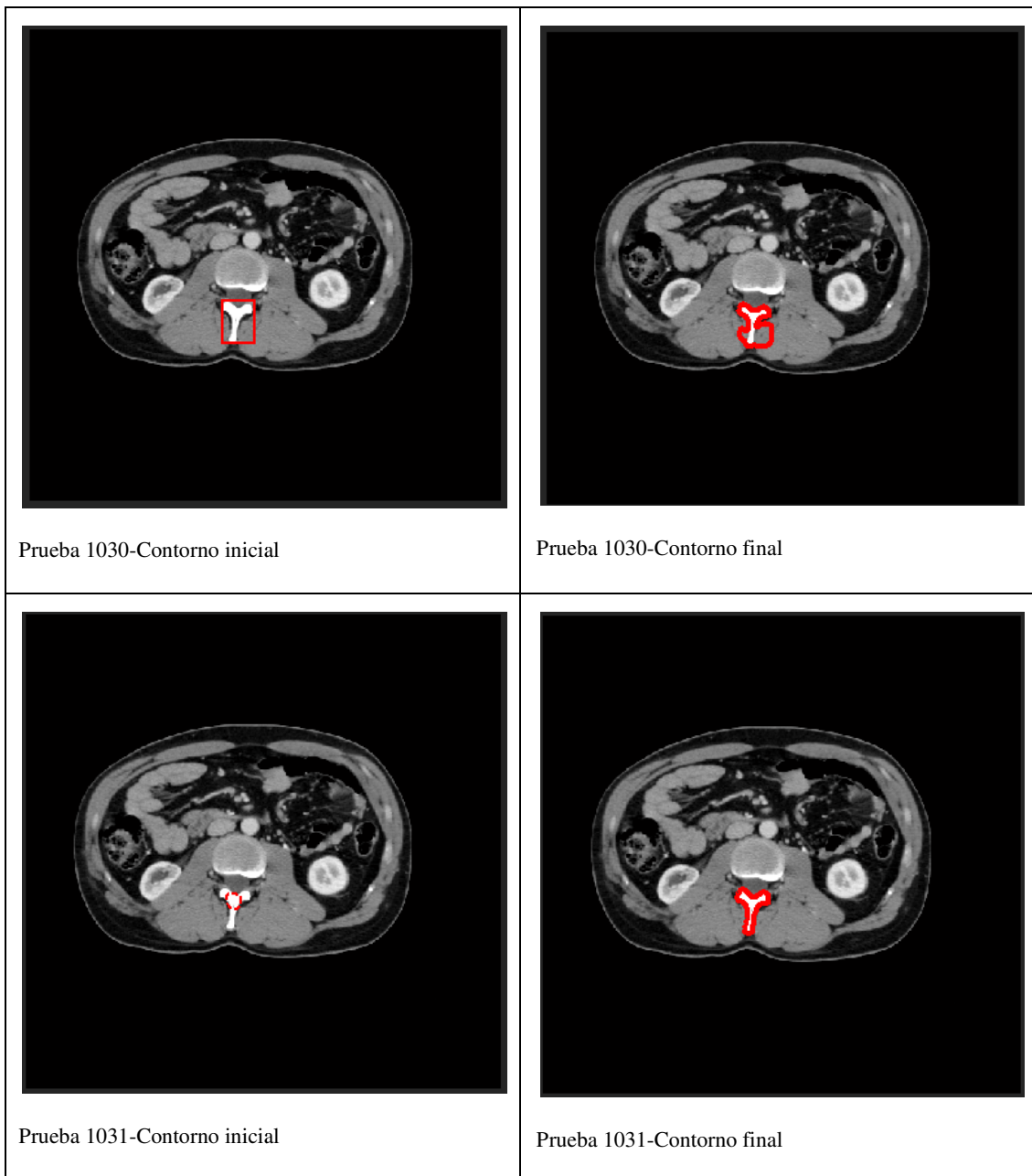


Figura 3.9. Pruebas realizadas con el método de Chan para la imagen 7

En este caso, el mejor resultado se obtiene iniciando el contorno dentro del objeto de interés.

Analicemos ahora los resultados de ejecutar el método de Chan sobre una imagen (Test-BN) donde sólo hay un objeto blanco sobre fondo negro:

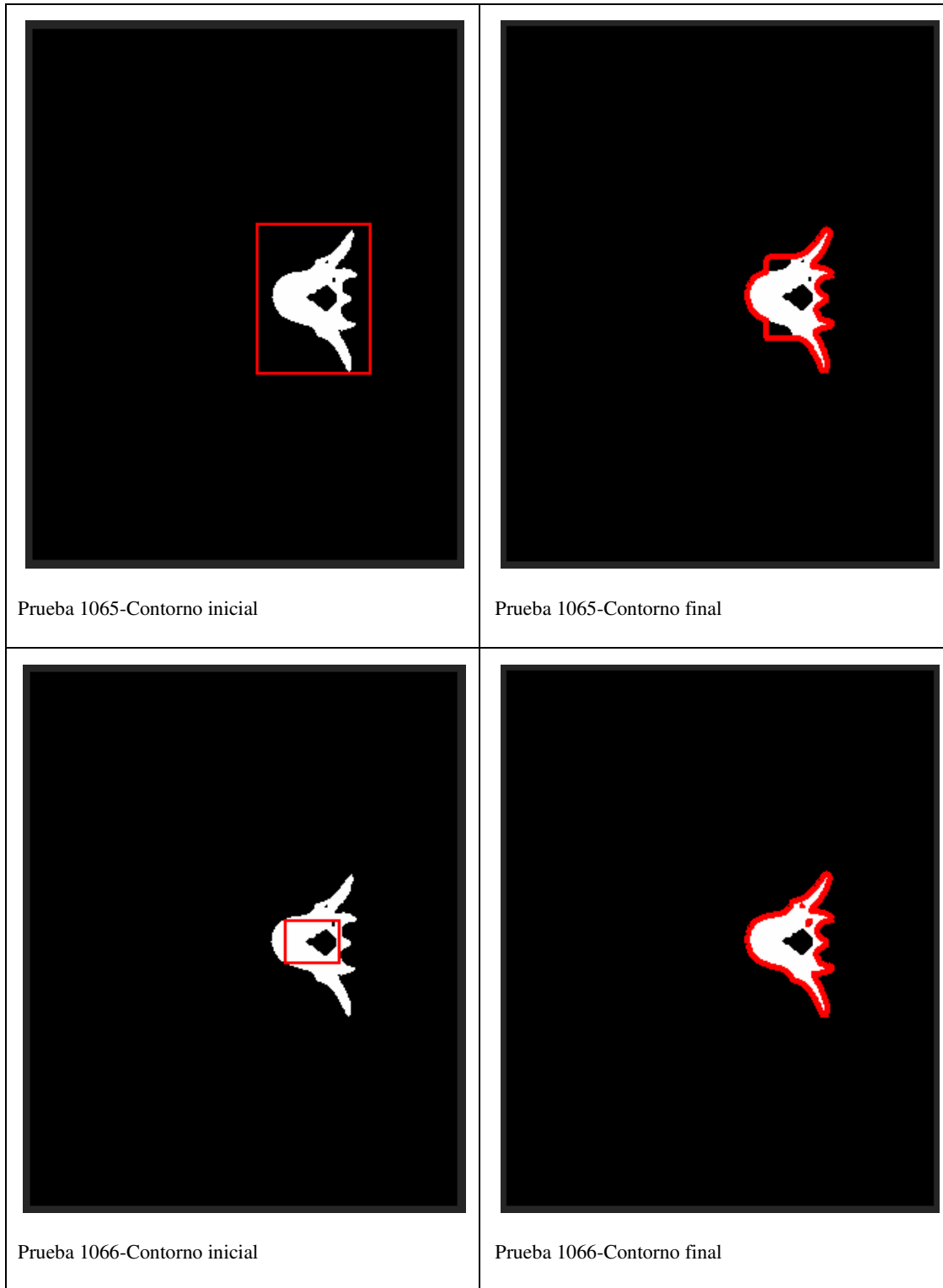


Figura 3.10. Pruebas realizadas con el método de Chan para la imagen Test-BN

Tras realizar estos experimentos con el método de Chan [CHAN01], podemos extraer las siguientes conclusiones:

-si inicializamos el contorno fuera del objeto de interés, los resultados no suelen ser buenos.

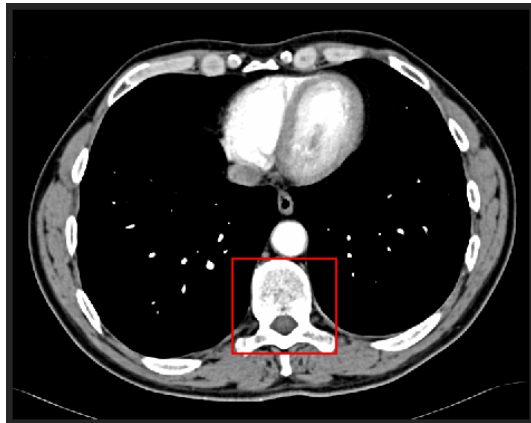
-los resultados mejoraron considerablemente inicializando el contorno dentro del objeto de interés y probando con diferentes valores del término de curvatura. El término de curvatura establece el peso del término de regularización en la ecuación de evolución. Se efectuaron ensayos con valores del término de curvatura entre 0,05 y 0,7, obteniéndose los mejores resultados visuales para 0,05.

-en algunos casos (pruebas 1017 y 1018) se observó que si se configuraba el método con un número de iteraciones demasiado alto, el resultado era peor que el mismo experimento con menos iteraciones, pues después de aproximarse mucho al contorno óptimo, el algoritmo seguía evolucionando y traspasaba la frontera del objeto, llegando a un resultado peor que con menos iteraciones. Esto hace que sea difícil utilizar este algoritmo de forma automática, pues necesita de bastantes ensayos previos para saber que configuración necesita para ese caso concreto.

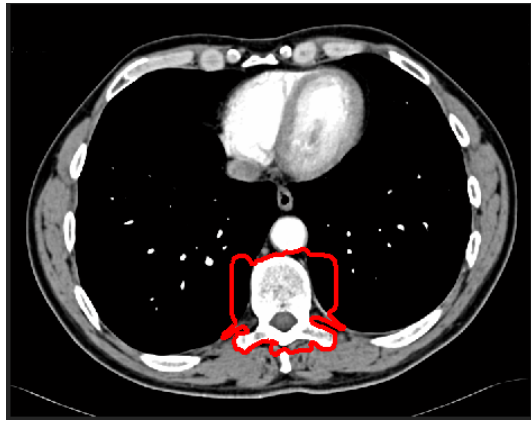
-este método de Chan es el método antecedente con el que mejores resultados hemos conseguido, junto con los métodos desarrollados para esta tesis, como se expondrá más detalladamente en el capítulo 4 de *Resultados*, donde se muestra una comparativa de todos estos métodos con los desarrollados en esta tesis.

3.2.1.3. Problemas encontrados en algoritmo de Lankton

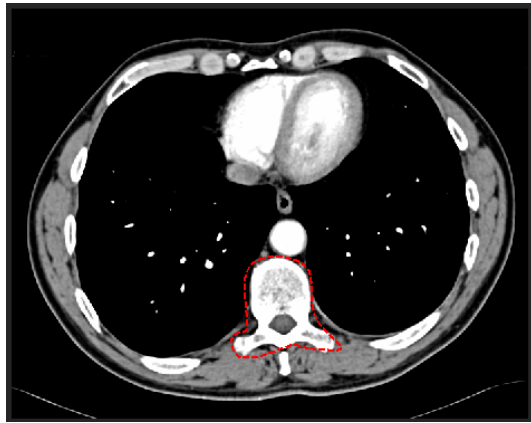
Estudiemos primero varios experimentos realizados con el algoritmo de Lankton [LANKTON08] y posteriormente extraeremos algunas conclusiones. Empezamos con algunas pruebas realizadas para la imagen 1:



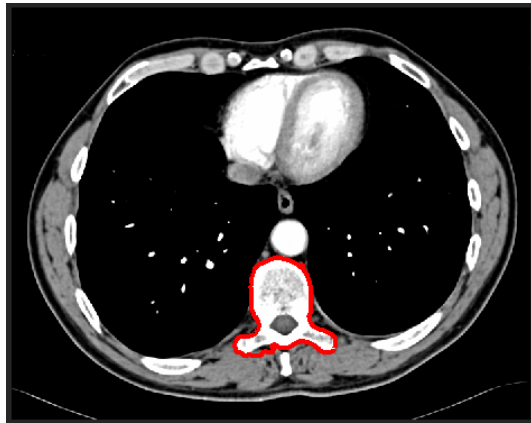
Prueba 1010-Contorno inicial



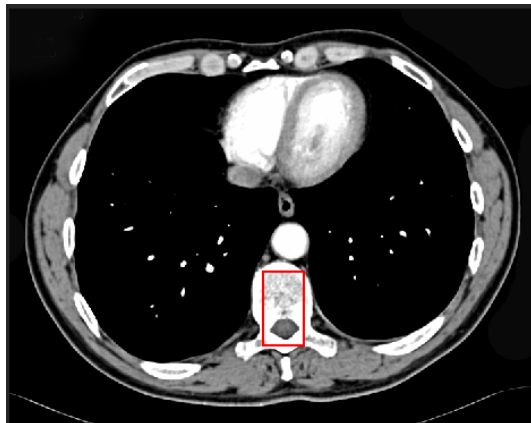
Prueba 1010-Contorno final



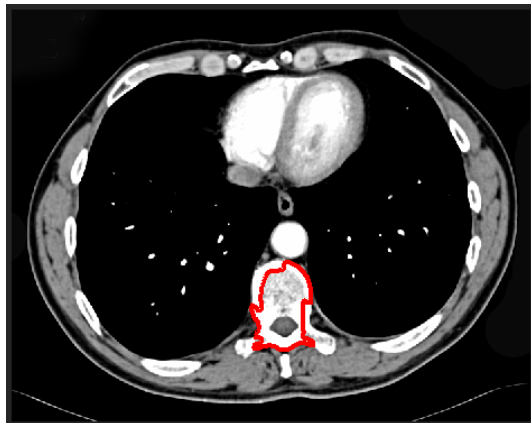
Prueba 1011-Contorno inicial



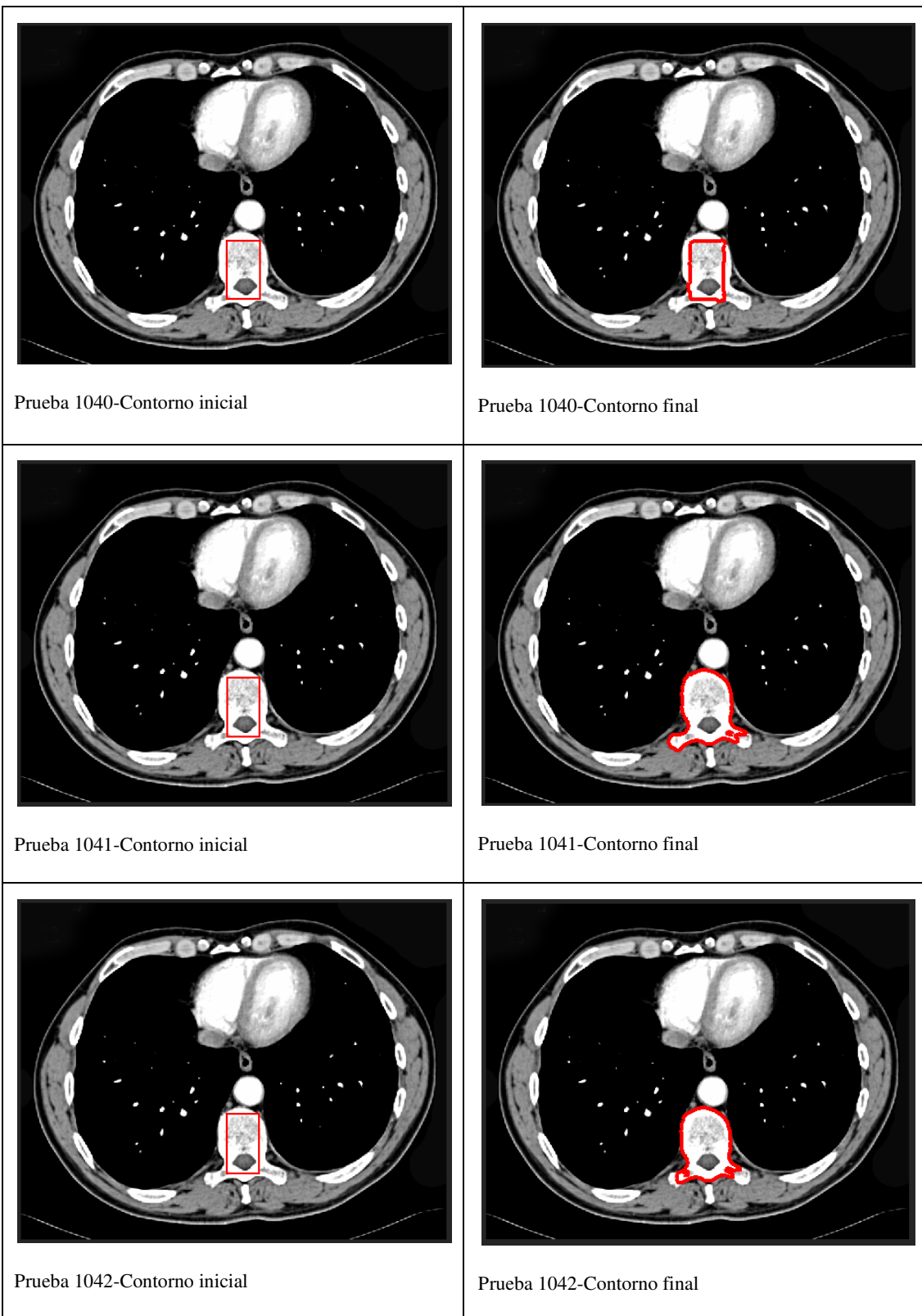
Prueba 1011-Contorno final



Prueba 1039-Contorno inicial



Prueba 1039-Contorno final



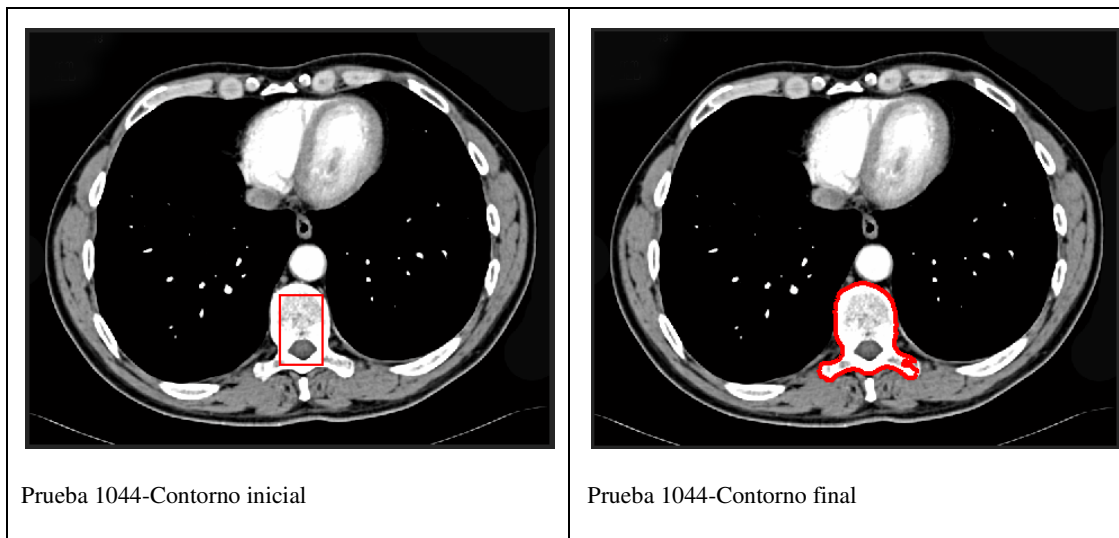
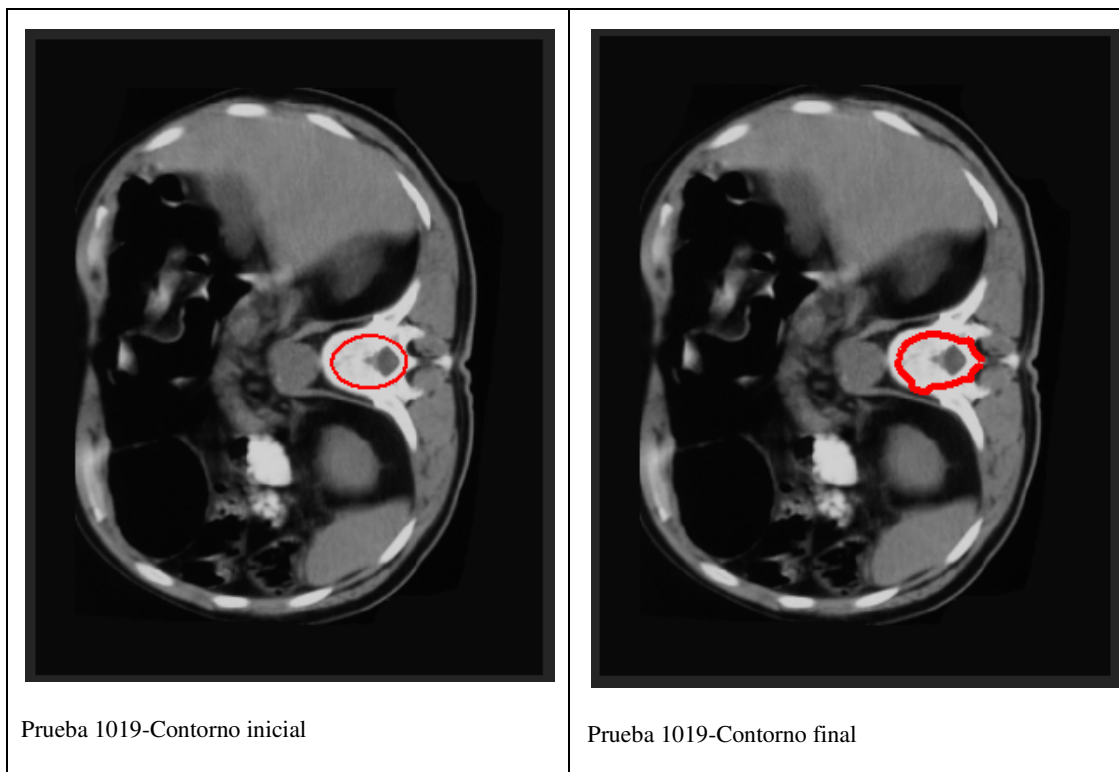


Figura 3.11. Pruebas realizadas con el método de Lankton para la imagen 1

Consideremos algunos ejemplos de pruebas realizadas con el método de Lankton para la imagen 2:



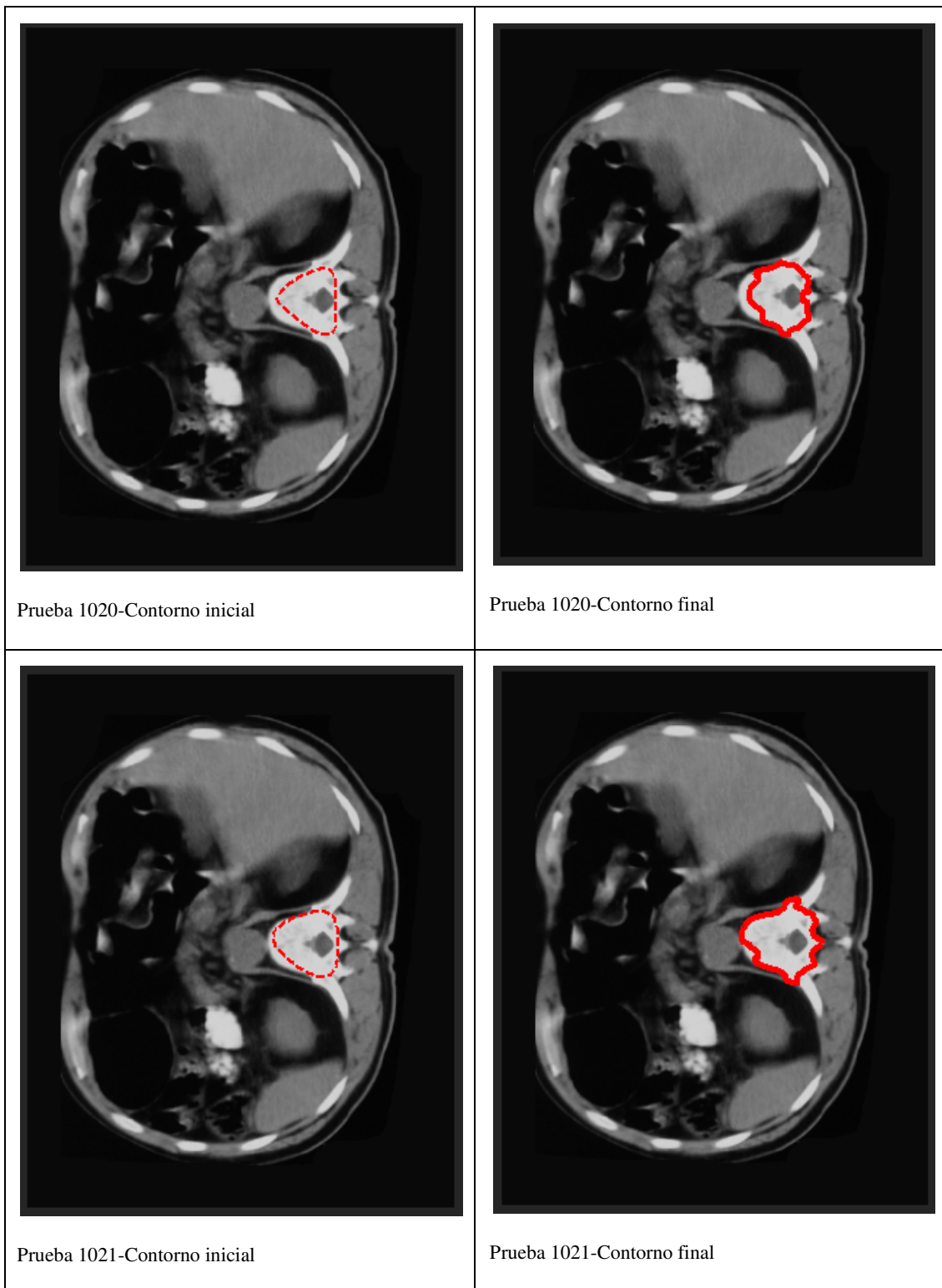
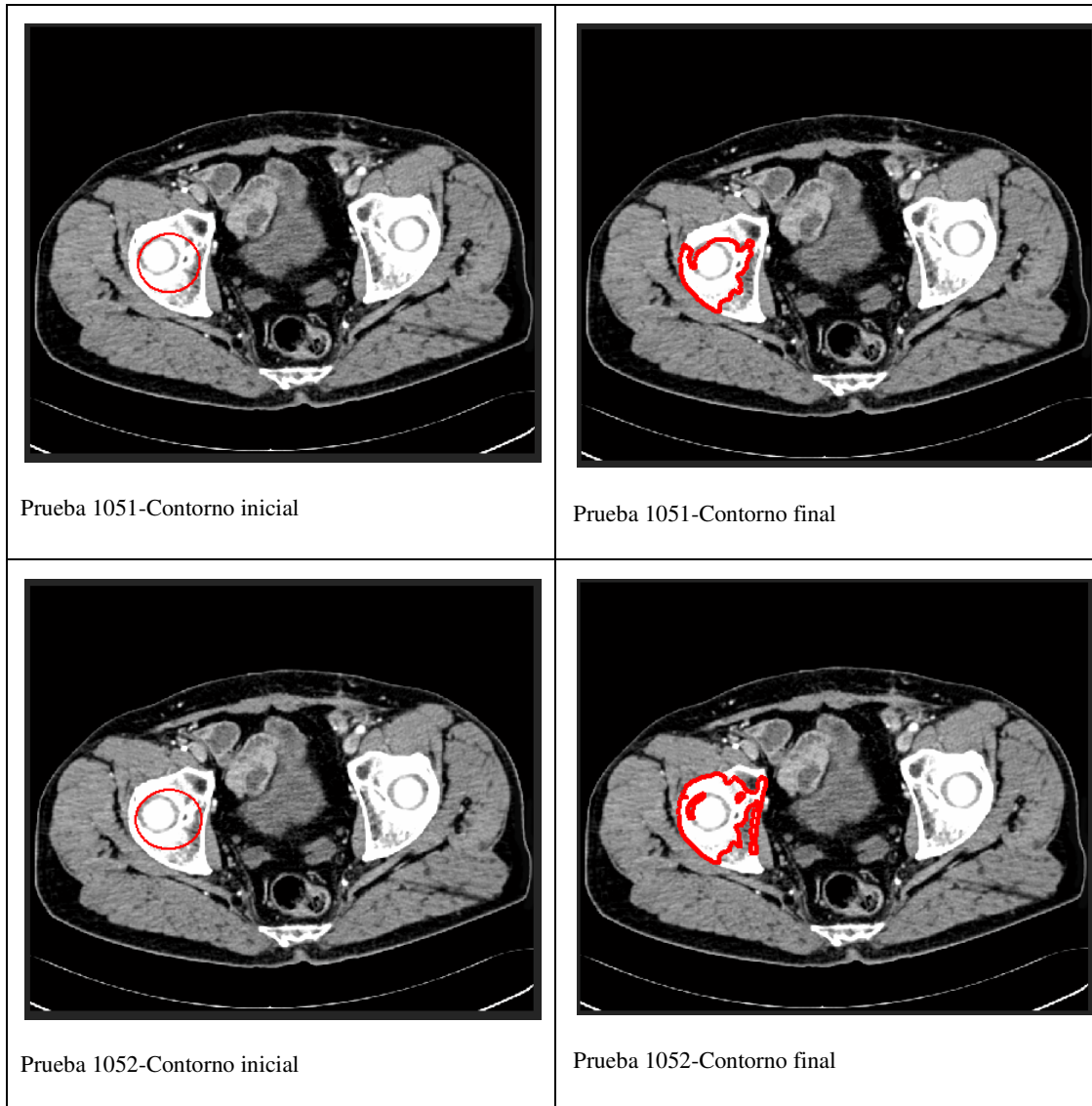


Figura 3.12. Experimentos realizados con el método de Lankton para la imagen 2

Conozcamos varios casos de ensayos realizados con el método de Lankton para la imagen 3:



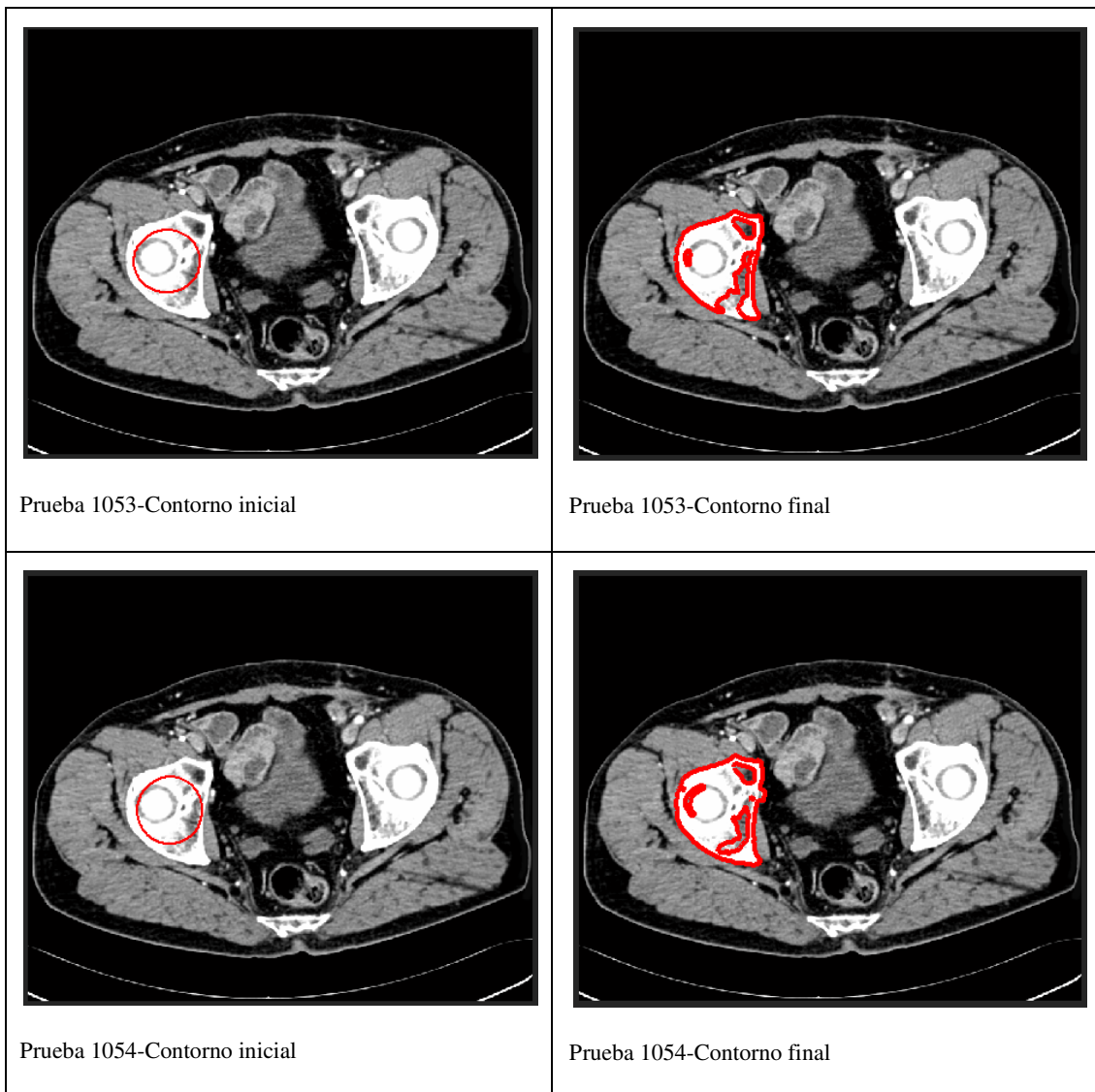


Figura 3.13. Pruebas realizadas con el método de Lankton para la imagen 3

Analicemos algunos casos de experimentos efectuados con el método de Lankton para la imagen 7:

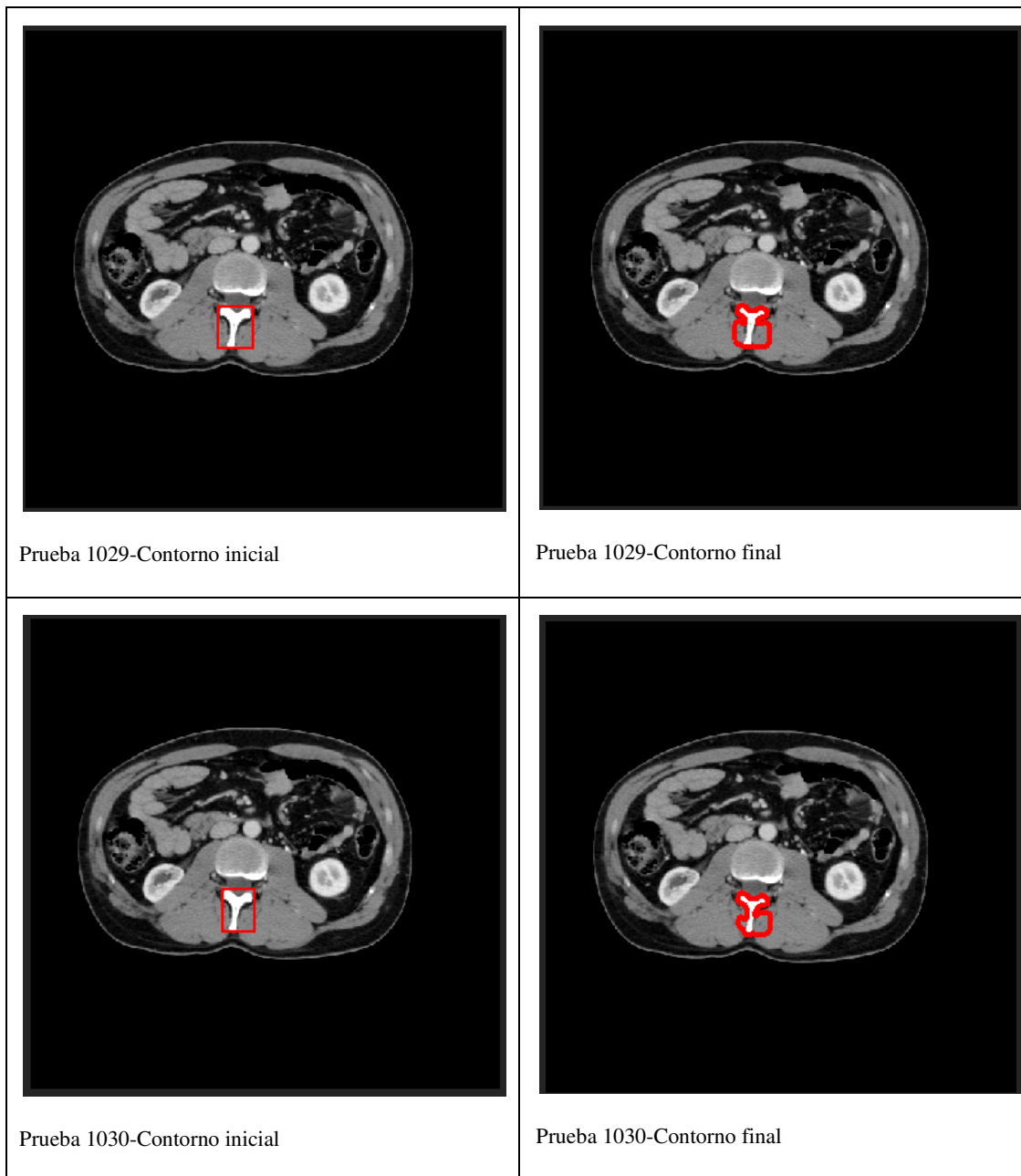


Figura 3.14. Pruebas realizadas con el método de Lankton para la imagen 7

Observemos ahora algunas pruebas realizadas con el método de Lankton para la imagen Test BN:

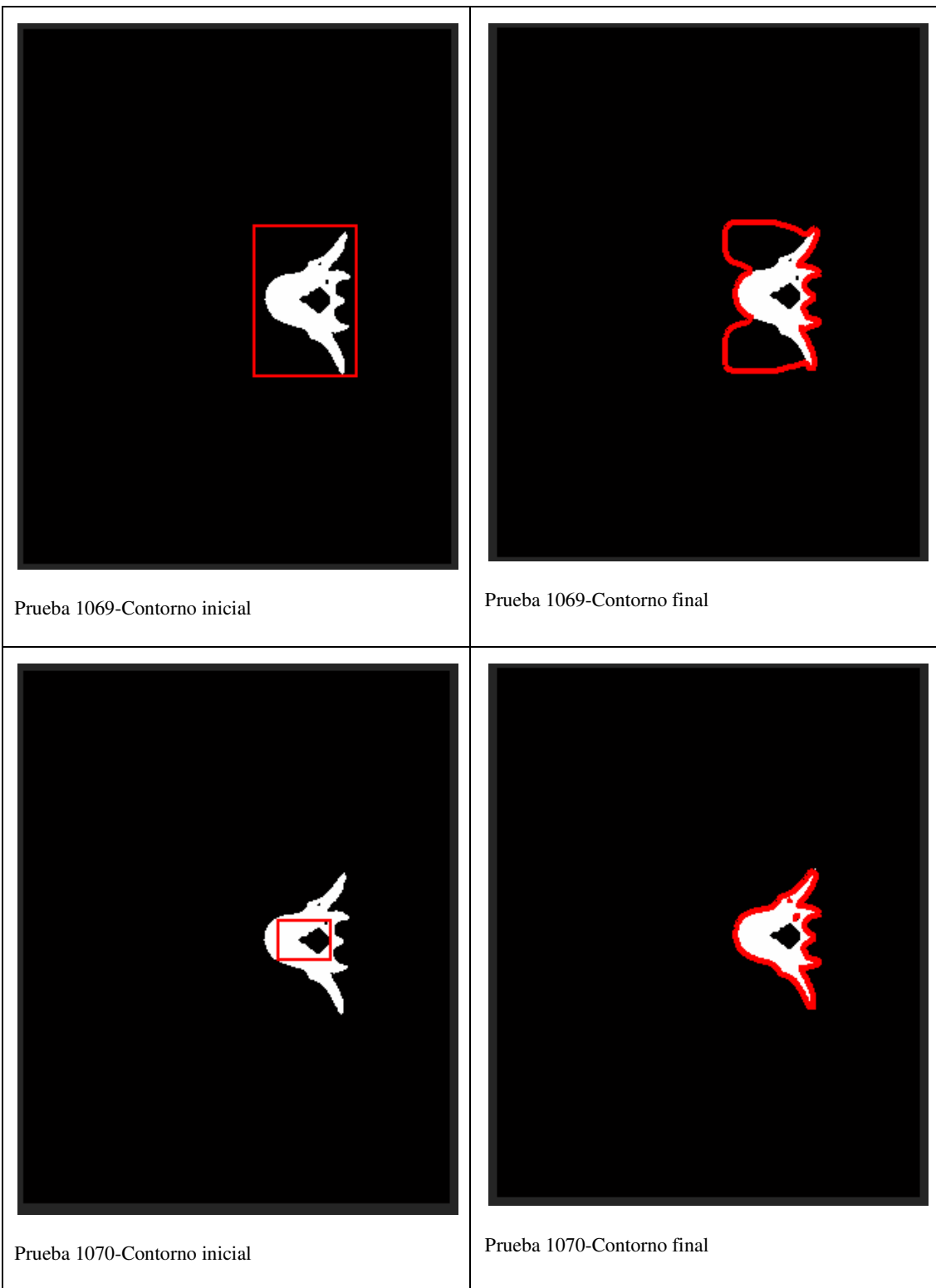




Figura 3.15. Ensayos realizados con el método de Lankton para la imagen Test BN

Tras realizar experimentos con el método de Lankton[LANKTON08], podemos extraer varias conclusiones:

-se ha visto que es más sensible a la inicialización que otros métodos. En concreto, influye en mayor medida si el contorno inicial está cerca o lejos del contorno buscado.

-en general, los resultados obtenidos con el método de Lankton han sido peores que con otros métodos, aunque hemos conseguido buenos resultados en las imágenes 1 y 3, inicializando el contorno dentro del objeto y probando con distintos valores de los parámetros. En general, este método tiene un mayor nivel de complejidad a la hora de configurar sus parámetros y de encontrar unos valores adecuados para obtener buenos resultados. El parámetro que tiene mayor influencia en el resultado final es el término del radio y hemos observado que los resultados mejoran cuando se aumenta el tamaño de la vecindad (estableciendo el término del radio con valores más altos).

-en algunas pruebas (por ejemplo en la 1054) se produjeron cambios topológicos en el contorno activo, se dividieron en varios contornos y se detectaron finalmente varios contornos, la mayoría de ellos con poca precisión.

3.2.1.4. Problemas encontrados en algoritmo de Bernard

Observemos en primer lugar algunos experimentos realizados con el algoritmo de Bernard [BERNARD09] y posteriormente extraeremos algunas conclusiones. Empecemos observando los resultados alcanzados con las pruebas efectuadas utilizando el método de Bernard para la imagen 1:

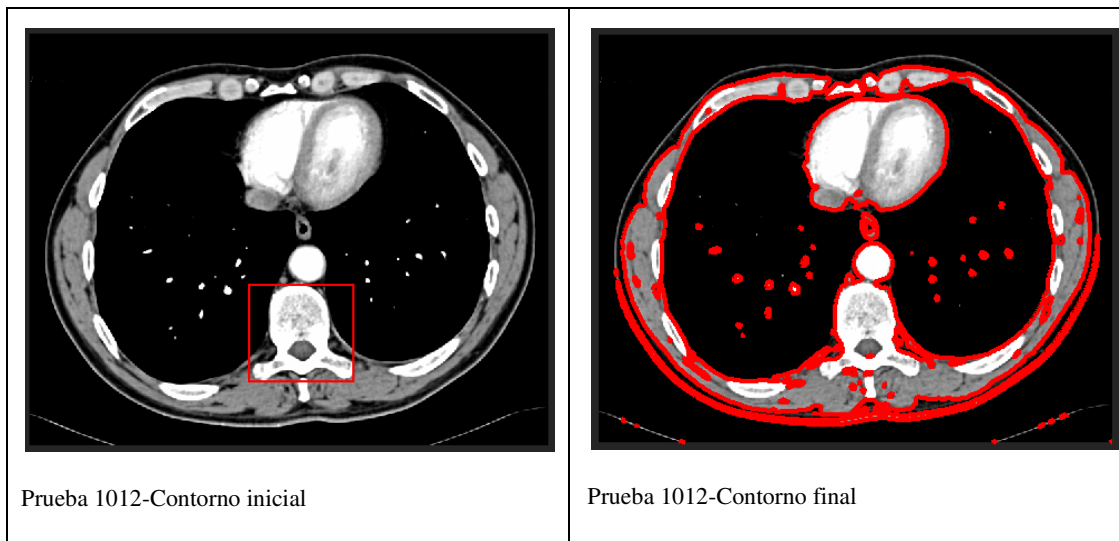


Figura 3.16. Prueba realizada con el método de Bernard para la imagen 1

Observemos algunos ejemplos de ensayos efectuados con el método de Bernard para la imagen 2:

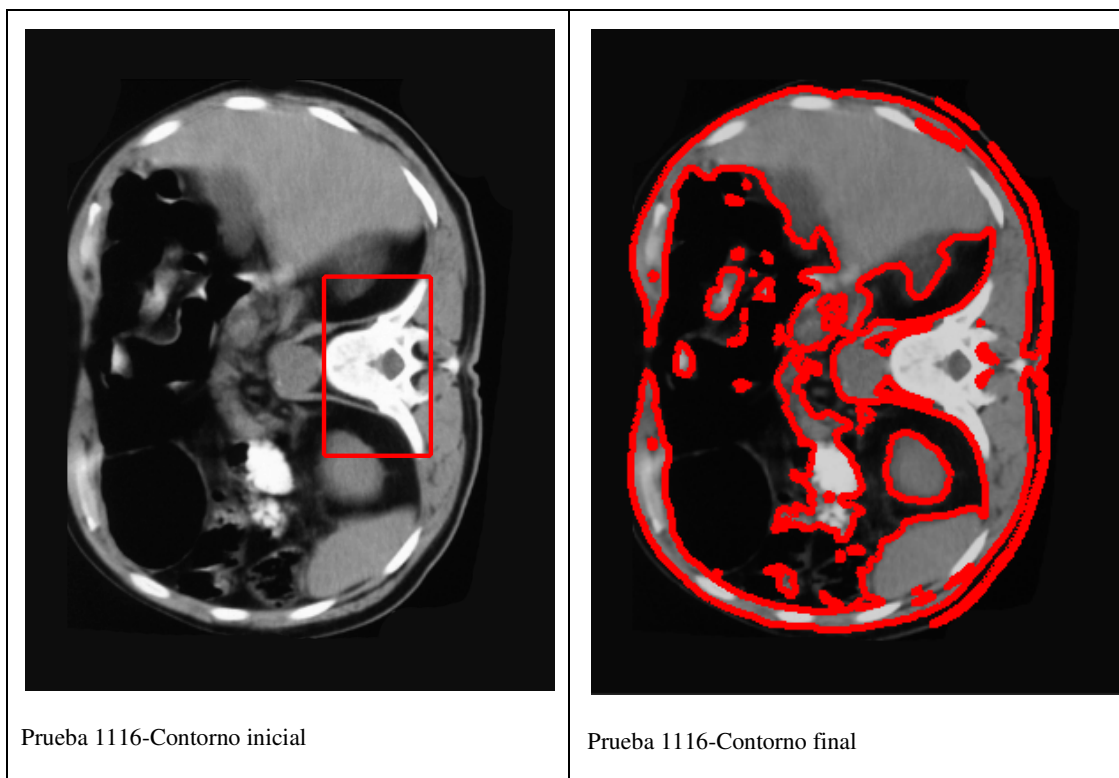


Figura 3.17. Prueba ejecutada con el método de Bernard para la imagen 2

Revisemos varios experimentos efectuados con el método de Bernard para la imagen 3:

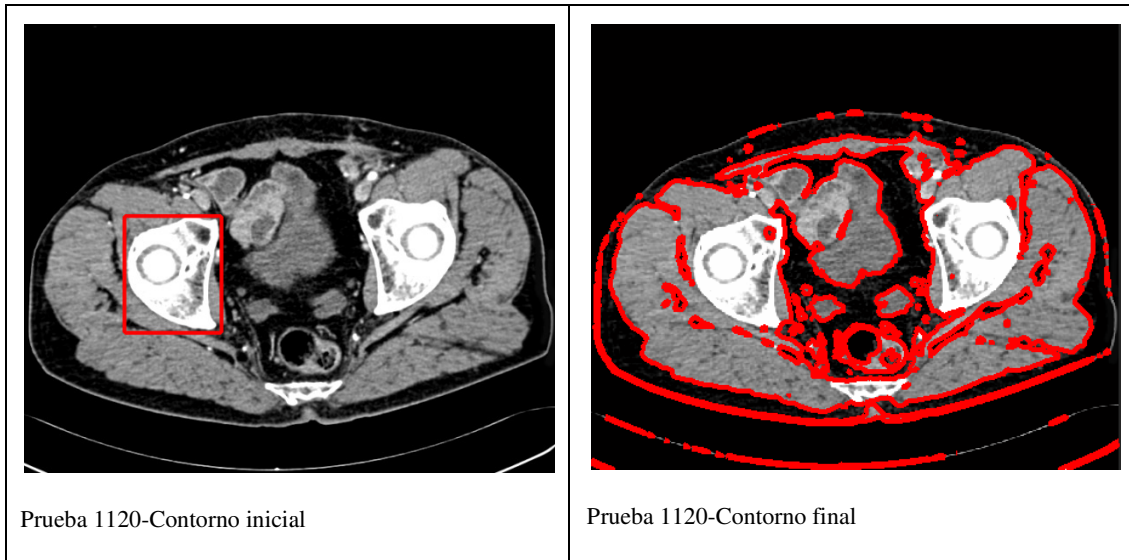


Figura 3.18. Prueba ejecutada con el método de Bernard para la imagen 3

Veamos varios casos de experimentos ejecutados con el método de Bernard para la imagen 7:

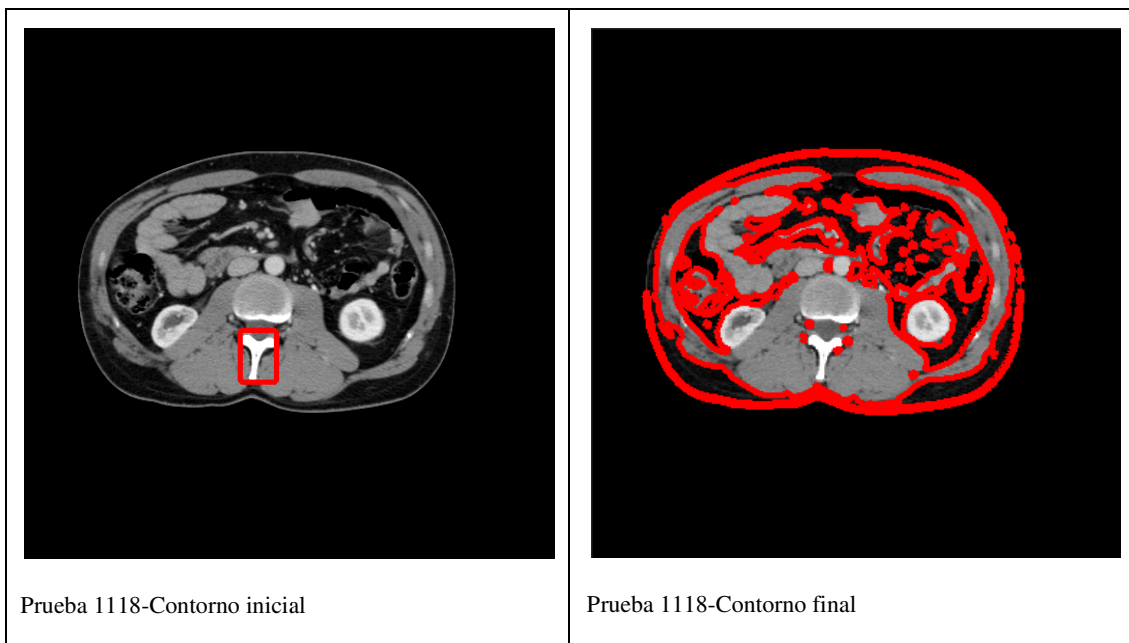


Figura 3.19. Experimento realizado con el método de Bernard para la imagen 3

Estudiamos algunos ejemplos de experimentos realizados con el método de Bernard para la imagen Test BN:

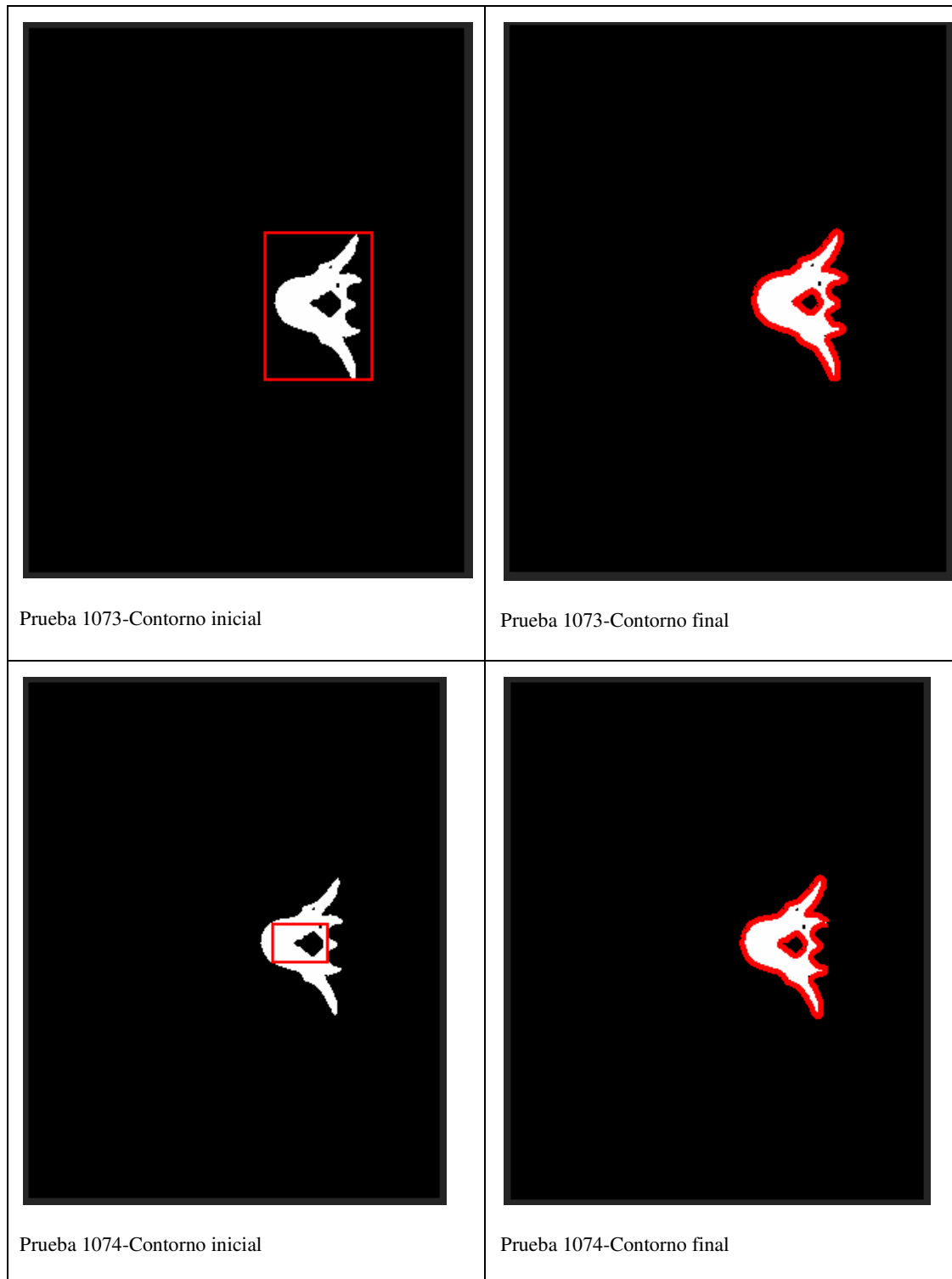


Figura 3.20. Ensayo realizado con el método de Bernard para la imagen Test-BN

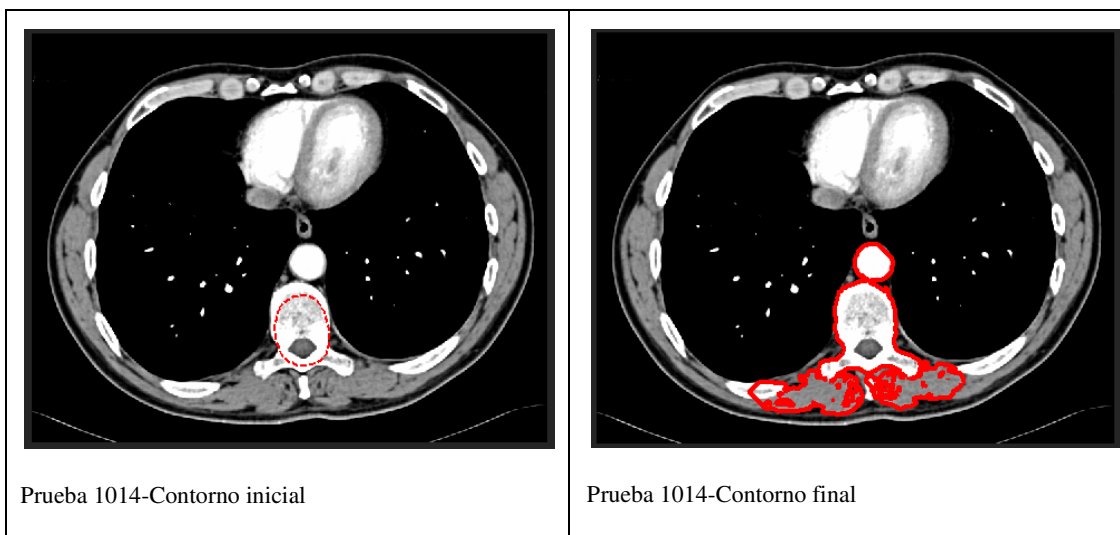
De los resultados obtenidos en estos y otros experimentos realizados con el método de Bernard, podemos extraer las siguientes conclusiones:

-independientemente de la forma en que se establezca el contorno inicial, el método de Bernard intenta encontrar todos los contornos que haya en toda la imagen

-la forma en que funciona este método y el hecho de segmentar toda la imagen convierten al algoritmo de Bernard en un buen método para imágenes en las que el objetivo sea segmentar muchos objetos a la vez y que la topología de estos objetos no sea muy compleja. Pero para imágenes donde haya varios objetos y el objetivo sea segmentar solamente uno, no es un buen método.

3.2.1.5. Problemas encontrados en algoritmo de Shi

Comenzaremos exponiendo algunos experimentos realizados con el algoritmo de Shi [SHI08] y posteriormente extraeremos algunas conclusiones. Empezamos con algunas pruebas realizadas para la imagen 1:



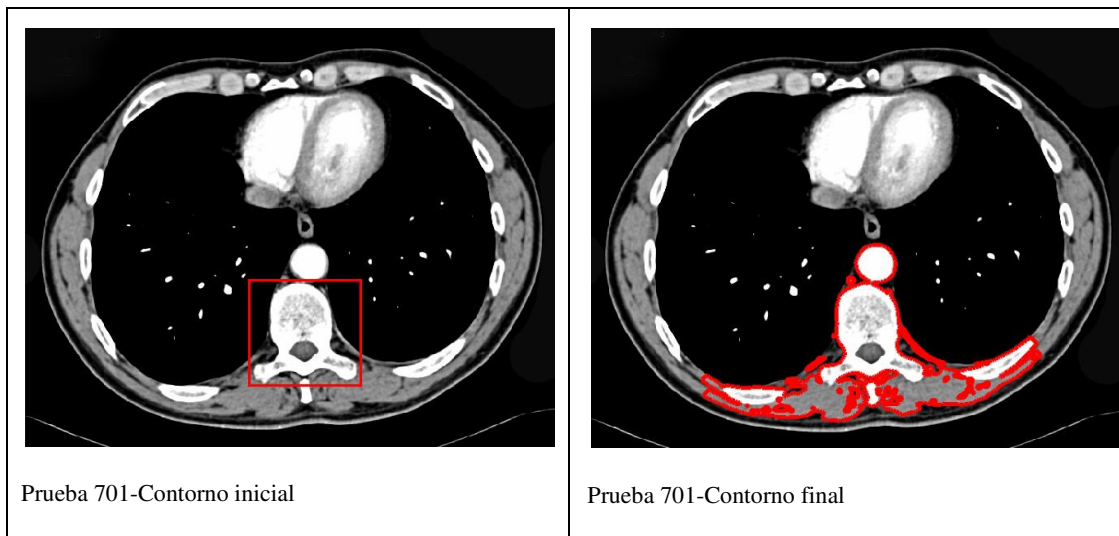


Figura 3.21. Pruebas realizadas con el método de Shi para la imagen 1

Estudiemos algunos ejemplos de experimentos efectuados con el método de Shi para la imagen 2:

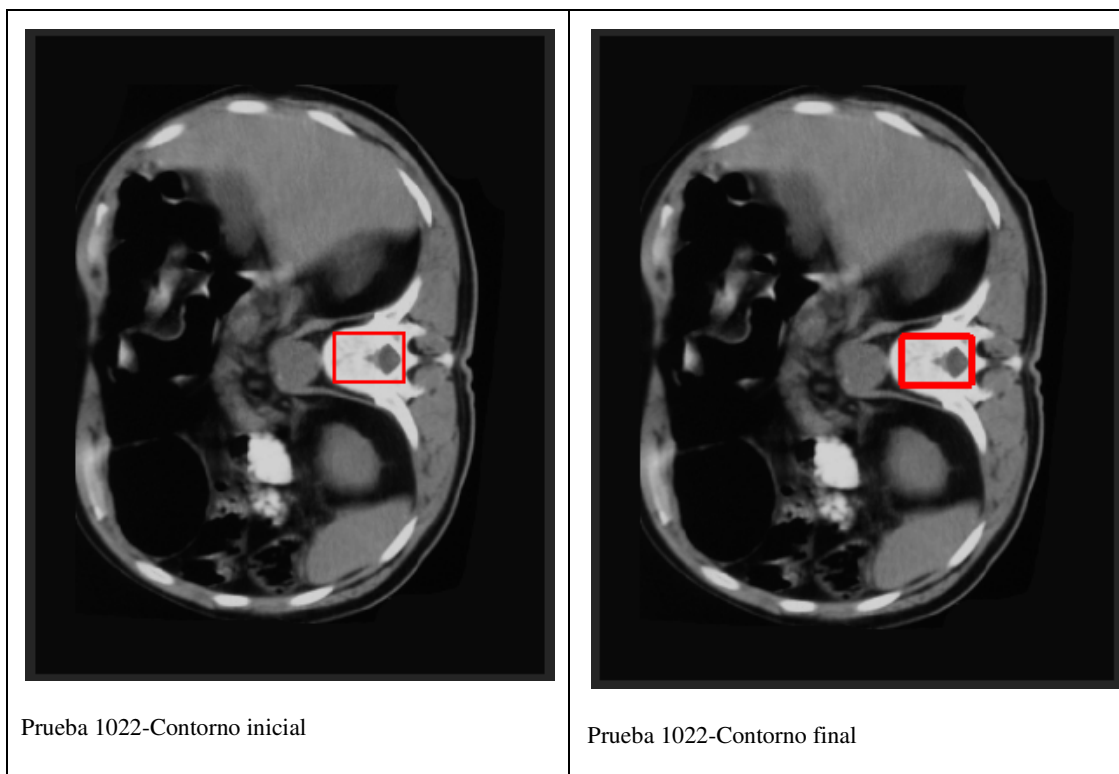


Figura 3.22. Pruebas realizadas con el método de Shi para la imagen 2

Observemos algunos ejemplos de ensayos practicados con el método de Shi para la imagen 3:



Prueba 1055-Contorno inicial



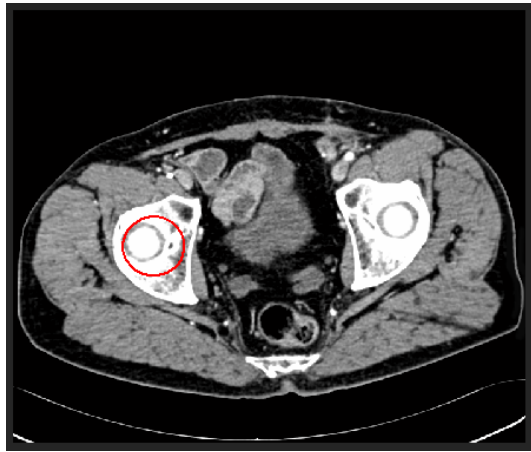
Prueba 1055-Contorno final



Prueba 1056-Contorno inicial



Prueba 1056-Contorno final



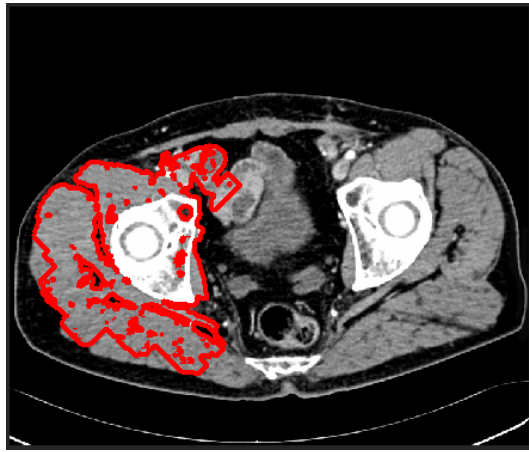
Prueba 1057-Contorno inicial



Prueba 1057-Contorno final



Prueba 1058-Contorno inicial



Prueba 1058-Contorno final



Prueba 1059-Contorno inicial



Prueba 1059-Contorno final



Prueba 1060-Contorno inicial



Prueba 1060-Contorno final



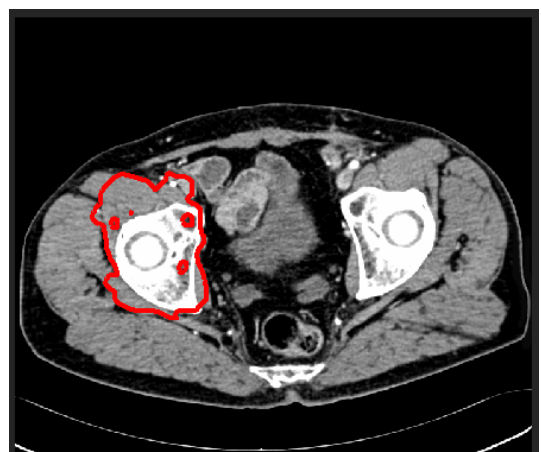
Prueba 1061-Contorno inicial



Prueba 1061-Contorno final



Prueba 1062-Contorno inicial



Prueba 1062-Contorno final

Figura 3.23. Pruebas ejecutadas con el método de Shi para la imagen 3

Analicemos varios ejemplos de experimentos practicados con el método de Shi para la imagen 7:

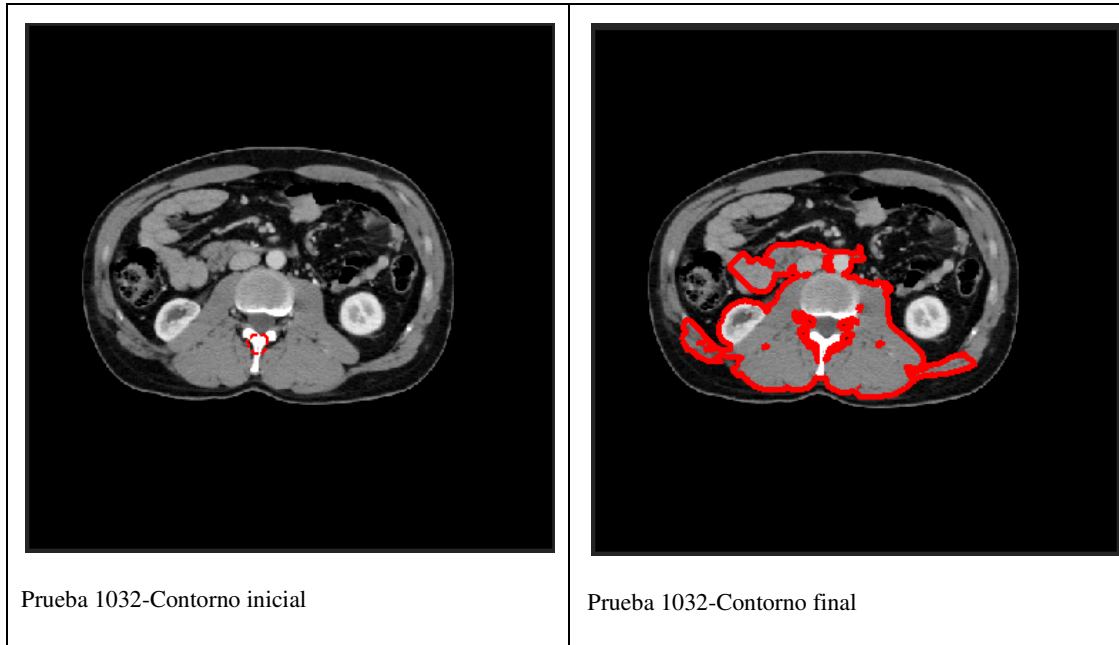


Figura 3.24. Pruebas realizadas con el método de Shi para la imagen 7

Observemos algunos ejemplos de pruebas realizadas con el método de Shi para la imagen Test-BN:

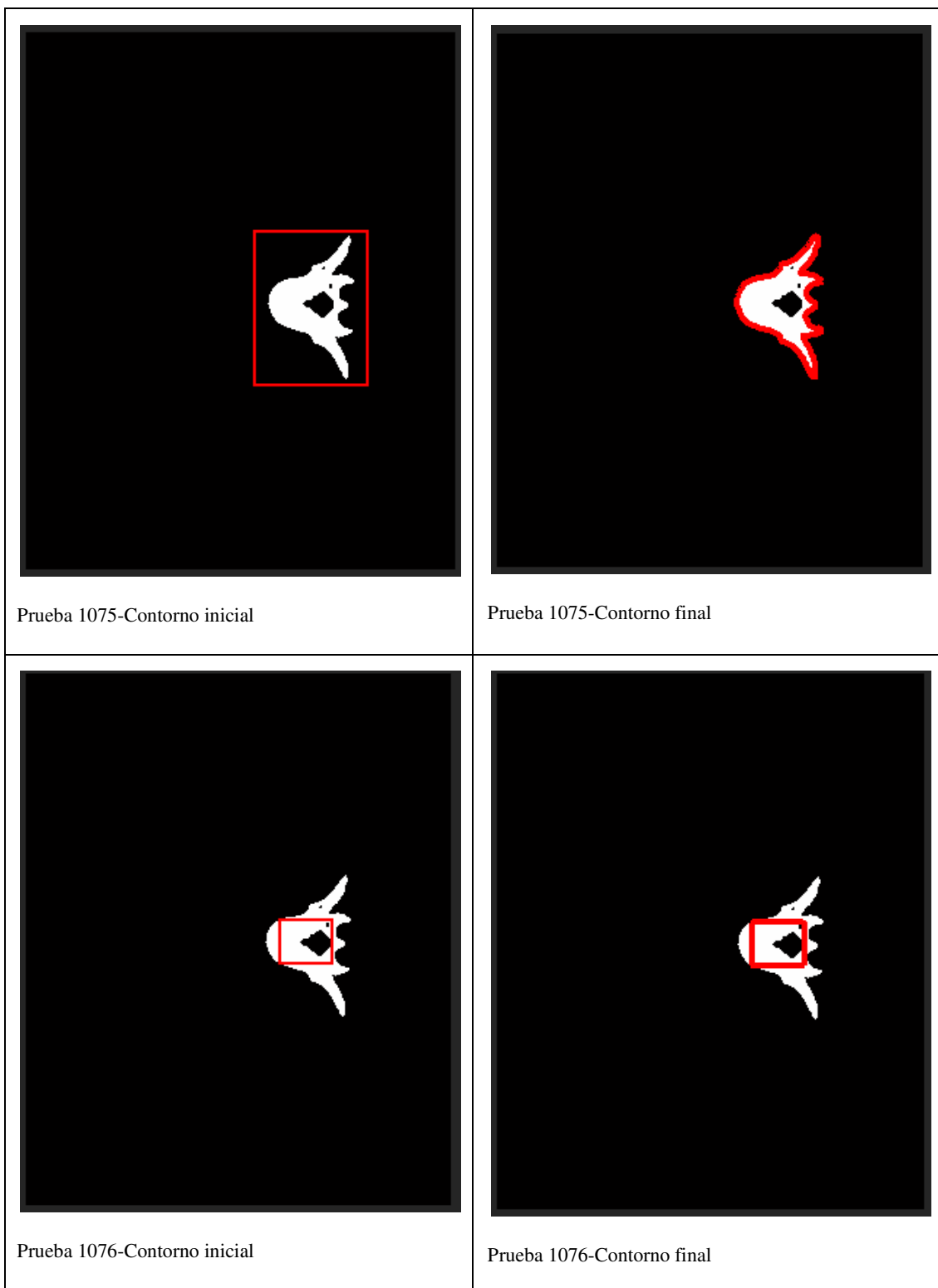


Figura 3.25. Pruebas ejecutadas con el método de Shi para la imagen Test-BN

Tras realizar estos experimentos, podemos concluir que:

-el método de Shi, normalmente sigue su evolución traspasando las fronteras del objeto y expandiéndose mucho más allá, con lo que los resultados suelen contener bastantes errores.

-hemos de seguir estudiando a fondo este método, para ver si obtiene buenos resultados en alguna circunstancia concreta. Por ahora, con los experimentos realizados, en la mayoría de los casos el resultado fue bastante malo.

-cuando inicializamos el contorno por dentro y en forma de rectángulo, suele evolucionar poco y termina con una forma parecida a la inicial.

-la prueba en la que mejor se comportó fue al ejecutarlo sobre una imagen binaria.

3.2.1.6. Problemas encontrados en algoritmo de Li

Veamos primero algunos ensayos realizados con el algoritmo de Li [LI08] y posteriormente extraeremos algunas conclusiones sobre estas pruebas. En este algoritmo los resultados fueron tan malos que no ha habido diferencias significativas entre los experimentos realizados con unas configuraciones u otras. Es por esto que mostramos menor cantidad de comprobaciones que en otros métodos. Empezamos con la imagen 1:

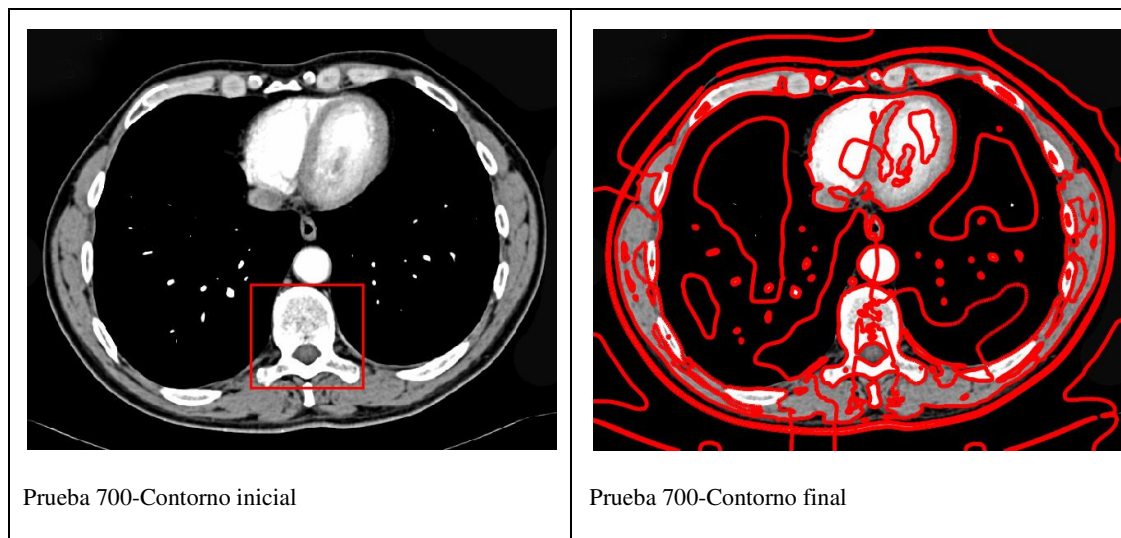


Figura 3.26. Pruebas realizadas con el método de Li para la imagen 1

Prueba con el método de Li para la imagen 2:

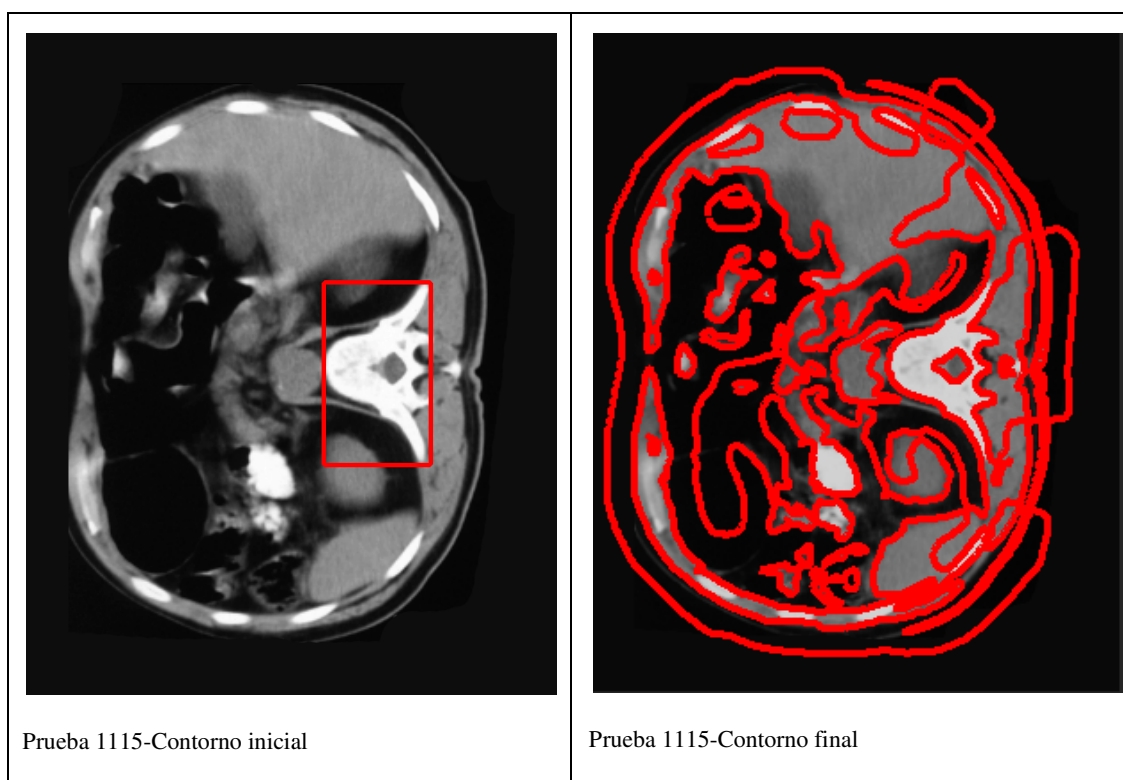


Figura 3.27. Ensayos realizadas con el método de Li para la imagen 2

Prueba con el método de Li para la imagen 3:

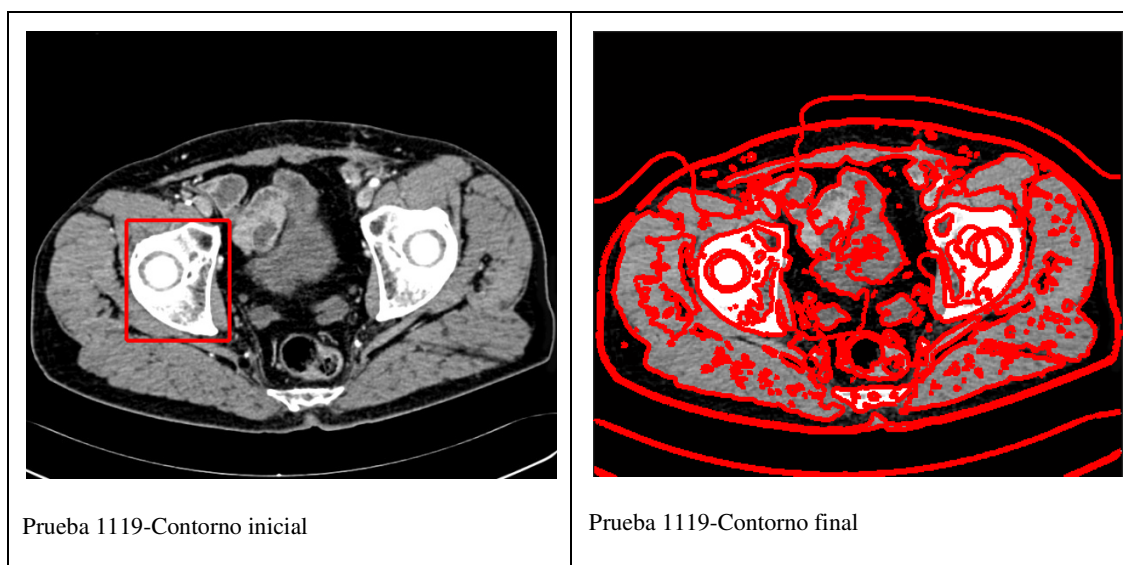


Figura 3.28. Ensayos realizados con el método de Li para la imagen 3

Experimento con el método de Li para la imagen 7:

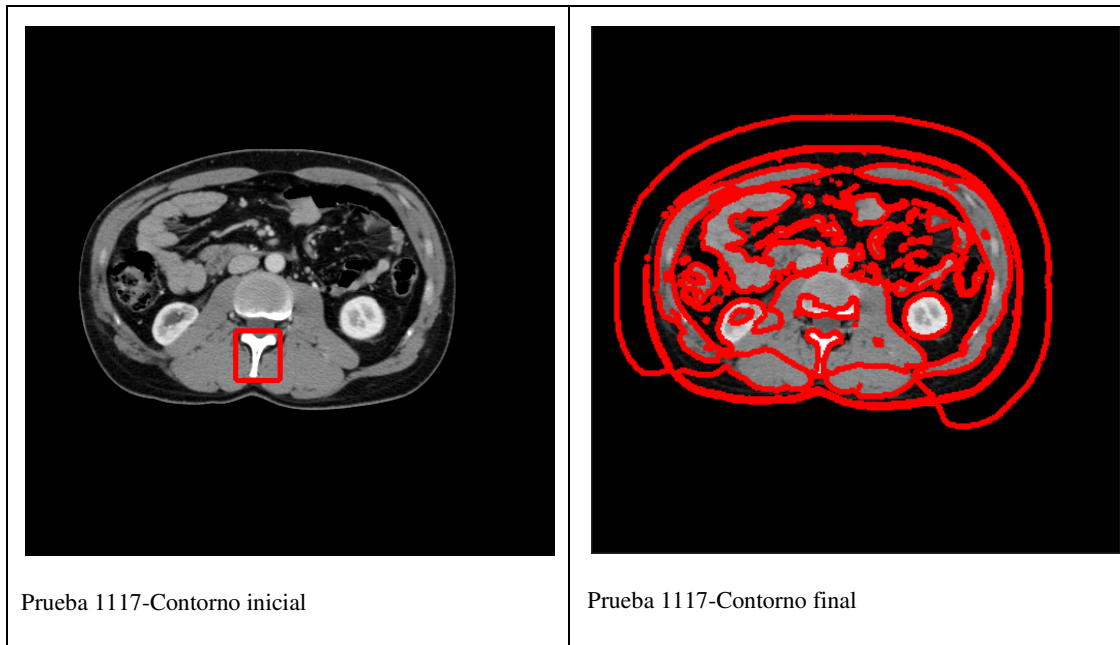


Figura 3.29. Prueba realizada con el método de Li para la imagen 7

Observemos algunos ejemplos de ensayos efectuados con el método de Li para la imagen Test BN:

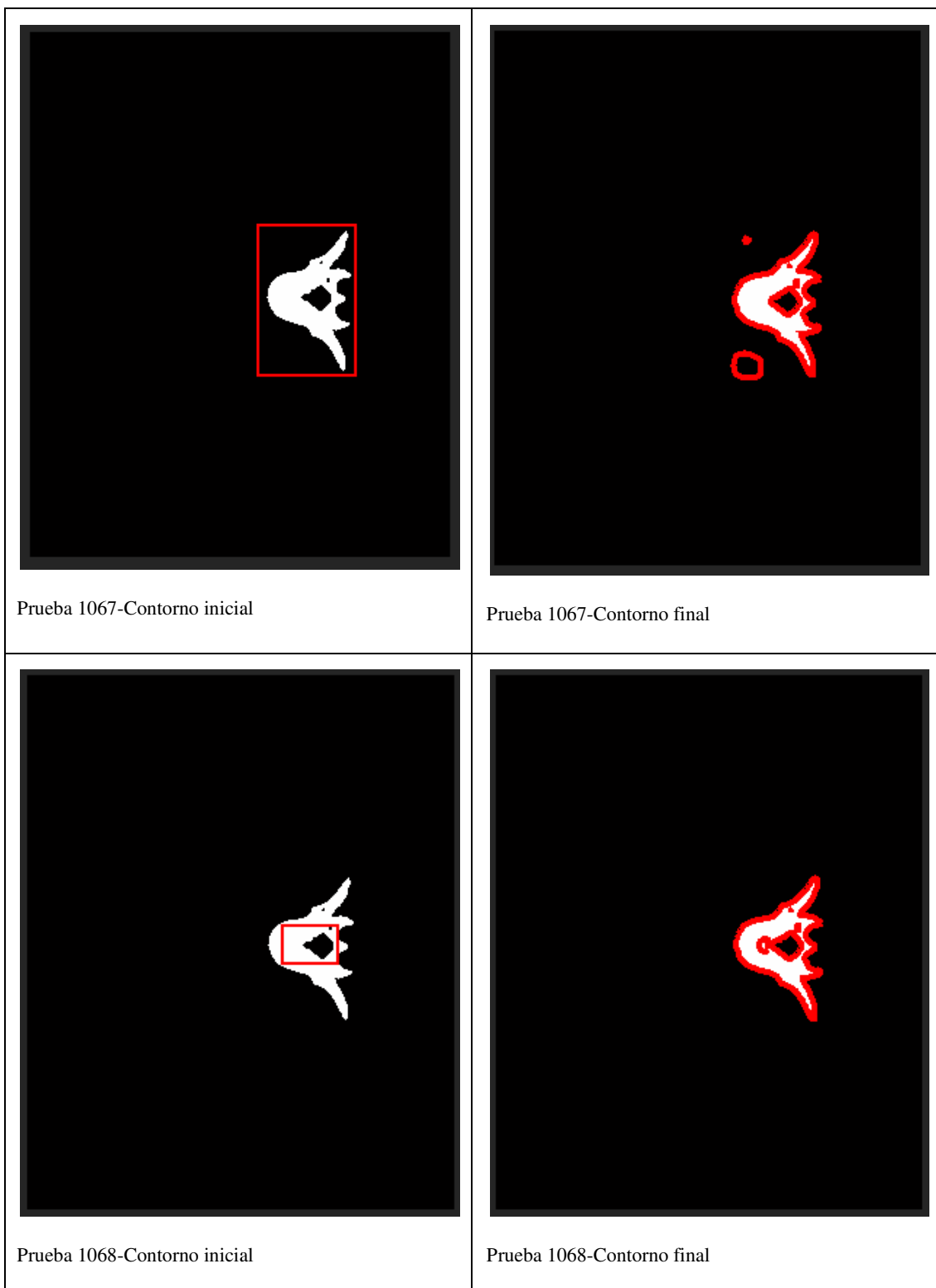


Figura 3.30. Pruebas realizadas con el método de Li para la imagen Test-BN

Sobre el algoritmo de Li podemos concluir que:

- procesa toda la imagen, sin tener en cuenta el contorno que se haya establecido inicialmente,

- suele encontrar demasiados contornos, pues parece detectar todas las fronteras que existan en la imagen y también algunas que aparentemente no lo son, al menos no para un experto humano. Parece establecer un contorno en cualquier cambio importante en el nivel de intensidad de la imagen, además sin separar unas fronteras de otras, por lo que no aporta buenos resultados para la segmentación,

- obtiene malos resultados incluso en los ensayos con imágenes binarias.

Conclusiones de las pruebas realizadas con contornos activos previos

Antes de indicar nuestras conclusiones sobre los experimentos anteriores, deseamos indicar que las pruebas aportadas en los apartados previos han tenido como objetivo facilitar un análisis visual de los resultados de los diferentes algoritmos para indicar las situaciones en los que fallaba cada uno, para posteriormente intentar solucionar con nuestros métodos este tipo de situaciones.

Cada método está diseñado para que genere automáticamente un vector de características (con un conjunto de medidas y descriptores) en cada ejecución, que se graba en disco como un fichero independiente, por lo que no hubiera sido complicado indicar todas las medidas generadas para cada ensayo realizado, pero no hemos considerado conveniente exhibir todos los resultados numéricos pues hemos entendido que podría ser contraproducente, por aportar una cantidad excesiva de datos que serían más difíciles de interpretar que los propios resultados visuales.

Como ha quedado demostrado a lo largo del presente capítulo, a tenor de los experimentos realizados y los algoritmos revisados y probados (el algoritmo Clásico de Kass, Witkin y Terzopoulos [KASS87], Williams y Shah [WILLIAMS92], Balloon [COHEN91a], Caselles [CASELLES97], Lankton[LANKTON08], Shi[SHI08], Chan [CHAN01], Bernard [BERNARD09] y Li [LI08], entre otros), presentan resultados satisfactorios en algunos casos y en otros no.

Los métodos de Kass, Witkin y Terzopoulos, Williams y Shah, los Balloon de Cohen et al. son considerados métodos clásicos dentro del mundo de los contornos activos y por esto nos hemos centrado menos en el estudio de estos y hemos realizado menos comprobaciones con ellos. En cambio, hemos realizado más ensayos con los métodos de Caselles, Lankton, Shi, Chan, Bernard y Li debido a que estos métodos los conocíamos menos y esos ensayos podían aportarnos más información desconocida hasta ese momento.

El contorno activo que nos dio mejores resultados fue el método de Chan. En algunos métodos la exactitud del contorno resultante dependía mucho de la habilidad que tuviese el usuario para configurar los diferentes aspectos abiertos a ser configurados por el mismo, lo que hacía que en muchos casos hubiese que ser un auténtico experto en ese tipo de contorno activo para obtener unos contornos resultantes satisfactorios. Esto dificulta mucho que puedan ser utilizados de forma automática. En muchos casos, la configuración que funcionaba bien para una imagen, no funcionaba bien para la siguiente, lo que nos llevó a concluir que era casi imposible encontrar, entre los métodos revisados y probados, un contorno activo que funcionase bien en todos los casos y en todas las situaciones.

Una vez analizados todos los resultados de todos los experimentos efectuados con los contornos activos que analizamos, pensamos en crear un nuevo tipo de contorno activo que fuese capaz de funcionar bien en una mayor cantidad de situaciones diferentes. Y esta ha sido la idea principal que ha inspirado todos los métodos desarrollados en esta tesis y que se muestran en el resto de este capítulo.

3.2.2. Segmentación con Contornos Activos propuestos

Ya que uno de los objetivos principales de esta tesis es la elaboración de una nueva metodología para la extracción de contornos activos bidimensionales, con el fin de realizar reconstrucciones tridimensionales, necesitamos técnicas capaces de extraer dichos contornos con una fiabilidad muy alta y en diversas condiciones. Los resultados expuestos hasta ahora en este documento en lo que respecta a las pruebas de algoritmos presentes en la bibliografía de contornos activos, nos muestran sin ningún lugar a dudas que todos presentan errores de funcionamiento en unas u otras circunstancias (ver conclusiones del apartado anterior).

Los métodos probados fallaban, por ejemplo, cuando había concavidades que eran de difícil acceso para el contorno activo, cuando había en el borde del objeto zonas muy puntiagudas, o pequeños entrantes en el borde que posteriormente, en la zona interior, se ensanchaban, como podemos observar en algunos de los resultados mostrados a continuación:

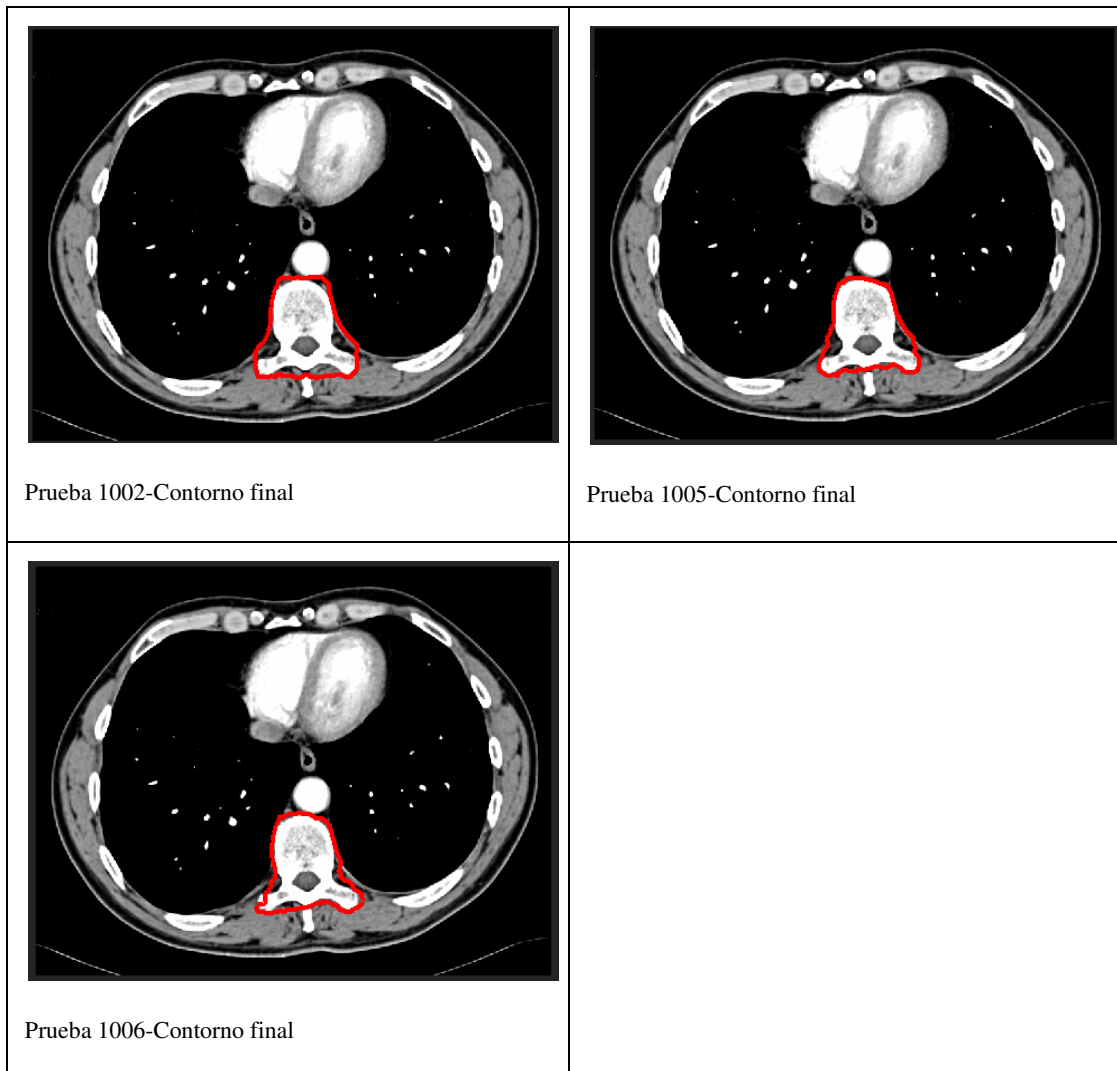


Figura 3.31. Ensayos realizados sobre la imagen 1 con el método de Caselles.

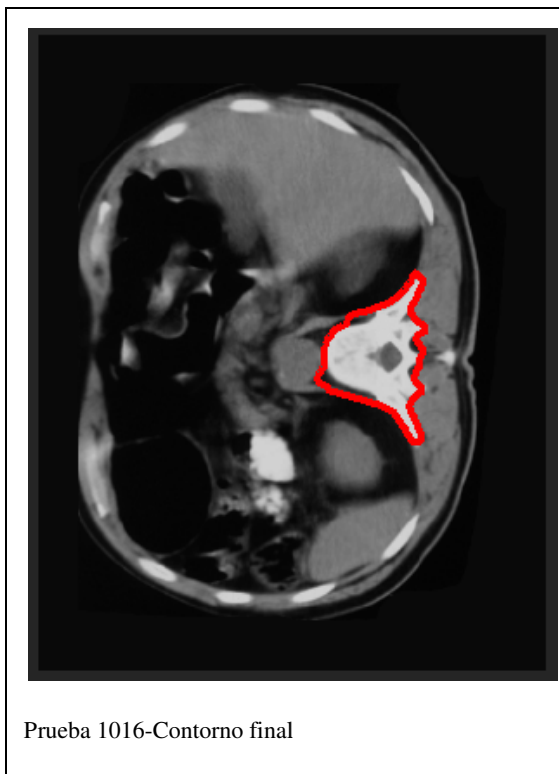


Figura 3.32. Prueba realizada sobre la imagen 2 con el método de Caselles

Para solucionar este tipo de situaciones donde muchos contornos activos fallaban, pensamos en utilizar una estrategia que solucionase la mayor cantidad posible de errores diferentes producidos en los contornos activos. Como unos tipos de contornos activos fallaban en unos casos y otros tipos diferentes fallaban en otros casos distintos, pensamos en desarrollar un nuevo tipo de contorno activo que fuese capaz de aplicar unas estrategias u otras según el tipo de dificultad que se encontrase en cada ocasión. De esta forma se idearon los Contornos Activos Auto-Reconfigurables que se explican en los siguientes apartados.

3.2.2.1. Contornos Activos Auto-Reconfigurables

Los Contornos Activos Auto-Reconfigurables (CAAR) son un nuevo tipo de contornos activos desarrollados en el marco de este trabajo. Como ya se ha comentado anteriormente, los CAAR consisten en una nueva filosofía de contornos activos que busca

obtener mejores resultados que otros tipos de contornos activos existentes previamente. Para conseguir esto, los CAAR se desarrollan en dos fases: en la primera fase, se realiza una primera aproximación a la segmentación del objeto de interés, obteniendo como resultado temporal un contorno. Este contorno, en muchas ocasiones, no habrá encontrado de forma exacta la frontera del objeto de interés. Hasta este punto sería como la ejecución de un contorno activo clásico. En la figura siguiente se muestra un esquema general de un CAAR, donde, en la primera fase, se ha utilizado el algoritmo clásico de Kass, Witkin y Terzopoulos[KASS87] (KWT), aunque podría utilizarse otro algoritmo cualquiera de contornos activos.

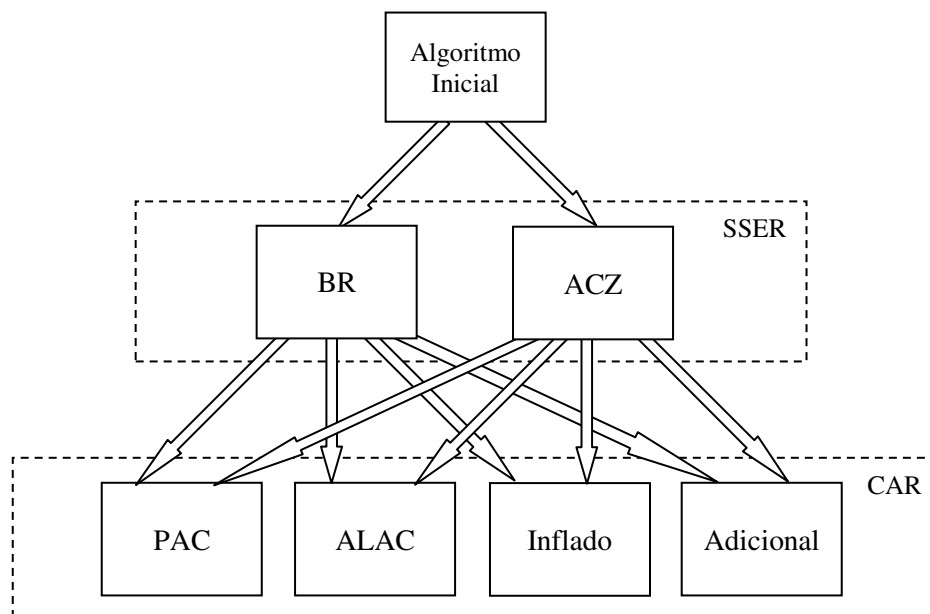


Figura 3.33. Esquema de Contorno Activo Auto-Reconfigurable (CAAR)

En este momento, en el que normalmente ya se habría acabado de ejecutar un contorno activo cualquiera (independientemente de que sus resultados hubiesen sido de menor o mayor exactitud) es cuando empieza a ejecutarse la segunda fase de los CAAR, que es una de las partes innovadoras de este nuevo tipo de contornos activos. En este punto, los Contornos Activos Auto-Reconfigurables, inician una etapa de estudio de la situación alcanzada al final de la primera fase, buscando zonas en las que posiblemente no se haya encontrado la frontera del objeto de interés, estudiando las características de esas zonas y buscando la mejor estrategia para segmentar correctamente esas zonas en las que

aparentemente el contorno activo no se ha adaptado de forma correcta. Estos métodos que se utilizarán para segmentar esas zonas son los que hemos llamado Contornos Activos Reconfiguradores (CAR) que explicaremos en próximos apartados.

En la segunda fase de los Contornos Activos Auto-Reconfigurables, se podrán utilizar uno o varios de los Contornos Activos Reconfiguradores (CAR). Para decidir en cada momento qué CAR es mejor utilizar, se utiliza un sistema de selección de estrategia reconfiguradora (SSER). Para esto, se han creado dos SSER, ambos buscan información sobre las zonas en las que no se ha detectado bien la frontera del objeto de interés. El primer SSER, llamado *Búsqueda Longitudinal en Abanico (BLA)*, busca información realizando prospecciones sobre la zona en cuestión. El segundo, llamado *Análisis de las Cualidades de la Zona (ACZ)* observa las cualidades de la zona que se está analizando y extrae información útil para la toma de decisión sobre el CAR a ejecutar en ese momento y en esa zona. Todos estos elementos de los CAAR se explicarán en los siguientes apartados de este capítulo.

3.2.2.2. Contornos Activos Reconfiguradores

En este apartado vamos a detallar los Contornos Activos Reconfiguradores (CAR), que son, a grandes rasgos, los algoritmos de contornos activos que pueden ejecutarse durante la segunda fase de los Contornos Activos Auto-Reconfigurables (CAAR).

En este trabajo se han creado varios CAR expresamente para ser usados en la segunda fase de los Contornos Activos Auto-Reconfigurables, pero podrían utilizarse también otros algoritmos de contornos activos ya existentes en la bibliografía, siempre teniendo en cuenta que deben ser adecuados para solucionar un problema concreto de segmentación, o para ejecutarse en una situación concreta, que debería ser estudiada en profundidad para poder incluirse dentro del algoritmo del CAAR.

En los siguientes apartados vamos a presentar dos algoritmos nuevos que se han creado para ser utilizados como contorno activo reconfigurador dentro de la segunda fase de los CAAR. Adicionalmente, se muestra el algoritmo de inflado, que es una modificación del algoritmo Balloon [COHEN91a] para ser utilizado con esa misma finalidad.

3.2.2.3. Contornos Activos con Reclamos Automáticos

Los Contornos Activos con Reclamos Automáticos (CARA), o si lo nombramos en inglés Automatic Lure Active Contour (ALAC) utilizan reclamos cuya posición se calcula de forma automática (no supervisada), lo que supone una novedad con respecto a algoritmos anteriores. Esto permite elegir de forma automática el punto al que debe aproximarse el contorno activo.

Antes de ejecutarse el algoritmo ALAC, previamente se habrá detectado la zona en la que el algoritmo inicial no se ha ajustado bien, delimitando los nodos del contorno activo que si han quedado situados en la frontera del objeto y los que no. Una vez identificados esos nodos que no han detectado correctamente la frontera del objeto, se define un conjunto de nodos (los nodos que no se han adaptado correctamente a la frontera). Después se identifican el primer y último nodo de los que si se han adaptado y se calcula el punto geométrico medio de ese tramo formado por el primer y último nodo. Desde ese punto geométrico medio se sigue una línea imaginaria en dirección perpendicular a ese tramo. Se sigue la línea hasta encontrar un punto que esté en la frontera del objeto o dentro del objeto de interés. En ese punto se establece un reclamo automático.

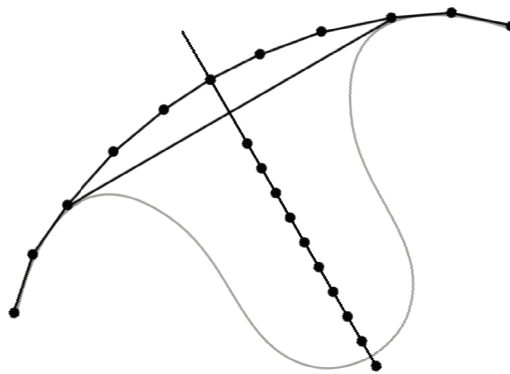


Figura 3.34. Búsqueda automática del punto donde establecer un reclamo automático.

Una vez establecido un reclamo automático se aplica, sobre los puntos que pertenecen al conjunto de nodos no adaptados a la frontera, una fuerza de atracción hacia el mismo. De esta forma, gracias al reclamo automático, todos los puntos de ese tramo se moverán en función de esa nueva fuerza que atrae a los nodos hacia el reclamo automático, haciendo que queden bien posicionados mayor cantidad de nodos.

En algunas casos, no es suficiente crear un único reclamo automático, por lo que todo el proceso anterior se repetirá tantas veces como sea necesario hasta que queden adaptados a la frontera todos los nodos de la zona que se quería corregir (o queden sin adaptar un número menor o igual que un umbral fijado previamente).



Figura. 3.35. La imagen muestra una lesión órbito-craneal (cortesía de Dra. María del Mar Ruiz-Martín) [RUIZ-MARTIN07]

En la imagen anterior, al aplicar el algoritmo empleado en la primera fase (en este caso el algoritmo de Williams y Shah [WILLIAMS92]) se obtuvo un resultado parecido al que se muestra esquemáticamente en la figura 3.36.a, que vamos a utilizar para mostrar gráficamente todo el proceso seguido durante el algoritmo ALAC.

En la segunda fase, se detectan los conjuntos de nodos que no están sobre la frontera del objeto. En el ejemplo de la figura 3.36.a, se detectan tres conjuntos y vamos a indicar el resultado de aplicar un contorno activo con reclamos automáticos al conjunto de nodos no adaptados a la frontera que aparece en la parte superior izquierda de dicha figura.

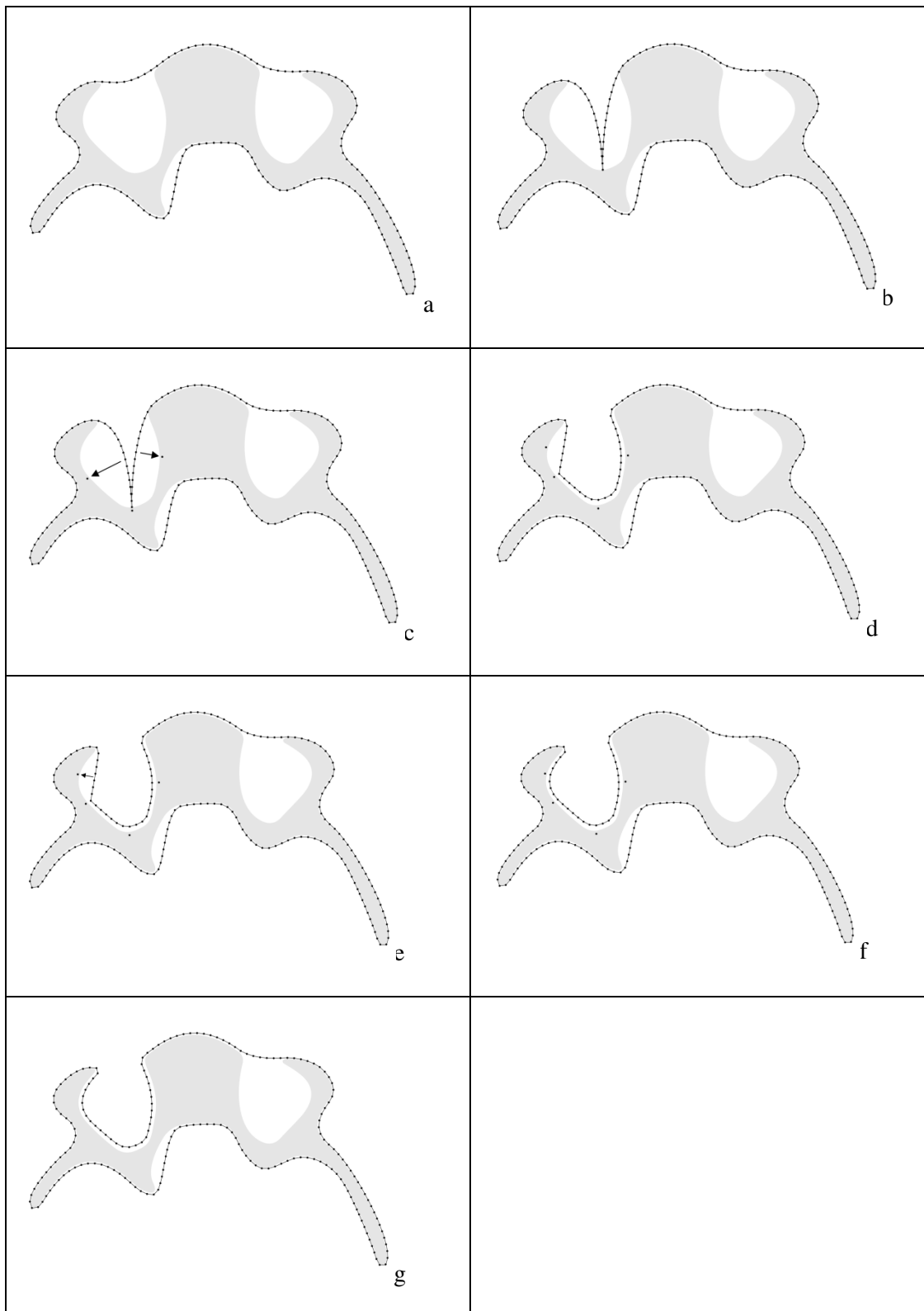


Figura 3.36. Fases de la ejecución de ALAC. Figura 3.36.a: resultado de aplicar un contorno activo clásico. Figura 3.36.b: resultado de aplicar un reclamo automático y creación de dos nuevos reclamos automáticos más. Figura 3.36.c: fuerzas resultantes de aplicar los dos nuevos reclamos

automáticos. Figura 3.36.d: resultado de aplicar los dos reclamos automáticos anteriores y creación de otro nuevo reclamo automático. Figura 3.36.e: fuerza resultante del último reclamo automático. Figura 3.36.f : resultado final de ese conjunto de nodos, indicando las posiciones de los reclamos automáticos generados. Figuras 3.36.g: resultado final de ese conjunto de nodos.

A partir de este punto, y todavía dentro de la segunda fase del CAAR, en cada tramo de nodos que no se han adaptado a la frontera al final de la primera fase, se aplicaría el Contorno Activo Reconfigurador (CAR) que se decide con el Sistema de Selección de Estrategia Reconfiguradora (SSER) ejecutado, dando como resultado que se aplicaría el CAR que se considere óptimo en cada zona. Todo el proceso dará como resultado final que el contorno completo quede mejor adaptado a la frontera del objeto de interés.

3.2.2.4. Pushing Active Contours

Los contornos activos *Pushing Active Contours (PAC)*, que se podrían traducir como *Contornos Activos de Empuje (CAE)* son otro de los algoritmos reconfiguradores que se han utilizado dentro de los CAAR. Los PAC consisten en un contorno activo basado en la formulación clásica, pero al que se le ha añadido una fuerza de empuje. Esta fuerza de empuje va siempre hacia dentro del contorno y afecta exclusivamente a una zona concreta, no a todo el contorno activo, y esto provoca que los nodos de esa zona del contorno activo tengan mucha tendencia a ir hacia dentro y en la dirección que indique la fuerza de empuje. Con esta nueva fuerza se consigue que los nodos del contorno activo se aproximen mejor a puntos de la frontera del objeto que tienen una posición alejada y profunda respecto al resto de la frontera reconocida hasta ese momento. Así, cuando el CAAR detecta una situación de este tipo, selecciona este algoritmo reconfigurador como solución a esa situación. Las situaciones en las que se seleccionará este algoritmo reconfigurador se verán en el apartado donde se explican los SSER (Sistemas de Selección de Estrategia Reconfiguradora).

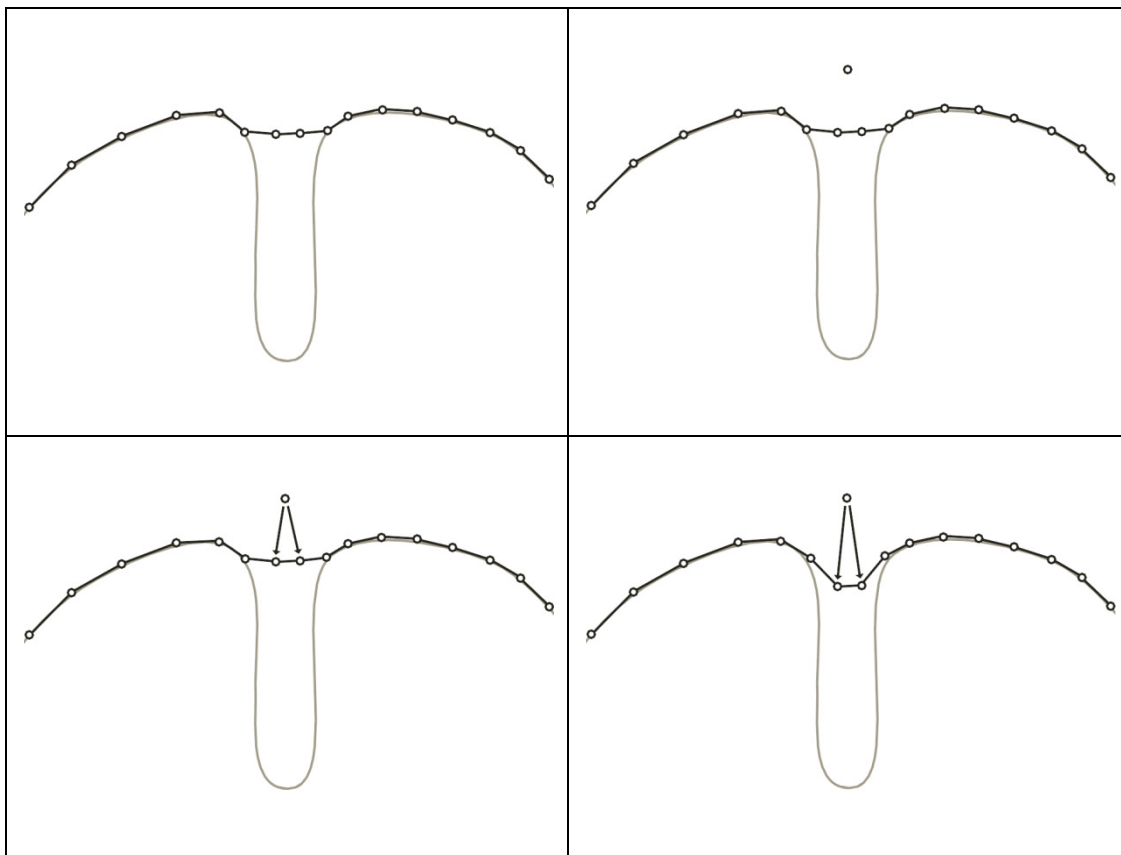
Como ya se ha comentado anteriormente, para conseguir el efecto corrector sobre el tramo problemático (definido con los nodos del contorno activo que no se han aproximado a la frontera) se define la *Energía de Empuje*, que sólo afectará a ese conjunto de nodos, pero no al resto del contorno activo. Esta nueva fuerza se ejerce desde un punto externo al contorno activo, llamado *Punto de Empuje* y desde el que parte la fuerza de empuje hacia los

nodos afectados. Esta energía se calcula como la distancia entre el punto de empuje y el nodo en cuestión, poniéndole signo negativo para que la energía disminuya cuanto mayor sea la distancia entre el nodo y el punto de empuje.

$$\text{Fuerza de Empuje} = -|\text{dist}(\text{PuntoEmpuje}, \text{nodo})|$$

Este algoritmo reconfigurador pararía cuando ya se hayan posicionado todos los nodos en la frontera del objeto o bien el número de puntos que no lo hayan hecho sean menores que un umbral.

Esta fuerza de empuje, a diferencia de las que se ejercen en el algoritmo de Reclamos Automáticos, se ejerce desde un único punto fijo, a diferencia de los reclamos automáticos que se van creando y posicionando dinámicamente, de forma que pueden ejercer fuerza desde varios puntos diferentes. Otra diferencia es que la fuerza de los reclamos automáticos es de atracción, no de empuje, y los puntos de influencia (los reclamos) se sitúan dentro del contorno activo y no fuera, como es el caso del algoritmo PAC.



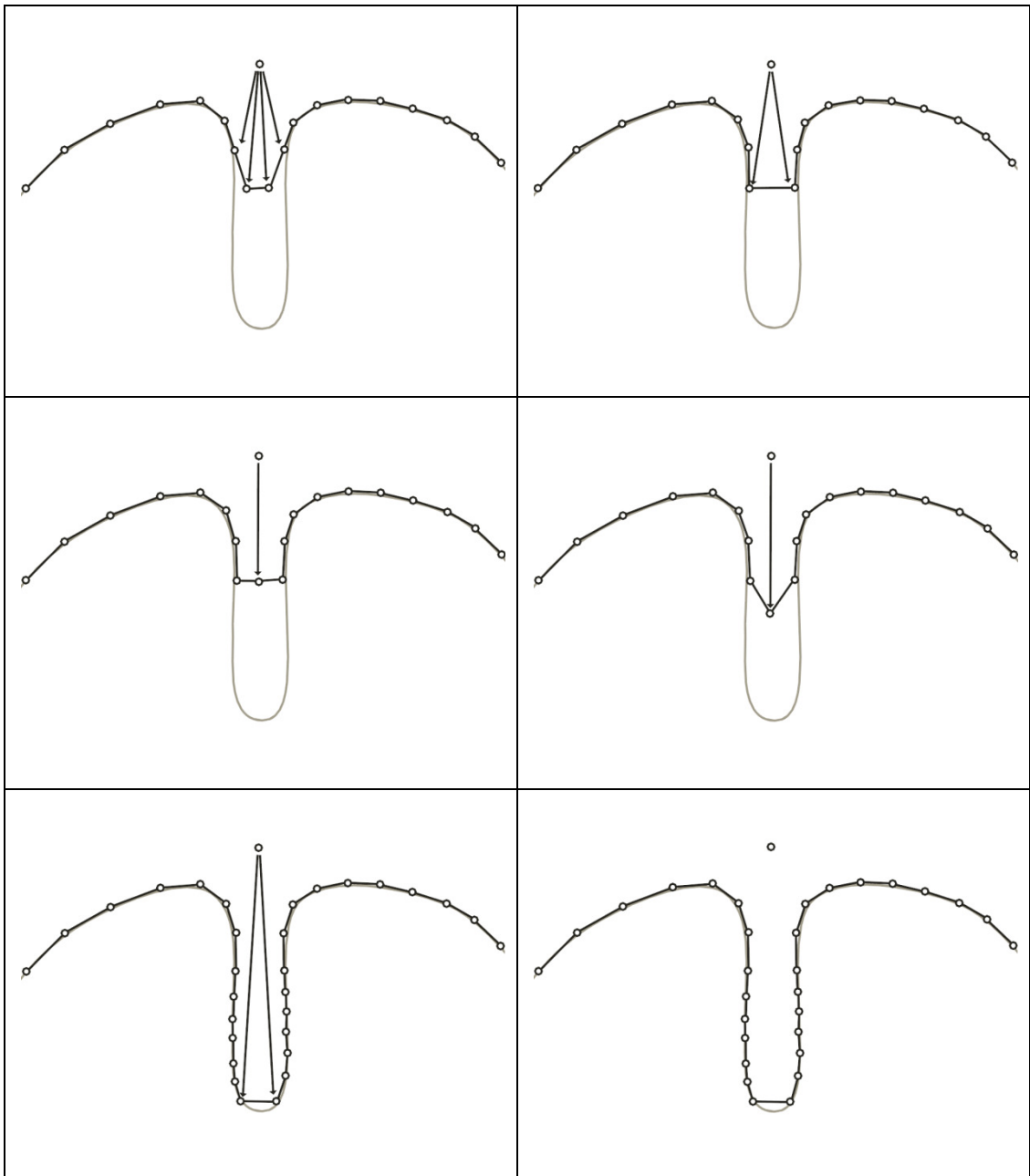


Figura 3.37: Fases de la ejecución de PAC. Figura 3.37.a: resultado de la primera fase del CAAR. Figura 3.37.b: creación del punto de empuje. Figura 3.37.c,d: la fuerza de empuje desplaza los nodos no adaptados alejándolos del punto de empuje. Figura 3.37.e,f,g,h: Cuando los nodos se separan más de un umbral, se crean nuevos nodos. Figuras 3.37.i, j: resultado final de ese tramo del CAAR.

3.2.2.5. Sistemas de Selección de Estrategia Reconfiguradora

En este apartado vamos a ver el sistema que se ha elaborado para seleccionar la estrategia más adecuada para, en el contorno activo resultante de la primera fase del contorno activo auto-reconfigurable, corregir los tramos que no se hayan ajustado correctamente. Por tanto, este sistema de selección nos sirve para escoger el algoritmo reconfigurador que se ejecutará en cada tramo y en cada momento durante la segunda fase de los contornos activos auto-reconfigurables.

En el sistema que mostramos, el SSER se ha utilizado para elegir entre tres algoritmos reconfiguradores (contornos activos con reclamos automáticos, pushing active contours y algoritmo de inflado).

En el futuro sería posible añadir más algoritmos entre los que elegir, sin hacer muchos cambios en el sistema. En este caso, además de insertar un nuevo algoritmo, habría que estudiar en que situaciones es mejor aplicar cada algoritmo reconfigurador, e incluir este conocimiento entre los criterios de decisión para elegir un algoritmo u otro.

Para que el sistema de selección elija una de estas tres estrategias (contornos activos con reclamos automáticos, pushing active contours o algoritmo de inflado), se pueden utilizar a su vez dos métodos diferentes: la *Búsqueda Longitudinal en Abanico* y el método basado en el *Análisis de las Cualidades de la Zona*. Los dos tienen la misma finalidad, por lo que podremos utilizar cualquiera de ellos como sistema de selección de estrategia reconfiguradora. Ambos se explican a continuación.

3.2.2.6. Búsqueda Longitudinal en Abanico

Examinemos, a continuación, uno de los métodos que se han diseñado para seleccionar el algoritmo más adecuado para corregir el error detectado en el contorno activo resultante de la primera fase del contorno activo auto-reconfigurable. Por tanto, este sistema de selección llamado *Búsqueda Longitudinal en Abanico (BLA)* nos sirve para escoger el algoritmo reconfigurador que se ejecutará en la segunda fase de los contornos activos auto-reconfigurables.

Como ya se ha adelantado en el apartado anterior, este método o sistema de selección tiene almacenado un identificador por cada posible CAR entre los que hay que elegir. En este caso lo utilizamos para elegir entre tres estrategias (contornos activos con reclamos automáticos, pushing active contours y algoritmo de inflado).

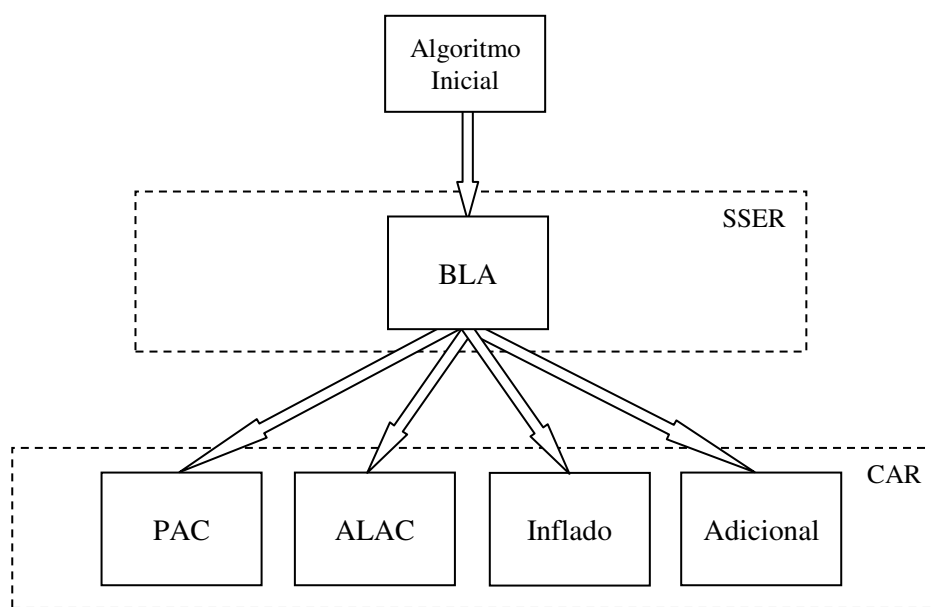


Figura 3.38. Esquema de CAAR utilizando BLA.

Como también se ha comentado anteriormente, este es un sistema que permite elegir entre muchos algoritmos, por lo que podríamos incorporar en el futuro más estrategias diferentes para solucionar los errores de segmentación que se vayan encontrando. Esto hace que sea un sistema muy abierto a incorporar nuevos algoritmos que vayan surgiendo en este campo científico. Para incorporarlo, bastaría analizar las circunstancias en las que se debería elegir este algoritmo con mayor preferencia sobre los demás algoritmos disponibles en el sistema.

Este método llamado *Búsqueda Longitudinal en Abanico (BLA)* está basado en el análisis de la estructura espacial de la región comprendida entre el contorno activo resultante de la primera fase del algoritmo de los contornos activos auto-reconfigurables y la frontera del objeto a segmentar. Los estudios experimentales realizados nos han mostrado que, según

la estructura espacial que tenga una región, es más probable que un tipo de algoritmo reconfigurador llegue a mejorar el resultado obtenido de la primera fase o no.

La estructura espacial de la zona en la que se ha producido el error en el resultado de la primera fase, se estudia en base a unas pequeñas prospecciones que se realizan sobre esa zona. Para realizar esas prospecciones, primero se identifica una serie de nodos en los que el contorno activo resultante de la primera fase del CAAR no ha quedado en la frontera del objeto a segmentar. Una vez identificada esa serie de nodos, se escoge el punto central del segmento formado por el primer y último nodos de esa serie. Este punto central será el punto de referencia desde el que iniciaremos las prospecciones.



Figura 3.39. Tramo de nodos a corregir

Las prospecciones se realizan en distintos puntos que se van alejando progresivamente y de forma equidistante respecto al punto de referencia.

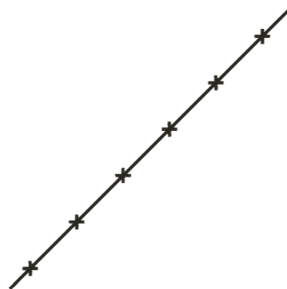


Figura 3.40. Distribución equidistante de prospecciones

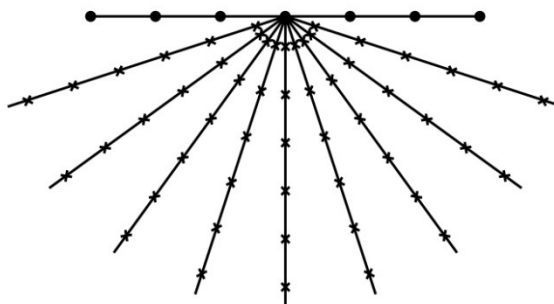


Figura 3.41. Distribución espacial de prospecciones, respecto al punto de referencia

Estas prospecciones se van realizando en posiciones que guardan una cierta distancia entre prospección y prospección, de forma que se puede describir como una búsqueda longitudinal a lo largo de cada una de las líneas rectas representadas en la figura. Después, se repite esa búsqueda en otra línea recta que dista de la anterior un ángulo preestablecido. En nuestro caso se le ha dado al ángulo un valor por defecto de 18 grados entre cada dos líneas de prospección, aunque este valor puede cambiarse.

Para cada uno de estos puntos (de prospección) se observa si el punto está encima del objeto que queremos segmentar o no. De estos puntos se estudiará, básicamente, la separación respecto al punto de referencia, cuantas prospecciones han sido necesarias hasta encontrar el objeto, y el ángulo respecto al segmento formado por el primer y último nodos del tramo de nodos a reposicionar.

Entre otros aspectos, nos interesa saber si los puntos encontrados están en una zona central o lateral respecto al punto de referencia. Si está en una de las dos rectas de la izquierda o en una de las dos rectas de la derecha, consideramos que está en una zona lateral. Si está en una de las tres rectas centrales, consideramos que está en una zona central.

Por otro lado, estudiando las prospecciones que se han situado encima del objeto, obtenemos información sobre la forma y amplitud de la "zona errónea" (zona donde no se ha adaptado bien el contorno activo de la primera fase).

En zonas donde el número de prospecciones necesarias hasta encontrar el objeto haya sido grande en casi todas las direcciones, se considera que es una zona amplia, lo que hace que sea una situación propicia para utilizar un algoritmo de inflado.

En los casos en los que las prospecciones que se hayan hecho en las zonas laterales hayan recaído en zona considerada como objeto de interés, mientras que en las prospecciones realizadas en las zonas frontales o centrales se haya tardado en encontrar píxeles encima del OI, se considera que es una zona más profunda que ancha, lo que hace que sea una situación adecuada para elegir los PAC (Pushing Active Contours) como algoritmo reconfigurador.

En los casos en los que las prospecciones necesarias en cada dirección hayan sido muy desiguales, se considera que es una zona asimétrica y probablemente con entrantes y salientes, lo que hace que sea más adecuado utilizar el algoritmo de reclamos automáticos.

El proceso anterior se repite para cada zona en la que hay nodos que no han encontrado la frontera del OI, de forma que el CAAR elegirá que CAR utilizar en cada zona siguiendo pautas como las indicadas anteriormente.

El algoritmo de Reclamos Automáticos tiene mayor probabilidad de ser utilizado cuando se encuentra con zonas asimétricas o recovecos de difícil acceso, pues tiene mayor flexibilidad en cuanto que se puede adaptar a mayor cantidad de situaciones diferentes, ya que va generando reclamos automáticos según los va necesitando.

La situación ideal para el algoritmo de Empuje es cuando la zona no ajustada tiene forma de hueco estrecho, muy profundo y simétrico, cuando no es tan simétrico, sigue comportándose bien. Pero si la asimetría es muy pronunciada, entonces ya no se comporta tan bien como el de Reclamos Automáticos.

En las pruebas realizadas el ángulo se incrementó en 18 grados entre dirección y dirección, por lo que se estudiaron 9 posibles direcciones (desde 0 a 180 grados, con separaciones de 18 grados), teniendo en cuenta que las direcciones de los grados 0 y 180 no se estudian pues estarían pegadas a los nodos del contorno activo que no encontraron la frontera (ver figura 3.____ de distribución espacial de prospecciones).

Una vez calculada la dirección en la que hay que hacer prospecciones, se realizan prospecciones en sentido longitudinal hasta que se encuentra que una prospección parece estar encima del objeto de interés. Al hacer las prospecciones en sentido longitudinal, esto permite que se pueda profundizar más en unas direcciones que en otras, pudiendo llegar a ser muy dispares las longitudes recorridas en unas direcciones o en otras, lo que nos da una información bastante fiel de la estructura espacial del hueco que se encuentra entre el

conjunto de nodos que no encontraron la frontera del Objeto de Interés (OI) y la propia frontera del OI.

Esta forma de escoger los emplazamientos de las prospecciones nos puede dar una gran variedad de resultados diferentes, y es especialmente buena para detectar casos en los que la mejor solución es aplicar un algoritmo de inflado (por ser una zona amplia y más o menos uniforme) o cuando utilizar reclamos automáticos (por ser una zona irregular o asimétrica).

Ejemplo de ejecución de CAAR con BLA (prueba 600):

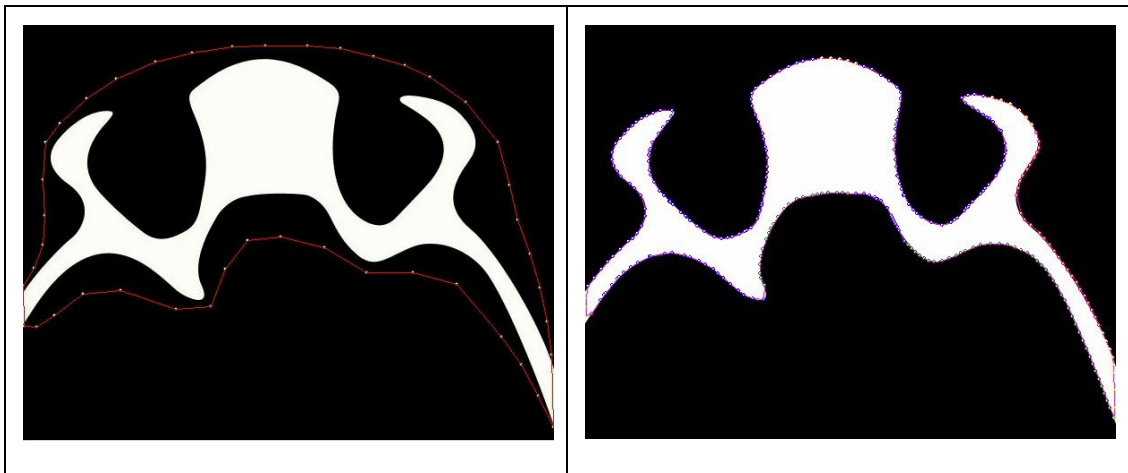


Figura 3.42. Izquierda: inicialización del contorno del CAAR. Derecha: contorno final resultante de ejecutar el CAAR utilizando BLA.

3.2.2.7. Análisis de las Cualidades de la Zona (ACZ)

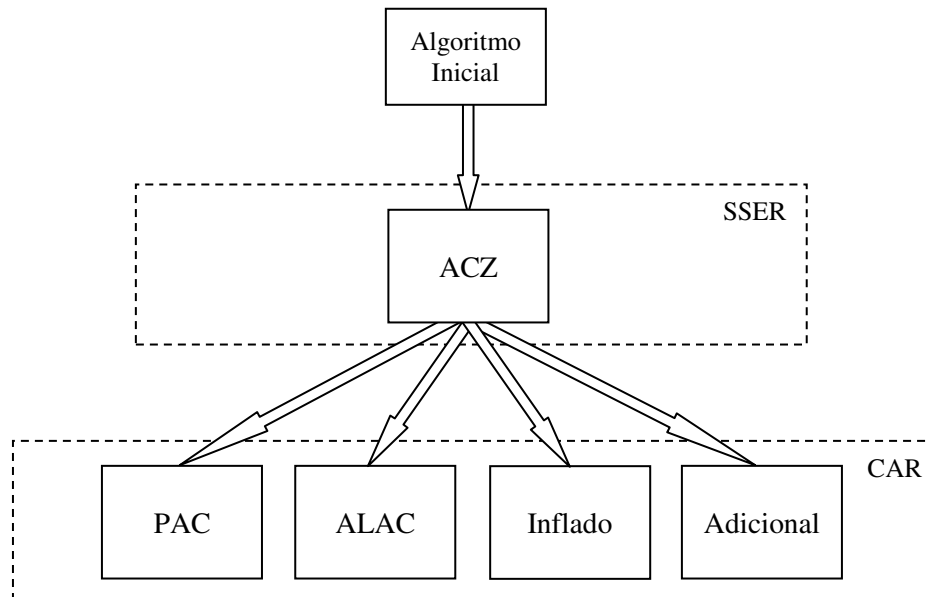


Figura 3.43. Esquema de CAAR utilizando ACZ

El Análisis de Cualidades de la Zona (ACZ) es el segundo SSER, es decir, el segundo método que se ha desarrollado para seleccionar qué algoritmo reconfigurador utilizar en cada situación en la segunda fase de los Contornos Activos Auto-Reconfigurables (CAAR). En este método, para seleccionar un contorno activo reconfigurador (CAR) u otro, se maneja información relativa a la zona comprendida entre el tramo de nodos a corregir y la frontera del objeto de interés (OI). Para la toma de decisiones se manejan características que se estudian sobre esa zona y que nos dan información sobre ese área, utilizando tanto características que tienen una clara traducción intuitiva (por ejemplo, área, perímetro, centroide, etc.) como características que describen la zona con medidas que no tienen una traducción tan directa a un concepto intuitivo (por ejemplo, los momentos estadísticos).

Una vez que tenemos toda esa información sobre la región en cuestión, la utilizamos para seleccionar que algoritmo utilizar como algoritmo reconfigurador.

En la figura anterior, el bloque *Algoritmo Inicial* representa cualquier algoritmo de contornos activos que queramos utilizar durante la primera fase del CAAR. En nuestro caso,

se han utilizado durante la primera fase, el algoritmo de Kass, Witkin y Terzopoulos [KASS87] y el de Williams y Shah [WILLIAMS92], obteniendo unos resultados de una calidad parecida, dejando algunos tramos de nodos que no se habían aproximado mucho a la frontera del objeto, lo que posteriormente se ha corregido durante la segunda fase del CAAR.

El bloque *Adicional* representa la posibilidad de añadir al CAAR cualquier algoritmo de contorno activo que se quiera para ser utilizado como otra opción de estrategia para mover los nodos que no se hayan aproximado correctamente al objeto de interés.

En este método en primer lugar se realizan los cálculos necesarios para obtener las características que nos interesan. Posteriormente se le va asignando una mayor probabilidad (de ser elegido) a cada algoritmo en función de las cualidades que se observan en esa zona, y esas cualidades se deducen a partir de las características calculadas previamente. Por ejemplo, basándonos en la compacidad de la zona, podemos deducir si se trata de una forma amplia y uniforme, o de una forma más alargada o asimétrica. Si calculamos la compacidad como $\text{perímetro}^2 / (\text{área} * 4 * \pi)$, la compacidad de una zona con forma de círculo tendría una compacidad cercana a 1, mientras que esta compacidad sube cuanto menos se parece a un círculo. Por tanto, si la compacidad es cercana a 1, el algoritmo reconfigurador más adecuado sería el algoritmo de inflado. Y si la compacidad es muy alta, entonces será más adecuado elegir el algoritmo de reclamos automáticos, por lo que se le asignará una probabilidad más alta.

Si el centroide del CAAR está alejado de la zona conflictiva (conjunto de nodos que no se adaptaron a la frontera del objeto), entonces se le asigna una mayor probabilidad al algoritmo de empuje.

Al final, cada algoritmo de los posibles (en nuestro caso Reclamos Automáticos, PAC y algoritmo de inflado) tendrá asignado un valor numérico proporcional a la probabilidad de ser elegido y finalmente se elegirá para ejecutar como CAR el que tenga asignada la mayor probabilidad.

Capítulo 4

Resultados y Discusión

4.1. Introducción.

En este apartado vamos a ver los resultados obtenidos y la discusión de los mismos. Primero mostraremos los resultados y posteriormente el análisis descriptivo de los resultados que se ha realizado.

4.2. Resultados.

En este apartado vamos a ver los resultados obtenidos con los diferentes métodos y, posteriormente se realizará un análisis de estos resultados.

Vamos a mostrar los resultados obtenidos para varias imágenes y, para cada imagen, se van a mostrar los resultados obtenidos después de aplicarle los algoritmos Clásico de Kass, Witkin y Terzopoulos [KASS88], Pushing Active Contours, Contornos Activos Auto-Reconfigurables, Balloon [COHEN91a], algoritmo de Caselles [CASELLES97], algoritmo de

Chan [CHAN01], algoritmo de Li[LI08], de Lankton [LANKTON08], de Bernard [BERNARD09] y de Shi [SHI08].

Para cada algoritmo aplicado, vamos a mostrar cómo queda el contorno final obtenido, para poder apreciar los resultados de forma visual y, posteriormente, los valores numéricos obtenidos para las características que describen de forma objetiva los contornos resultantes. Todo esto para poder apreciar los resultados de forma cuantitativa. Después se mostrará una tabla resumen donde se podrán observar todos estos resultados numéricos de forma conjunta, para poder compararlos más fácilmente.

En todas estas ejecuciones se ha unificado lo máximo posible la configuración de cada algoritmo, aunque en algunos casos hay parámetros que no pueden ser configurados por el usuario por las características propias de cada algoritmo. Como vimos en el capítulo de antecedentes, algunos de los algoritmos de contornos activos están basados en filosofías muy distintas, lo que hace que unos tengan parámetros muy diferentes a otros y, por tanto, no sea posible configurarlos de forma totalmente similar, pues hay parámetros de configuración de unos algoritmos que no tienen ningún parámetro equivalente en otro. Por todo esto, aunque los parámetros con los que se ha ejecutado cada algoritmo son muy similares, en algunos casos muy concretos no son exactamente iguales pues depende del algoritmo en concreto.

Para este estudio, hemos comparado las características del contorno obtenido con cada método y las hemos comparado con las características del contorno óptimo. Esta comparación se ha realizado calculando el error relativo entre el valor obtenido y el valor óptimo esperado.

$$\text{Error relativo} = (\text{ABS}(\text{valor óptimo} - \text{valor obtenido}) / \text{valor óptimo}) * 100$$

Se ha tomado valor absoluto del error relativo y se ha multiplicado por 100 para obtener el resultado en tantos por ciento (%). Así hemos obtenido un vector que contiene el error relativo de cada característica, para ese método y para esa prueba concreta.

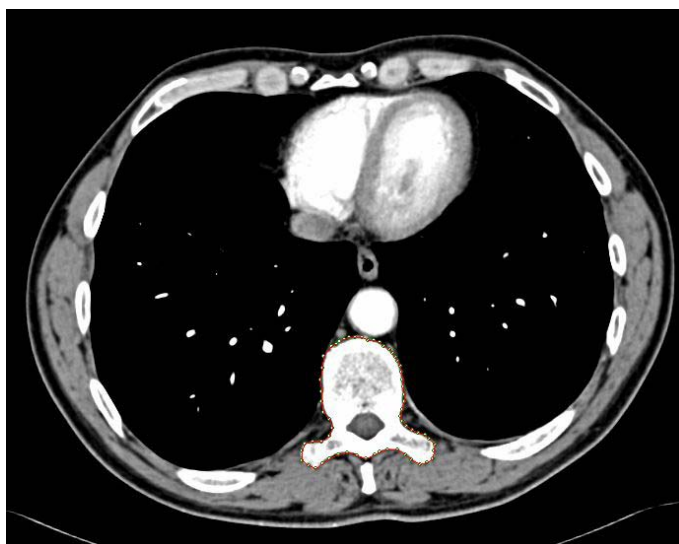
Después se ha hallado la media del vector de error, que hemos indicado en las tablas como "Media error", que indicará el error medio de ese método para ese contorno, así como la desviación estándar de dicho vector.

Posteriormente se ha realizado el proceso anterior para cada uno de los métodos estudiados, por lo que tenemos un dato de error medio y de desviación estándar para cada uno de los métodos.

4.2.1. Ejemplos de ejecución de todos los métodos para la imagen 1

1. Contorno óptimo para la imagen 1

(prueba 508) El contorno óptimo fue definido manualmente por un experto y posteriormente se obtuvieron de forma automática las características que definen de forma objetiva dicho contorno.



Área	Cent_X	Centr_Y	LEMAY	LEMEN	Exc	Per
10449	364,7	412,17	391,98	274,83	0,7	445

mu_02	mu_20	mu_11	mu_12	mu_21	mu_22
51618506	98054224	25507241	768377949	46421341	221761248

Cent_X, Cent_Y: Coordenadas X e Y del centroide del contorno.

LEMAY: Longitud del Eje Mayor.

LEMEN: Longitud del Eje Menor.

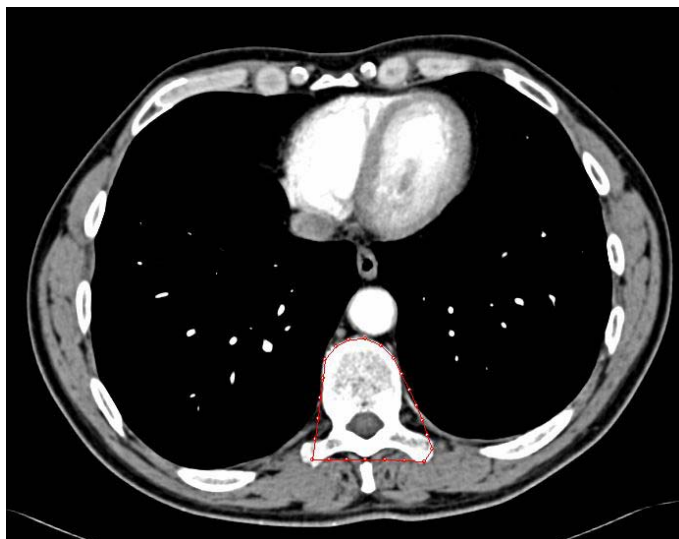
Exc: Excentricidad.

Per: Perímetro del contorno.

mu_02, mu_20, mu_11, mu_12, mu_21, mu_22: momentos estadísticos centrales.

2-Resultado del algoritmo Clásico para la imagen 1.

(prueba 510) Resultado de aplicar el algoritmo clásico de Kass [KASS88] a la imagen 1.



	Área	Cent_X	Centr_Y	LEMAY	LEMEN	Excentr	Perímetro
Óptimo	10449	364,7	412,17	391,98	274,83	0,7	445
Clásico	7392	362,99	410,98	454,81	307,34	0,67	391
Error Relativo (%)	8,6133	0,7184	0,0534	4,5283	3,9916	0,0000	11,9101

	mu_02	mu_20	mu_11	mu_12	mu_21	mu_22
Óptimo	51618506	98054224	-25507241	768377949	-46421341	221761248
Clásico	45944273	93264980	-25964675	788175297	-9948206	1005101916
Error Relativo (%)	1,4470	0,2055	2,0800	16,2345	372,1230	298,8514

Mostramos ahora el error medio y la desviación estándar de dos formas diferentes: la primera, calculándolos respecto a todas las características mostradas anteriormente. La segunda, respecto a cinco características (área, centroide_X, centroide_Y, Excentricidad y Perímetro) debido a que, como se explicará más en profundidad después de mostrar estos resultados, después de una primera fase de análisis, se observó que era más representativo el subconjunto formado por esas cinco características.

Media Error	Desv. Estand.
55,4428	125,2830

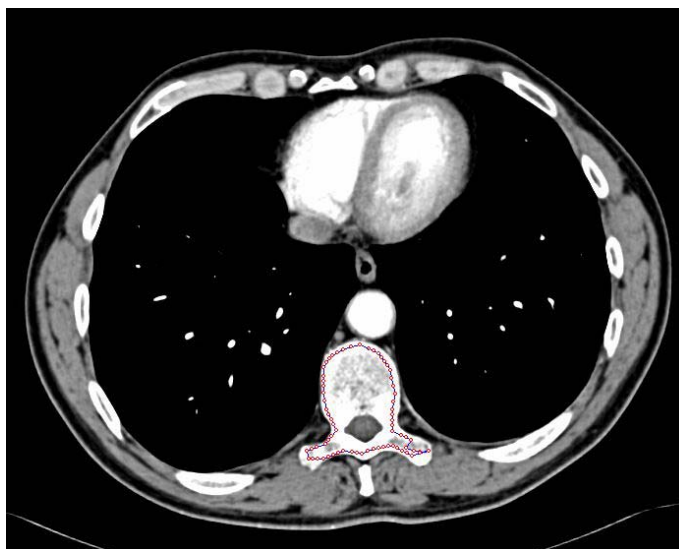
Con todas las caract.

Media Error	Desv. Estand.
4,2590	5,6094

Con cinco caract.

3-Resultado del algoritmo Balloon con la imagen 1

Prueba 515. Resultado de aplicar el algoritmo Balloon sobre la imagen 1.



Nº prueba	Método	Área	Cent_X	Centr_Y	LEMAY	LEMEN	Excentr	Perímetro
	Óptimo	10449	364,7	412,17	391,98	274,83	0,7	445
515	Balloon	7392	362,99	410,98	454,81	307,34	0,67	391
515	Error Relativo (%)	29,2564	0,4689	0,2887	16,0289	11,8291	4,2857	12,1348

Nº prueba	Método	mu_02	mu_20	mu_11	mu_12	mu_21	mu_22
	Óptimo	51618506	98054224	-25507241	768377949	-46421341	221761248
515	Balloon	45944273	93264980	-25964675	788175297	-9948206	1005101916
515	Error Relativo (%)	10,9926	4,8843	1,7933	2,5765	78,5698	353,2360

Media Error	Desv. Estand.
40,4881	96,2685

515

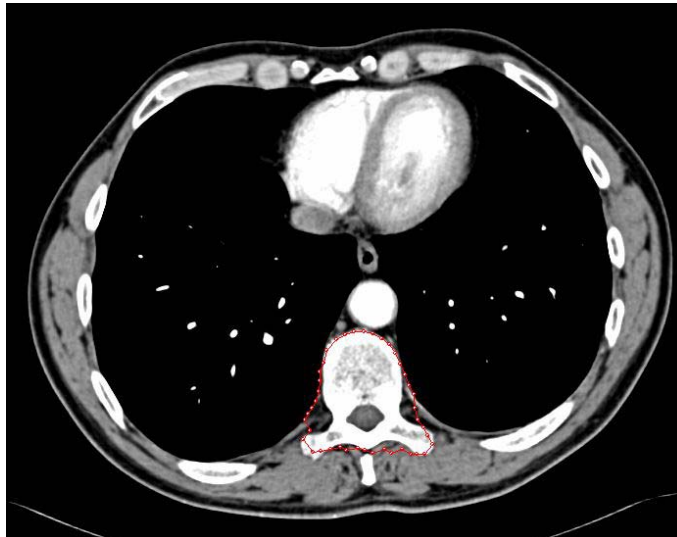
Media Error	Desv. Estand.
9,2869	12,1514

Con todas las caract.

Con cinco caract.

4-Resultado del algoritmo Pushing Active Contours con la imagen 1

Prueba 514. Resultado de aplicar el algoritmo Pushing Active Contours sobre la imagen 1.



	Área	Cent_X	Centr_Y	LEMAY	LEMEN	Excentr	Perímetro
Óptimo	10449	364,7	412,17	391,98	274,83	0,7	445
Pushing Active Contours	11462	365,31	411,22	374,18	263,34	0,7	395
Error Relativo (%)	9,6947	0,1673	0,2305	4,5410	4,1808	0,0000	11,2360

	mu_02	mu_20	mu_11	mu_12	mu_21	mu_22
Óptimo	51618506	98054224	-25507241	768377949	-46421341	221761248
Pushing Active Contours	50782109	99204952	-25310737	843739417	-199157343	680270797
Error Relativo (%)	1,6203	1,1736	0,7704	9,8079	329,0211	206,7582

Media Error	Desv. Estand.
44,5540	102,2893

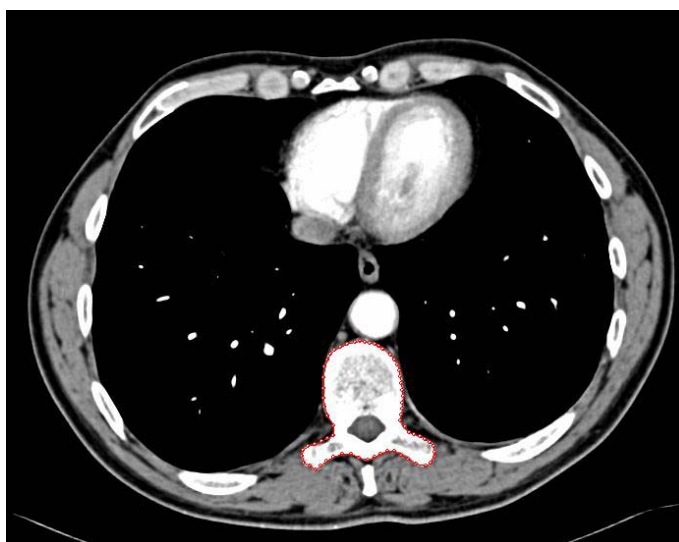
Con todas las caract.

Media Error	Desv. Estand.
4,2657	5,6863

Con cinco caract.

5-Resultado del algoritmo CAAR con la imagen 1

Prueba 509b. Resultado de aplicar el algoritmo Contornos Activos Auto-Reconfigurables (CAAR) sobre la imagen 1.



	Área	Cent_X	Centr_Y	LEMAY	LEMEN	Excentr	Perimetro
Óptimo	10449	364,7	412,17	391,98	274,83	0,7	445
Auto-Reconf	9904	364,28	412,67	400,76	278,97	0,69	436
Error Relativo (%)	5,2158	0,1152	0,1213	2,2399	1,5064	1,4286	2,0225

	mu_02	mu_20	mu_11	mu_12	mu_21	mu_22
Óptimo	51618506	98054224	-25507241	768377949	-46421341	221761248
Auto-Reconf	50415589	97178555	-25623589	745693948	-9250833	563741304
Error Relativo (%)	2,3304	0,8930	0,4561	2,9522	80,0720	154,2109

Media Error	Desv. Estand.
19,5049	45,9189

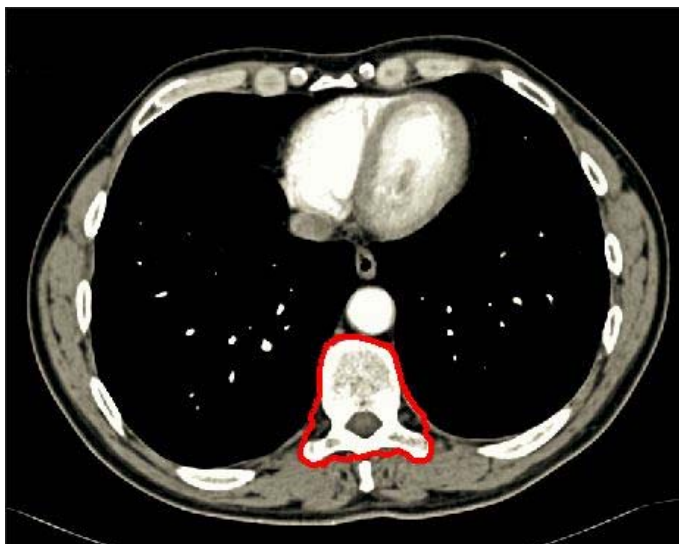
Con todas las caract.

Media Error	Desv. Estand.
1,7807	2,0922

Con cinco caract.

6- Resultado del algoritmo de Caselles con la imagen 1

Prueba 1005. Resultado de aplicar el algoritmo de Caselles sobre la imagen 1.



Nº prueba	Método	Área	Cent_X	Centr_Y	LEMai	LEMen	Excentr	Perímetro
	Óptimo	10449	364,7	412,17	391,98	274,83	0,7	445
1005	Caselles	11940	366,094	407,922	138,36022	123,7769	0,446869	489,688384
1005	Error Relativo (%)	14,2693	0,38213	1,03068	64,702224	54,96236	36,16152	10,0423334

Nº prueba	Método	mu_02	mu_20	mu_11	mu_12	mu_21	mu_22
	Óptimo	51618506	98054224	-25507241	768377949	-46421341	221761248
1005	Caselles	11723615,3	13993366,1	864085,361	169818112,7	-47483708,96	13307927590
1005	Error Relativo (%)	77,2879608	85,7289513	103,387608	77,89914287	2,288533543	5901,015827

Media Error	Desv. Estand.
494,5507	1624,8423

Con todas las caract.

Media Error	Desv. Estand.
12,3772	14,5546

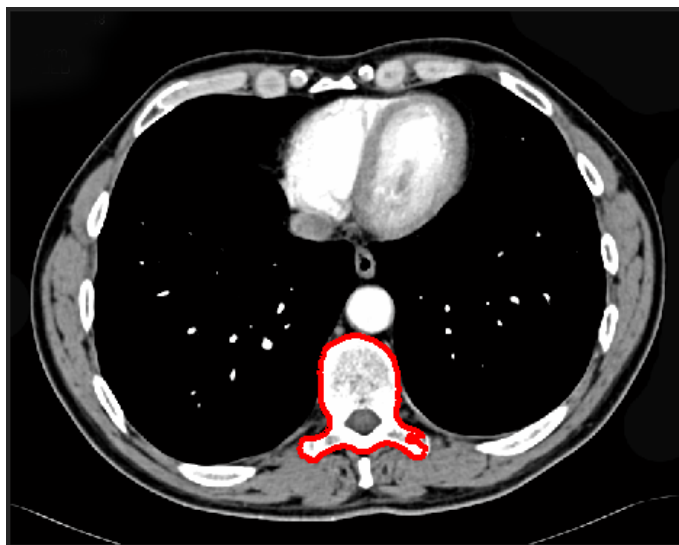
Con cinco caract.

En este algoritmo existen pocas opciones de configuración, la más importante es que se puede seleccionar que el contorno evolucione hacia dentro de la curva o hacia fuera. Para cambiar esta configuración, basta poner el término de propagación a -1 si queremos que se mueva hacia dentro de la curva o a 1 si queremos que se mueva hacia fuera. Se han realizado numerosas pruebas, pero sólo mostramos la que ha obtenido mejores resultados. En este caso

el mejor resultado fue inicializando el contorno fuera del objeto de interés y configurando el parámetro específico a -1.

7- Resultado del algoritmo de Lankton con la imagen 1

Prueba 1044. Resultado de aplicar el algoritmo de Lankton sobre la imagen 1.



Nº prueba	Método	Área	Cent_X	Centr_Y	LEMAY	LEMEN	Excentr	Perímetro
	Óptimo	10449	364,7	412,17	391,98	274,83	0,7	445
1044	Lankton	9442	364,054	403,912	137,05436	106,4221	0,630122	530,27417
1044	Error Relativo (%)	9,63729	0,17707	2,00354	65,035369	61,27711	9,9825	19,1627348

Nº prueba	Método	mu_02	mu_20	mu_11	mu_12	mu_21	mu_22
	Óptimo	51618506	98054224	-25507241	768377949	-46421341	221761248
1044	Lankton	6748334,24	11018499,9	533154,062	113130193,8	19791347,07	9772004281
1044	Error Relativo (%)	86,9265216	88,7628504	102,090207	85,27675164	57,36584372	4306,542788

Media Error	Desv. Estand.
376,4800	1181,4079

Con todas las caract.

Media Error	Desv. Estand.
8,1926	7,5529

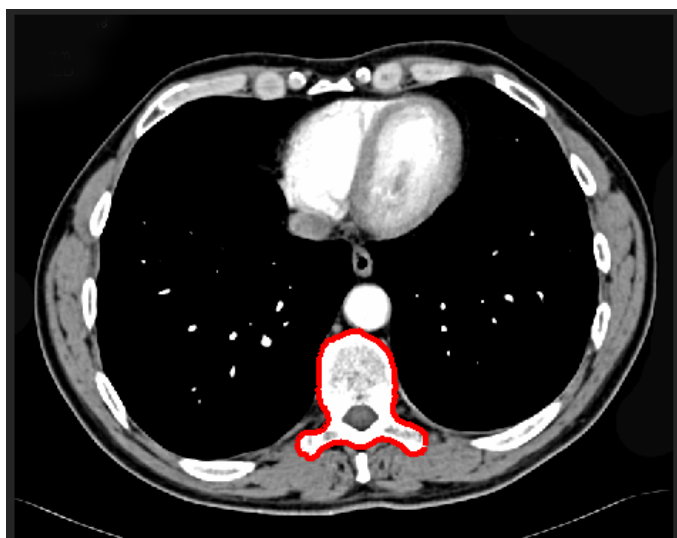
Con cinco caract.

La prueba que se muestra es la que obtuvo mejor resultado de todas las realizadas. Durante las pruebas, se observó que los mejores resultados se obtenían inicializando el

contorno dentro del objeto y aumentando el parámetro específico del radio (a mayor radio, mejor resultado). Los resultados que se muestran corresponden a una ejecución asignándole al parámetro del radio un valor de 26.

8-Resultado del algoritmo de Chan con la imagen 1

Prueba 1038. Resultado de aplicar el algoritmo de Chan sobre la imagen 1.



Nº prueba	Método	Área	Cent_X	Centr_Y	LEMAY	LEMEN	Excentr	Perímetro
	Óptimo	10449	364,7	412,17	391,98	274,83	0,7	445
1038	Chan	9896	365,348	405,374	141,82673	112,6111	0,607912	525,002092
1038	Error Relativo (%)	5,29237	0,17755	1,64893	63,817866	59,02518	13,15549	17,9779982

Nº prueba	Método	mu_02	mu_20	mu_11	mu_12	mu_21	mu_22
	Óptimo	51618506	98054224	-25507241	768377949	-46421341	221761248
1038	Chan	8099097,9	12183627,9	1055353,24	161114119,4	-17599347,9	12410807100
1038	Error Relativo (%)	84,3097011	87,5746017	104,137465	79,03191787	62,08780806	5496,472428

Media Error	Desv. Estand.
467,2853	1511,5222

Con todas las caract.

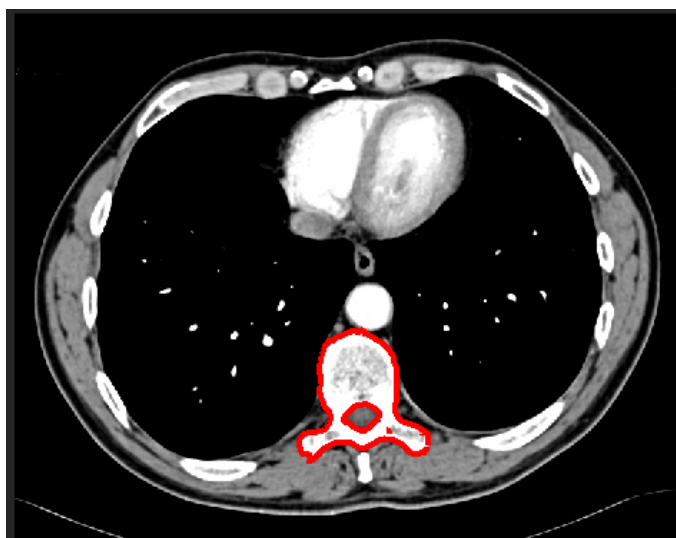
Media Error	Desv. Estand.
7,6505	7,6548

Con cinco caract.

El algoritmo de Chan es uno de los que mejores resultados ha arrojado. En las pruebas en las que se inicializó el contorno fuera del objeto los resultados no fueron nada buenos. Al inicializar el contorno dentro del objeto los resultados fueron bastante mejores.

9-Resultado del algoritmo de Shi con la imagen 1

Prueba 1013. Resultado de aplicar el algoritmo de Shi sobre la imagen 1.



Nº prueba	Método	Área	Cent_X	Centr_Y	LEMai	LEMen	Excentr	Perímetro
	Óptimo	10449	364,7	412,17	391,98	274,83	0,7	445
1013	Shi	9161	365,164	403,764	144,91715	116,2583	0,597002	531,587879
1013	Error Relativo (%)	12,3265	0,12734	2,03945	63,029453	57,69812	14,71395	19,4579502

Nº prueba	Método	mu_02	mu_20	mu_11	mu_12	mu_21	mu_22
	Óptimo	51618506	98054224	-25507241	768377949	-46421341	221761248
1013	Shi	7989638,43	11771953,8	1007559,42	171450658,1	13631510,26	12838194665
1013	Error Relativo (%)	84,5217558	87,994445	103,950092	77,68667642	70,63525102	5689,196616

Media Error	Desv. Estand.
483,3367	1564,5750

Con todas las caract.

Media Error	Desv. Estand.
9,7330	8,3302

Con cinco caract.

En esta prueba se puede observar que también encuentra un pequeño objeto dentro del objeto a segmentar, lo cual se puede considerar un fallo en el resultado, aunque esto no se ve reflejado en las características calculadas para esta prueba.

10-Resultado del algoritmo de Bernard con la imagen 1

Prueba 1012. Resultado de aplicar el algoritmo de Bernard sobre la imagen 1.



Como ya mencionamos en el capítulo 3, el método de Bernard intenta encontrar todos los contornos que haya en toda la imagen, hecho que convierte a este algoritmo en un método no apropiado para segmentar un único objeto en una imagen donde se encuentran varios objetos. A pesar de haber inicializado el contorno cerca de la frontera del objeto de interés, sucedió, al igual que con el método de Li para esta imagen, que el algoritmo de Bernard era excesivamente lento y no llegaba a ejecutarse el conjunto de instrucciones que generan el vector de características, por lo que no pudimos obtener los descriptores en ninguna de las pruebas realizadas con este algoritmo y para esta imagen.

11-Resultado del algoritmo de Li con la imagen 1

Prueba 1114. Resultado de aplicar el algoritmo de Li sobre la imagen 1.



En esta prueba el algoritmo no llegaba a terminar y no se ejecutaba la instrucción en la que comenzaba a generarse el vector de características, por lo que no se pudieron obtener medidas objetivas en ninguna de las pruebas realizadas con este algoritmo para esta imagen. En cualquier caso, se puede apreciar visualmente que los resultados fueron poco precisos, por lo que el error relativo de cada característica habría salido muy elevado si se hubiera generado el vector de características y habrían sido de los peores resultados que se habrían obtenido en comparación con el resto de métodos.

Desde un punto de vista estadístico, esta prueba habría que descartarla, pero no debemos perder la oportunidad de observar los resultados que ofrece este algoritmo en este tipo de imágenes.

4.2.2. Ejemplos de ejecución de todos los métodos para la imagen 2

1-Contorno óptimo para la imagen 2

En esta imagen 2 vamos a segmentar una vértebra que en este caso se encuentra en sentido vertical. El contorno óptimo fue definido manualmente por un experto y, a partir de ese contorno, se obtuvieron las características que definen de forma objetiva dicho contorno óptimo.

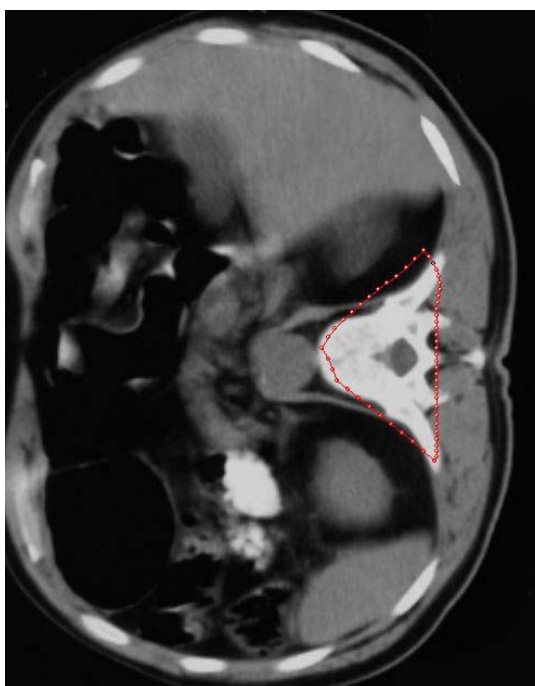


	Área	Cent_X	Centr_Y	LEMAY	LEMEN	Excentr	Perímetro
Óptimo	11158	369,74	325,2	343,7	255,82	0,74	556

	mu_02	mu_20	mu_11	mu_12	mu_21	mu_22
Óptimo	86951043	41073922	-18372473	-1571070600	1066542794	405878643

2-Resultado del algoritmo Clásico con la imagen 2

(prueba 519c) Resultado de aplicar el algoritmo clásico de Kass, Witkin y Terzopoulos [KASS88].



	Área	Cent_X	Centr_Y	LEMAY	LEMEN	Excentr	Perímetro
Óptimo	11158	369,74	325,2	343,7	255,82	0,74	556
Clásico	12144	366,62	321,45	331,24	249,11	0,75	413
Error Relativo (%)	8,8367	0,8438	1,1531	3,6253	2,6229	1,3514	25,7194

	mu_02	mu_20	mu_11	mu_12	mu_21	mu_22
Óptimo	86951043	41073922	-18372473	-1571070600	1066542794	405878643
Clásico	92065114	38315060	-18088105	-1653412981	1062614578	832546409
Error Relativo (%)	5,8816	6,7168	1,5478	5,2412	0,3683	105,1220

Media Error	Desv. Estand.
13,0023	28,4693

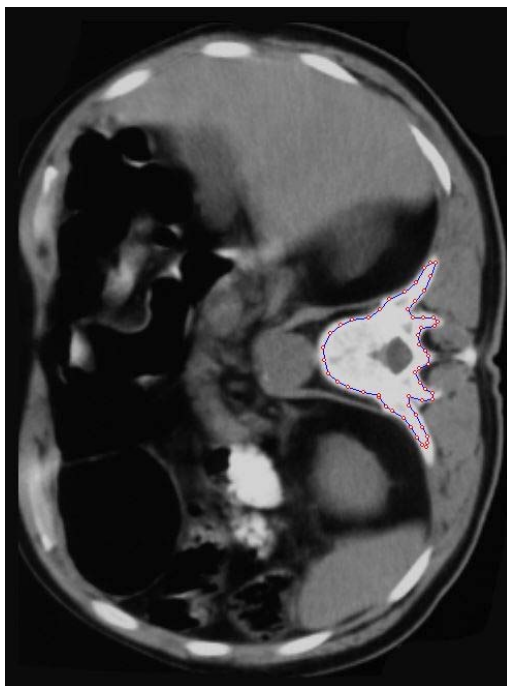
Con todas las caract.

Media Error	Desv. Estand.
7,5809	10,6782

Con cinco caract.

3-Resultado del algoritmo Balloon con la imagen 2

Prueba 531. Resultado de aplicar el algoritmo Balloon sobre la imagen 2.



Nº prueba		Área	Cent_X	Centr_Y	LEMai	LEMen	Excentr	Perímetro
	Óptimo	11158	369,74	325,2	343,7	255,82	0,74	556
531	Balloon	8595	414,81	378,5	568,49	392,69	0,69	490
531	Error Relativo (%)	22,9701	12,1896	16,3899	65,4030	53,5025	6,7568	11,8705

Nº prueba		mu_02	mu_20	mu_11	mu_12	mu_21	mu_22
	Óptimo	86951043	41073922	-18372473	-1571070600	1066542794	405878643
531	Balloon	171638770	84815507	-45386201	-413634540	569292532	293512499
531	Error Relativo (%)	97,3970	106,4948	147,0337	73,6718	46,6226	27,6847

Media Error	Desv. Estand.
52,9221	43,4700

Con todas las caract.

Media Error	Desv. Estand.
14,0354	6,0507

Con cinco caract.

4-Resultado del algoritmo Pushing Active Contours con la imagen 2

Prueba 530c. Resultado de aplicar el algoritmo Pushing Active Contours sobre la imagen 2.



Nº prueba		Área	Cent_X	Centr_Y	LEMay	LEMen	Excentr	Perímetro
	Óptimo	11158	369,74	325,2	343,7	255,82	0,74	556
530c	Push	12961	417,4	379,43	475,4	335,24	0,7	511
530c	Error Relativo (%)	16,1588	12,8901	16,6759	38,3183	31,0453	5,4054	8,0935

Nº prueba		mu_02	mu_20	mu_11	mu_12	mu_21	mu_22
	Óptimo	86951043	41073922	-18372473	-1571070600	1066542794	405878643
530c	Push	182630156	91214724	-45881271	-274512384	489613949	1262029728
530c	Error Relativo (%)	110,0379	122,0745	149,7283	82,5270	54,0934	210,9377

Media Error	Desv. Estand.
65,9989	64,7216

Con todas las caract.

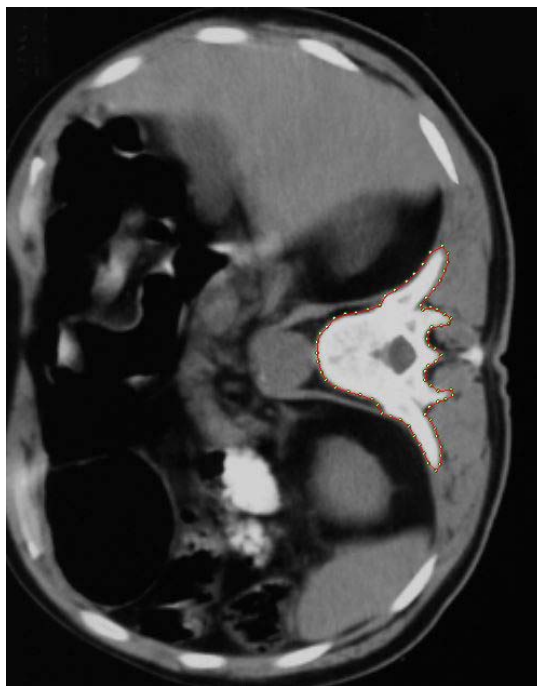
Media Error	Desv. Estand.
11,8448	4,9644

Con cinco caract.

En este caso, a pesar de que el resultado no ha sido muy erróneo, el contorno se ha detenido en algunos cambios de intensidad de la imagen.

5-Resultado del algoritmo Aut-Rec con la imagen 2

Prueba 520. Resultado de aplicar el algoritmo Auto-Reconfigurable sobre la imagen 2.



	Área	Cent_X	Centr_Y	LEMAY	LEMEN	Excentr	Perímetro
Óptimo	11158	369,74	325,2	343,7	255,82	0,74	556
Aut.Rec.	11811	370,96	324,9	339,56	252,19	0,74	556
Error Relativo (%)	5,8523	0,3300	0,0923	1,2045	1,4190	0,0000	0,0000

	mu_02	mu_20	mu_11	mu_12	mu_21	mu_22
Óptimo	86951043	41073922	-18372473	-1571070600	1066542794	405878643
Aut.Rec.	90236046	41829894	-19083091	-1506778779	1134136961	1318698165
Error Relativo (%)	3,7780	1,8405	3,8678	4,0922	6,3377	224,8996

Media Error	Desv. Estand.
19,5165	61,7492

Con todas las caract.

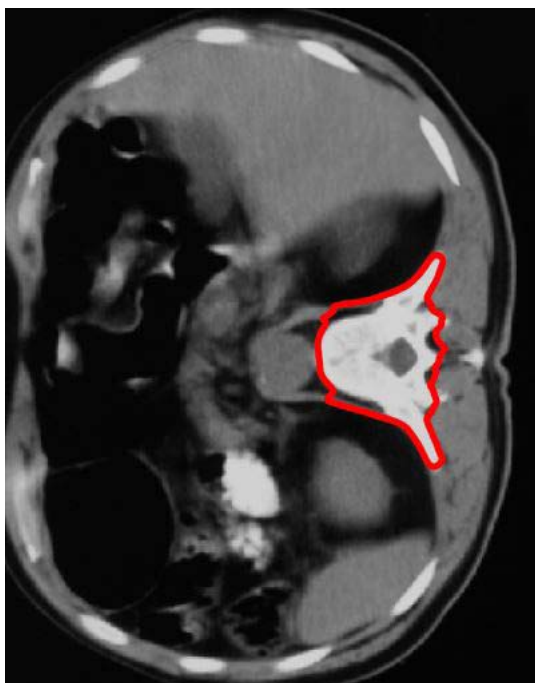
Media Error	Desv. Estand.
1,2549	2,5736

Con cinco caract.

En esta prueba, al ejecutar el algoritmo Auto-Reconfigurable se utilizó Búsqueda Longitudinal en Abanico.

6-Resultado del algoritmo de Caselles con la imagen 2

Prueba 522e. Resultado de aplicar el algoritmo de Caselles sobre la imagen 2.



	Área	Cent_X	Centr_Y	LEMAY	LEMEN	Excentr	Perímetro
Óptimo	11158	369,74	325,2	343,7	255,82	0,74	556
Caselles	12404	355,1	323,156	153,87249	121,8554	0,610619	557,818
Error Relativo (%)	11,1669	3,9596	0,6286	55,2306	52,3667	17,4840	0,3270

	mu_02	mu_20	mu_11	mu_12	mu_21	mu_22
Óptimo	86951043	41073922	-18372473	-1571070600	1066542794	405878643
Caselles	11541378,4	18323443,1	458962,1738	11972914,41	297384599,6	18220550588
Error Relativo (%)	86,7266	55,3891	102,4981	99,2379	72,1170	4389,1622

Media Error	Desv. Estand.
380,4842	1205,0291

Con todas las caract.

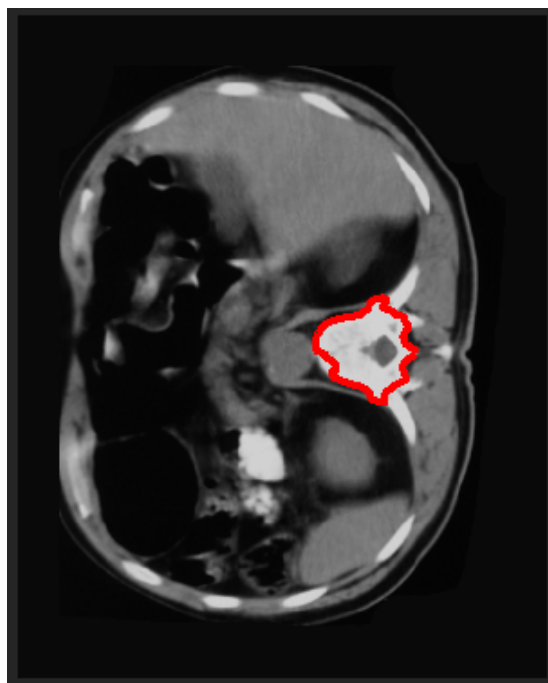
Media Error	Desv. Estand.
6,7132	7,4370

Con cinco caract.

El algoritmo de Caselles necesitó un número muy elevado de iteraciones (1200) para poder ajustarse medianamente bien.

7-Resultado del algoritmo de Lankton con la imagen 2

Prueba 1021. Resultado de aplicar el algoritmo de Lankton sobre la imagen 2.



Nº prueba		Área	Cent_X	Centr_Y	LEMay	LEMen	Excentr	Perímetro
	Óptimo	11158	369,74	325,2	343,7	255,82	0,74	556
1021	Lankton	7742	397,844	378,461	104,79694	99,73643	0,306995	418,617316
1021	Error Relativo (%)	30,6148	7,6009	16,3779	69,5092	61,0130	58,5143	24,7091

Nº prueba		mu_02	mu_20	mu_11	mu_12	mu_21	mu_22
	Óptimo	86951043	41073922	-18372473	-1571070600	1066542794	405878643
1021	Lankton	5144286,89	4981807,64	236871,644	12117950,24	25845180,86	2112534027
1021	Error Relativo (%)	94,0837	87,8711	101,2893	99,2287	97,5767	420,4842

Media Error	Desv. Estand.
89,9133	104,7862

Con todas las caract.

Media Error	Desv. Estand.
27,5634	19,3586

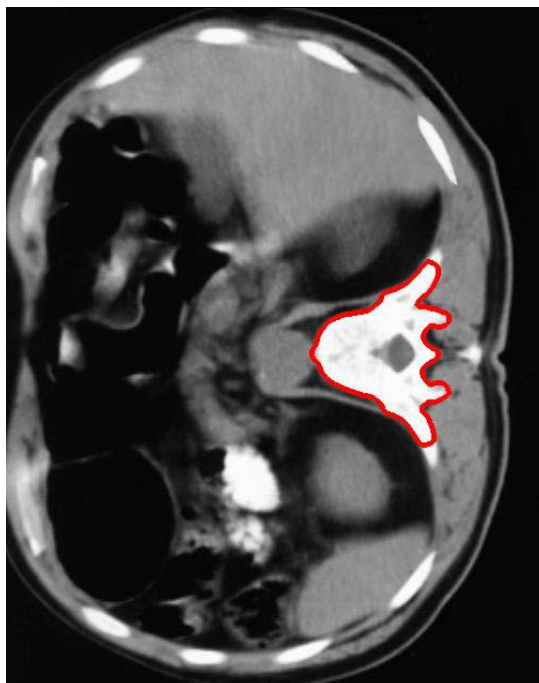
Con cinco caract.

Hemos escogido la prueba 1021 como la que presenta resultados más favorables, pero es probable que si siguiésemos haciendo muchas más pruebas, con todas las combinaciones posibles de todos los parámetros, se consiguiese que se ajustase un poco más. Además, como

se puede comprobar en las pruebas incluidas en el capítulo 3, esta fue la mejor debido a que la inicialización se estableció cerca de la frontera del objeto de interés. En todas las demás pruebas realizadas con la imagen 2, los resultados fueron peores para este algoritmo.

8-Resultado del algoritmo de Chan con la imagen 2

Prueba 521. Resultado de aplicar el algoritmo de Chan sobre la imagen 2.



	Área	Cent_X	Centr_Y	LEMAY	LEMEN	Excentr	Perímetro
Óptimo	11158	369,74	325,2	343,7	255,82	0,74	556
Chan y Vese	10702	357,078	323,008	140,34957	121,6917	0,498201	570,904
Error Relativo (%)	4,0868	3,4245	0,6742	59,1651	52,4307	32,6756	2,6806

	mu_02	mu_20	mu_11	mu_12	mu_21	mu_22
Óptimo	86951043	41073922	-18372473	-1571070600	1066542794	405878643
Chan y Vese	9907539,23	13171469,4	101292,5715	9625066,229	234139848,8	10940318391
Error Relativo (%)	88,6056	67,9323	100,5513	99,3874	78,0468	2595,4654

Media Error	Desv. Estand.
245,0097	707,2218

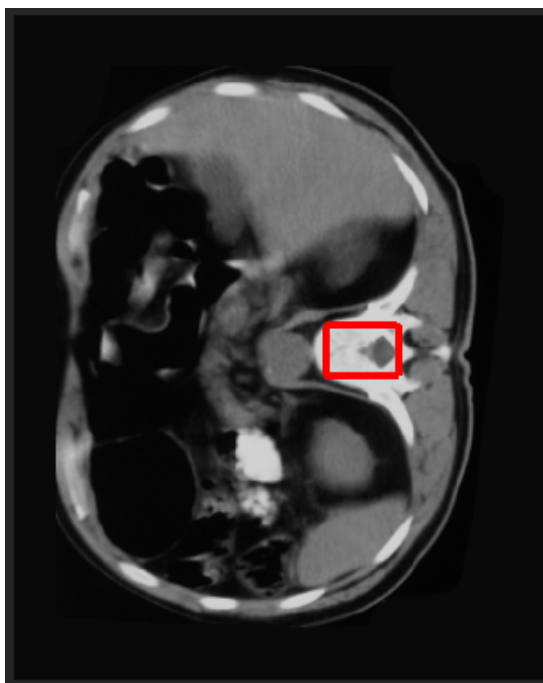
Con todas las caract.

Media Error	Desv. Estand.
8,7083	13,4591

Con cinco caract.

9-Resultado del algoritmo de Shi con la imagen 2

Prueba 1022. Resultado de aplicar el algoritmo de Shi sobre la imagen 2.



Nº prueba		Área	Cent_X	Centr_Y	LEMai	LEMen	Excentr	Perímetro
	Óptimo	11158	369,74	325,2	343,7	255,82	0,74	556
1022	Shi	4674	394,5	380	94,685444	65,81793	0,718892	274
1022	Error Relativo (%)	58,1108	6,6966	16,8512	72,4511	74,2718	2,8525	50,7194

Nº prueba		mu_02	mu_20	mu_11	mu_12	mu_21	mu_22
	Óptimo	86951043	41073922	-18372473	-1571070600	1066542794	405878643
1022	Shi	2618608,5	1265096	0	0	0	708770034
1022	Error Relativo (%)	96,9884	96,9200	100,0000	100,0000	100,0000	74,6261

Media Error	Desv. Estand.
65,4221	36,2633

Con todas las caract.

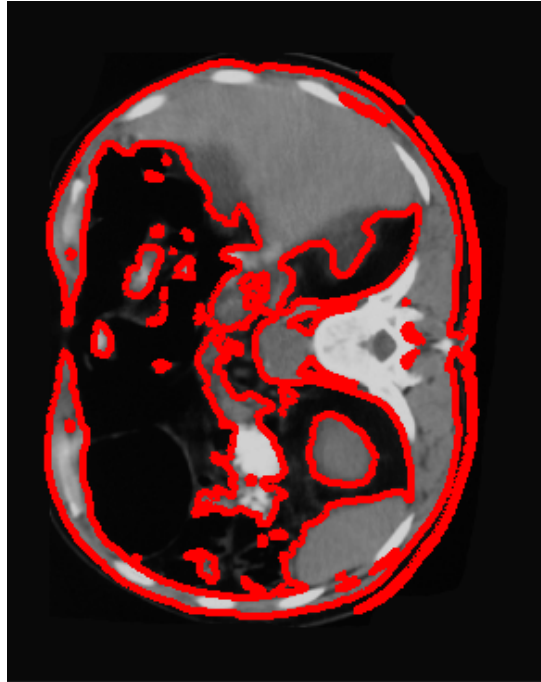
Media Error	Desv. Estand.
27,0461	25,6360

Con cinco caract.

Pueden observarse resultados muy poco satisfactorios en esta prueba, pero ha sido la que ha obtenido mejores resultados de todas las realizadas con el método de Shi para la imagen 2.

10-Resultado del algoritmo de Bernard con la imagen 2

Prueba 1116. Resultado de aplicar el algoritmo de Bernard sobre la imagen 2.



Al igual que sucedía en las pruebas de funcionamiento del método de Bernard registradas en el capítulo 3, en esta prueba el método intenta encontrar todos los contornos que haya en la imagen. Esto nos demuestra una vez más que el método de Bernard no es un buen método para segmentar imágenes médicas donde solo se desee segmentar un objeto.

11-Resultado del algoritmo de Li con la imagen 2

Prueba 1115. Resultado de aplicar el algoritmo de Li sobre la imagen 2.



En esta prueba, al igual que sucedió para la realizada con el algoritmo de Li para la imagen 1, el algoritmo no llegó a terminar, por lo que no llegaron a ejecutarse las líneas de código que generan el vector de características, por lo que no se pudieron obtener las características y, por tanto, medidas objetivas. En cualquier caso, es evidente que los resultados no fueron buenos y que, de haberse generado el vector de características, se habrían generado unos valores muy alejados de los valores del contorno óptimo.

Desde un punto de vista estadístico, esta prueba sería nula, al no generar datos con los que poder trabajar, por lo que no nos aporta información cuantitativamente hablando, pero si nos aporta información desde un punto de vista cualitativo, en este caso concreto podemos deducir que no es un buen método para trabajar con este tipo de imágenes.

4.2.3. Estudio conjunto de los resultados:

A continuación vamos a mostrar una tabla resumen donde podemos ver en cada fila el algoritmo utilizado y en cada columna la característica estudiada del contorno:

Resultados para la imagen 1

Tabla comparativa de todos los métodos. Valores absolutos:

Nº prueba	Método	Área	Cent_X	Centr_Y	LEMay	LEMen	Excentr	Perímetro
	Óptimo	10449	364,7	412,17	391,98	274,83	0,7	445
510	Clásico	7392	362,99	410,98	454,81	307,34	0,67	391
515	Balloon	7392	362,99	410,98	454,81	307,34	0,67	391
514	Pushing A. C.	11462	365,31	411,22	374,18	263,34	0,7	395
509b	Auto-Reconf	9904	364,28	412,67	400,76	278,97	0,69	436
1005	Caselles	11940	366,094	407,922	138,36022	123,7769	0,446869	489,688384
1044	Lankton	9442	364,054	403,912	137,05436	106,4221	0,630122	530,27417
1038	Chan	9896	365,348	405,374	141,82673	112,6111	0,607912	525,002092
1013	Shi	9161	365,164	403,764	144,91715	116,2583	0,597002	531,587879
1012	Bernard	*	*	*	*	*	*	*
1114	Li	*	*	*	*	*	*	*

Nº prueba	Método	mu_02	mu_20	mu_11	mu_12	mu_21	mu_22
	Óptimo	51618506	98054224	-25507241	768377949	-46421341	221761248
510	Clásico	45944273	93264980	-25964675	788175297	-9948206	1005101916
515	Balloon	45944273	93264980	-25964675	788175297	-9948206	1005101916
514	Pushing A. C.	50782109	99204952	-25310737	843739417	-199157343	680270797
509b	Auto-Reconf	50415589	97178555	-25623589	745693948	-9250833	563741304
1005	Caselles	11723615,3	13993366,1	864085,361	169818112,7	-	13307927590
1044	Lankton	6748334,24	11018499,9	533154,062	113130193,8	47483708,96	9772004281
1038	Chan	8099097,9	12183627,9	1055353,24	161114119,4	19791347,07	12410807100
1013	Shi	7989638,43	11771953,8	1007559,42	171450658,1	-17599347,9	12838194665
1012	Bernard	*	*	*	*	-	*
1114	Li	*	*	*	*	13631510,26	*

Tabla comparativa de todos los métodos. Diferencias Relativas:

Nº prueba	Método	Área	Cent_X	Centr_Y	LEMAY	LEMEN	Excentr	Perímetro
	Óptimo	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
510	Clásico	8,6133	0,7184	0,0534	4,5283	3,9916	0	11,9101
515	Balloon	29,2564	0,4689	0,2887	16,0289	11,8291	4,2857	12,1348
514	Pushing A. C.	9,6947	0,1673	0,2305	4,541	4,1808	0	11,236
509b	Auto-Reconf	5,2158	0,1152	0,1213	2,2399	1,5064	1,4286	2,0225
1005	Caselles	14,2693	0,38213	1,03068	64,702224	54,96236	36,16152	10,0423334
1044	Lankton	9,63729	0,17707	2,00354	65,035369	61,27711	9,9825	19,1627348
1038	Chan	5,29237	0,17755	1,64893	63,817866	59,02518	13,15549	17,9779982
1013	Shi	12,3265	0,12734	2,03945	63,029453	57,69812	14,71395	19,4579502
1012	Bernard	*	*	*	*	*	*	*
1114	Li	*	*	*	*	*	*	*

Nº prueba	Método	mu_02	mu_20	mu_11	mu_12	mu_21	mu_22
	Óptimo	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
510	Clásico	1,447	0,2055	2,08	16,2345	372,123	298,8514
515	Balloon	10,9926	4,8843	1,7933	2,5765	78,5698	353,2360
514	Pushing A. C.	1,6203	1,1736	0,7704	9,8079	329,0211	206,7582
509b	Auto-Reconf	2,3304	0,893	0,4561	2,9522	80,072	154,2109
1005	Caselles	77,2879608	85,7289513	103,387608	77,89914287	2,288533543	5901,015827
1044	Lankton	86,9265216	88,7628504	102,090207	85,27675164	57,36584372	4306,542788
1038	Chan	84,3097011	87,5746017	104,137465	79,03191787	62,08780806	5496,472428
1013	Shi	84,5217558	87,994445	103,950092	77,68667642	70,63525102	5689,196616
1012	Bernard	*	*	*	*	*	*
1114	Li	*	*	*	*	*	*

Tabla comparativa de todos los métodos, con medidas estadísticas (media de error y desviación estándar):

Nº prueba	Método	Media Error (1)	Desv. Estand. (1)	Media Error (2)	Desv. Estand. (2)
	Óptimo	0,0000	0,0000	0,0000	0,0000
510	Clásico	55,4428	125,2830	4,2590	5,6094
515	Balloon	40,4881	96,2685	9,2869	12,1514
514	Pushing A. C.	44,5540	102,2893	4,2657	5,6863
509b	Auto-Reconf	19,5049	45,9189	1,7807	2,0922
1005	Caselles	494,5507	1624,8423	12,3772	14,5546
1044	Lankton	376,4800	1181,4079	8,1926	7,5529
1038	Chan	467,2853	1511,5222	7,6505	7,6548
1013	Shi	483,3367	1564,5750	9,7330	8,3302
1012	Bernard	*	*	*	*
1114	Li	*	*	*	*

(1): Con todas las caract.

(2): Con cinco caract.

Resultados para la imagen 2:

Tabla comparativa de todos los métodos. Valores absolutos:

Nº prueba	Método	Area	Centroid_1	Centroid_2	LEMAY	LEMEN	Excent	Perimeter
517	Óptimo	11158	369,74	325,2	343,7	255,82	0,74	556
519c	Clásico	12144	366,62	321,45	331,24	249,11	0,75	413
531	Balloon	8595	414,81	378,5	568,49	392,69	0,69	490
530c	Pushing A.C.	12961	417,4	379,43	475,4	335,24	0,7	511
520 (inic)	Auto-Reconf	11811	370,96	324,9	339,56	252,19	0,74	556
LS-522e	Caselles	12404	355,1	323,156	153,87249	121,8554	0,610619	557,818
1021	Lankton	7742	397,844	378,461	104,79694	99,73643	0,306995	418,617316
LS-521	Chan	10702	357,078	323,008	140,34957	121,6917	0,498201	570,904
1022	Shi	4674	394,5	380	94,685444	65,81793	0,718892	274
1116	Bernard	*	*	*	*	*	*	*
1115	Li	*	*	*	*	*	*	*

Nº prueba	Método	mu_02	mu_20	mu_11	mu_12	mu_21	mu_22
517	Óptimo	86951043	41073922	-18372473	-1571070600	1066542794	405878643
519c	Clásico	92065114	38315060	-18088105	-1653412981	1062614578	832546409
531	Balloon	171638770	84815507	-45386201	-413634540	569292532	293512499
530c	Pushing A.C.	182630156	91214724	-45881271	-274512384	489613949	1262029728
520 (inic)	Auto-Reconf	90236046	41829894	-19083091	-1506778779	1134136961	1318698165
LS-522e	Caselles	11541378,4	18323443,1	458962,1738	- 11972914,41	297384599,6	18220550588
1021	Lankton	5144286,89	4981807,64	236871,644	- 12117950,24	25845180,86	2112534027
LS-521	Chan	9907539,23	13171469,4	101292,5715	- 9625066,229	234139848,8	10940318391
1022	Shi	2618608,5	1265096	0	0	0	708770034
1116	Bernard	*	*	*	*	*	*
1115	Li	*	*	*	*	*	*

Tabla comparativa de todos los métodos. Diferencias Relativas:

Nº prueba	Método	Area	Centroid_1	Centroid_2	LEMAY	LEMEN	Excent	Perimeter
517	Óptimo	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
519c	Clásico	8,8367	0,8438	1,1531	3,6253	2,6229	1,3514	25,7194
531	Balloon	22,9701	12,1896	16,3899	65,4030	53,5025	6,7568	11,8705
530c	Pushing A. C.	16,1588	12,8901	16,6759	38,3183	31,0453	5,4054	8,0935
520 (inic)	Auto-Reconf	5,8523	0,3300	0,0923	1,2045	1,4190	0,0000	0,0000
LS-522e	Caselles	11,1669	3,9596	0,6286	55,2306	52,3667	17,4840	0,3270
1021	Lankton	30,6148	7,6009	16,3779	69,5092	61,0130	58,5143	24,7091
LS-521	Chan	4,0868	3,4245	0,6742	59,1651	52,4307	32,6756	2,6806
1022	Shi	58,1108	6,6966	16,8512	72,4511	74,2718	2,8525	50,7194
1116	Bernard	*	*	*	*	*	*	*
1115	Li	*	*	*	*	*	*	*

Nº prueba	Método	mu_02	mu_20	mu_11	mu_12	mu_21	mu_22
517	Óptimo	0,0000	0,0000	0,0000	0,0000	0,0000	0,0000
519c	Clásico	5,8816	6,7168	1,5478	5,2412	0,3683	105,1220
531	Balloon	97,3970	106,4948	147,0337	73,6718	46,6226	27,6847
530c	Pushing A. C.	110,0379	122,0745	149,7283	82,5270	54,0934	210,9377
520 (inic)	Auto-Reconf	3,7780	1,8405	3,8678	4,0922	6,3377	224,8996
LS-522e	Caselles	86,7266	55,3891	102,4981	99,2379	72,1170	4389,1622
1021	Lankton	94,0837	87,8711	101,2893	99,2287	97,5767	420,4842
LS-521	Chan	88,6056	67,9323	100,5513	99,3874	78,0468	2595,4654
1022	Shi	96,9884	96,9200	100,0000	100,0000	100,0000	74,6261
1116	Bernard	*	*	*	*	*	*
1115	Li	*	*	*	*	*	*

Tabla comparativa de todos los métodos, con medidas estadísticas (media de error y desviación estándar):

Nº prueba	Método	Media Error (1)	Desv. Estand. (1)	Media Error (2)	Desv. Estand. (2)
517	Óptimo	0,0000	0,0000	0,0000	0,0000
519c	Clásico	13,0023	28,4693	7,5809	10,6782
531	Balloon	52,9221	43,4700	14,0354	6,0507
530c	Pushing A. C.	65,9989	64,7216	11,8448	4,9644
520 (inic)	Auto-Reconf	19,5165	61,7492	1,2549	2,5736
LS-522e	Caselles	380,4842	1205,0291	6,7132	7,4370
1021	Lankton	89,9133	104,7862	27,5634	19,3586
LS-521	Chan	245,0097	707,2218	8,7083	13,4591
1022	Shi	65,4221	36,2633	27,0461	25,6360
1116	Bernard	*	*	*	*
1115	Li	*	*	*	*

(1): Con todas las caract.

(2): Con cinco caract.

4.3. Discusión

Una vez terminadas las pruebas y realizado un primer análisis, observamos que el vector de error y la desviación estándar no reflejaban muy bien los resultados visuales obtenidos. Es decir, que en algunas pruebas (por ejemplo la prueba 1038 realizada con el método de Chan sobre la imagen 1) el resultado visual no era malo y, sin embargo, el error medio y la desviación estándar si estaban muy lejos de reflejar el buen resultado que reflejaba el análisis visual. Para solucionar esta situación, se hicieron pruebas con varios subconjuntos de características y se observó que el subconjunto formado por las 5 características área, centroide_X, centroide_Y, excentricidad y perímetro, fue el que mejor reflejaba numéricamente los resultados del análisis visual. Concretamente, en la mencionada prueba 1038, los resultados fueron:

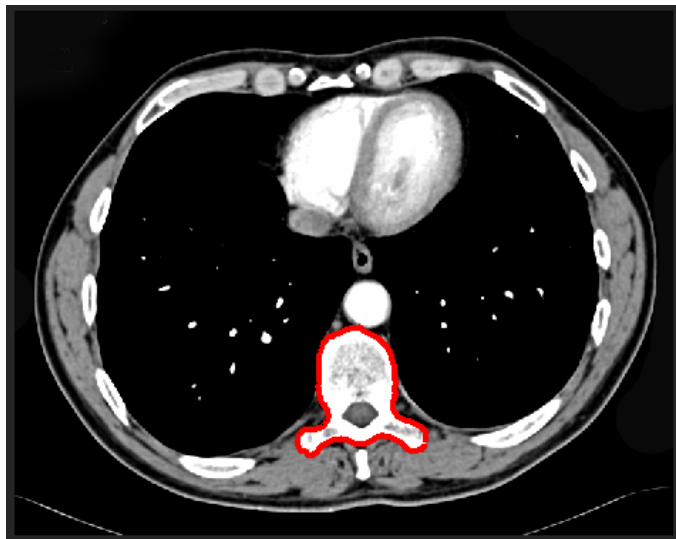
Con todas las características:

Media Error	Desv. Estand.
467,2853	1511,5222

Con las cinco características:

Media Error	Desv. Estand.
7,6505	7,6548

y el resultado visual fue el siguiente:



Con lo que podemos deducir que las medidas estadísticas realizadas sobre esas cinco características reflejan mucho mejor el resultado visual obtenido. Por esto, también podemos deducir que ese subconjunto de cinco características refleja mucho mejor los resultados y, por tanto, es más adecuado observar el subconjunto de esas cinco características mejor que el conjunto completo de las 13 características que analizamos inicialmente en nuestro estudio. Aún así, se han mostrado el error medio y la desviación estándar para los dos grupos de características para poder ver mejor la diferencia entre ambos casos.

Al igual que se han mostrado previamente los resultados de ejecutar los diferentes algoritmos para estas dos imágenes anteriores, también se han realizado pruebas con otras imágenes como se ha podido observar en el capítulo 3, donde se mostraron numerosas pruebas con otras imágenes. En el caso del capítulo 3 se consideró que no era necesario mostrar los vectores de características de cada prueba para no hacer excesivamente tediosa la lectura de ese capítulo, teniendo en cuenta que en el capítulo 3 sólo pretendíamos analizar visualmente las situaciones en los que fallaba cada algoritmo, para enseñar posteriormente como los CAAR podían solventar ese tipo de situaciones.

En cada prueba ejecutada se genera automáticamente un vector de características que se graba en disco como un fichero independiente, por lo que no hubiera sido complicado mostrar todas las características para cada prueba realizada, pero no quisimos exhibir todos los resultados numéricos pues entendimos que podría ser contraproducente.

En este capítulo de resultados hemos mostrado resultados numéricos suficientes para mostrar la forma en la que se han medido cuantitativamente los resultados, eligiendo una

serie de características, integrando dentro de nuestros algoritmos el cálculo automático de las mismas y utilizándolas para analizar los resultados obtenidos. Si bien es verdad que después de un primer análisis se vio la conveniencia de utilizar un subgrupo de esas características, en lugar de todas ellas, pues las estadísticas realizadas con el subgrupo de características reflejaba mejor los resultados visuales estudiados que el conjunto completo de trece características.

Capítulo 5

Conclusiones

A modo de conclusión vamos a enumerar las aportaciones más relevantes de esta tesis doctoral y los resultados obtenidos en la comparación empírica de esta nueva filosofía de contornos activos con algunos de los contornos activos ya existentes:

A) Se ha creado (diseñado, implementado y testado) un nuevo tipo de algoritmo global de segmentación de contornos activos, que hemos nominado como Contornos Activos Auto- Reconfigurables (CAAR), y hemos demostrado de forma empírica que sus resultados mejoran ostensiblemente los conseguidos por los algoritmos de contornos activos existentes en la bibliografía.

La estrategia de uso del CAAR, enfocada a la mejora de los resultados del resto de algoritmos publicados, está basada en el uso de una metodología que incluye un primer paso de ejecución de un algoritmo de segmentación clásico de snakes, seguido de un segundo paso consistente en la detección de forma automática de las

zonas en las cuales el contorno activo resultante de la primera fase no se ha ajustado adecuadamente al objeto a detectar. A continuación se elige, también de forma automática la estrategia que mejore el resultado obtenido en el método inicial. Finalmente se ejecuta dicha estrategia. Empíricamente hemos demostrado que siempre, independientemente del algoritmo utilizado, se mejoran los resultados del snake previo de la primera fase.

En el caso de que el objetivo sea mejorar los resultados de un algoritmo de contornos activos concreto, se podría hacer que este algoritmo fuese el ejecutado durante la primera fase de los CAAR. De esta forma, durante la segunda fase del CAAR, se mejorarían los resultados obtenidos durante la primera fase y, por tanto, los resultados de ese algoritmo del que queríamos mejorar su resultado.

Como es lógico, en el caso expuesto anteriormente, es muy probable que el contorno activo auto-reconfigurable sea capaz de mejorar los resultados del algoritmo utilizado en la primera fase del auto-reconfigurable. Pero los contornos activos auto-reconfigurables no solo obtienen mejores resultados que el algoritmo usado en su primera fase, si no que, los resultados obtenidos con el algoritmo CAAR suele obtener mejores resultados que cualquier otro algoritmo independiente con el que se compare, entendiéndose por independiente un algoritmo que no se use en ninguna fase del CAAR.

B) Se han desarrollado dos métodos novedosos para la selección de la estrategia a utilizar en la segunda fase de los contornos activos auto-reconfigurables. El primer método, llamado *Búsqueda Longitudinal en Abanico (BLA)*, está basado en pequeñas prospecciones progresivas que se van haciendo en la zona comprendida entre el contorno obtenido en la primera fase y el objeto a segmentar, buscándose información de la zona que sirve para elegir el mejor algoritmo a utilizar en la segunda fase. El nombre de *Búsqueda Longitudinal en Abanico* se debe a la distribución espacial en la que se organiza la búsqueda para realizar las prospecciones.

Una de las aportaciones de este método BLA es que el contorno activo no se mueve hasta que no han terminado todas las prospecciones y hasta que no se decida la estrategia a utilizar en la siguiente fase. En este cambio de enfoque reside el principal elemento diferenciador y es el que permite obtener mejores resultados que en el resto de contornos activos. Pues permite utilizar tantas estrategias diferentes como sean necesarias y elegir la

óptima en cada caso, todo esto durante la ejecución del algoritmo de los contornos activos auto-reconfigurables.

El segundo método, el basado en el *Análisis de Cualidades de la Zona (ACZ)*, estudia la zona de interés de la imagen utilizando características parecidas a las utilizadas para medir la similitud entre contornos, pero en este caso se utilizan como elementos de apoyo a la decisión para seleccionar el mejor algoritmo a utilizar en la segunda fase de los contornos activos auto-reconfigurables.

C) El método desarrollado tiene, entre otras innovaciones, un criterio mixto cimentado en el hecho de utilizar varios tipos diferentes de contornos activos dentro de un mismo algoritmo de segmentación, lo cual no ha sido encontrado en ninguna de las búsquedas bibliográficas realizadas.

D) Los CAAR, al utilizar varios tipos de contornos activos, permiten ser utilizados en gran variedad de situaciones diferentes, lo que permite adaptarlos a un ámbito de aplicación o a otro, simplemente cambiando los algoritmos de contornos activos utilizados dentro del CAAR.

E) Se ha comparado de forma empírica esta nueva filosofía de contornos activos con algunos de los contornos activos ya existentes, como por ejemplo, los algoritmos Clásico (de Kass, Witkin y Terzopoulos), Balloon, Bernard, Caselles, Shi, Li, Chan, Lankton, y en todos estos casos se han obtenido mejores resultados con el nuevo tipo de contorno activo propuesto, los CAAR.

F) Se ha realizado un estudio del estado del arte de los contornos activos, mostrando las características principales de una amplia variedad de ellos.

Capítulo 6

Trabajos futuros

En este capítulo planteamos algunas posibles líneas de trabajo para continuar, mejorar o ampliar los resultados obtenidos hasta este momento. En concreto, planteamos los siguientes posibles trabajos futuros:

A) Continuar con las pruebas de las capacidades de otros algoritmos no testeados de segmentación de contornos activos en conjunción con nuestros contornos activos auto-reconfigurables con el fin de intentar mejorar los resultados conseguidos hasta el momento, salvo que el algoritmo con el que se compare obtenga unos resultados tan buenos (cercaos al 100% de exactitud) que no quede ninguna zona del objeto mal segmentada y, por tanto, no sea posible mejorar sus resultados. Si se encuentra(n) algún(os) algoritmo(s) que funcione(n) mejor que los auto-reconfigurables, se podría iniciar otra línea de trabajo, analizando porqué obtiene(n) mejores resultados y estudiando la posibilidad de incorporar a los CAAR los factores

de mejora que tenga(n) esos algoritmo(s) que hubiesen cosechado mejores resultados.

B) Intentar la generalización de nuestra metodología en base al estudio y adaptación de la misma a otros ámbitos distintos al de las imágenes radiológicas.

C) Otra de las posibles líneas de desarrollo futuro de los métodos planteados sería modificar y ampliar nuestra técnica de contornos activos auto-reconfigurables para adaptarla a casos en los que se tengan que segmentar varios objetos al mismo tiempo, ampliando los campos de aplicación de las técnicas desarrolladas. En este trabajo se han realizado siempre pruebas con un único objeto a segmentar, pues es el caso más típico que se realiza en la mayoría de los estudios clínicos consultados, donde es el experto sanitario el que selecciona la estructura anatómica que le interesa estudiar y, por tanto, el objeto a segmentar.

D) Encontrar nuevas situaciones en las que fallen otros contornos activos ya existentes y hallar estrategias para solucionar los problemas encontrados en esas situaciones. Pensamos que en muchos casos bastará con adaptar el CAAR a ese nuevo ámbito de aplicación, por ejemplo, cambiando los algoritmos de contornos activos involucrados dentro del CAAR.

Bibliografía

- [UDUPA00] Udupa, J.K. 2000. 3D Imaging: Principles and Approaches. In 3D Imaging in Medicine. J.K. Udupa & G.T. Herman, Eds.: 1-73. CRC Press. Boca Raton, FL.
- [HARALICK92] Haralick R., Shapiro L. 1992. Computer And Robot Vision. Vols. I, II. Addison-Wesley. New York
- [GONZALEZ92] Gonzalez, R.C. & Woods, R.E. 1992. Digital Image Processing. Addison-Wesley. New York.
- [RUSS99] Russ, J.C. The Image Processing Handbook. CRC Press, IEEE Press, Boca Raton, FL, Third Edition, 1999.

- [AYACHE95] Ayache, N., Cinquin, P., Cohen, I. et al. 1995. Segmentation of complex 3D medical objects: a challenge and a requirement for computer assisted surgery planning and performing. In Computer Integrated Surgery. R. Taylor, S. Lavalée, G. Burdea & R. Moesges, Eds.:59-74. MIT Press.
- [RANGANATH95] Ranganath, S. 1995. Contour extraction from cardiac MRI studies using snakes. IEEE Trans. Medical Imaging. 14 (2): 328-338.
- [UDUPA00b] Udupa, J.K. & Herman, G.T., Eds. 2000. 3D Imaging in Medicine. CRC Press, Boca Raton, FL.
- [DUNCAN00] Duncan, J.S. & Ayache, N. 2000. Medical Image Análisis: Progress over Two Decades and the Challenges Ahead. IEEE Transactions on Pattern Analysis and Machine Intelligence. 22(1): 85-106.
- [AMINI91] Amini, A., Owen, R., Staib, L. et al. 1991. Non-rigid motion models for tracking the left ventricular wall. In Lecture notes in computer science: Information processing in medical images. A. C. F. Colchester & D. Hawkes, Eds: 343-357. Springer-Verlag.
- [AYACHE92] Ayache, N., Cohen, I. & Herlin, I. 1992. Medical image tracking. In Active Vision. Blake, A. & Yuille, A., Eds.: 285-302. MIT.

- [COOTES94] Cootes, T., Hill, A., Taylor, C. et al. 1994. The use of active shape models for locating structures in medical images. *J. Image and Vision computing*. 12, 6, 355-366.

- [MCINERNEY96] McInerney, T. & Terzopoulos. D. 1996. Deformable Models in Medical Image Analysis: A Survey. *Medical Image Analysis*. 1(2): 91-108.

- [BLAKE98] Blake, A. & Isard, M.. 1998. *Active Contours: The application of techniques from graphics, vision, control theory and statistics to visual tracking of shapes in motion*. Springer-Verlag, London.

- [HONEA97] Honea, D. M., Ge, Y., Snyder, W. E. et al. 1997. Lymph-Node Segmentation Using Active Contours. In *Proceedings of SPIE Medical Imaging 1997: Image Processing*. Vol. 3034.

- [HUG99] Hug, J., Brenchbuhler, C. & Szekely, G. 1999. Tamed snake: A particle system for robust semi-automatic segmentation. In *Second International conference on Medical Image computing and Computer-assisted intervention (MICCAI'99)*, number 1679 in LNCS. C. Taylor, & A. Colchester, Eds.:106-115. Springer. Cambridge.

- [DELINGETTE01] Delingette, H. & Montagnat, J. 2001. Shape and Topology Constraints on Parametric Active Contours. *Computer Vision and Image Understanding*. 83: 140-171.

- [KASS88] Kass, M; Witkins, A; Terzopoulos, D. Snakes: Active Contour Models. *Internat. Journal Computer Vision*, vol.1, nº 4, Enero1988, pp. 321-331.

- [COHEN92] Cohen, I., Cohen, L. & Ayache, N. 1992. Using Deformable Surfaces to Segment 3-D Images and Infer Differential Structures. *Computer Vision, Graphics, and Image Processing: Image Understanding*. 56(2): 242-263.

- [METAXAS91] Metaxas, D. & Terzopoulos, D. 1991. Constrained Deformable Superquadrics and nonrigid Motion Tracking. In *International Conference on Computer Vision and Pattern Recognition (CVPR' 91)*: 337-343. Maui, Hawaii.

- [MALLADI95] Malladi, R., Sethian, J., & Vemuri, B. 1995. Shape Modeling with Front Propagation: A Level Set Approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 17 (2): 158-174.

- [SETHIAN96] Sethian, J. 1996. *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision and Materials Science*. Cambridge University Press.

- [CASELLES97] Caselles, V., Kimmel, R. & Sapiro, G. 1997. Geodesic Active Contours. *International Journal of Computer Vision*. 22(1):61-79.

- [UDUPA90] Udupa, J.K. & Herman, G.T., Eds. 1990. 3D Imaging in Medicine. CRC Press, Boca Raton.

- [MARAVALL93] Maravall, D. 1993. Reconocimiento de Formas y Visión Artificial. Ra-ma. Madrid.

- [ROBERTS65] Roberts, L. G., "Machine Perception of Three-Dimensional Solids," in Optical and Electro-Optical Information Processing, pp. 159-197, J. T. Tippett, et al., (Eds), MIT Press, Cambridge, Mass., 1965.

- [SOBEL73] Sobel, I; Feldman, G. *A 3x3 Isotropic Gradient Operator for Image Processing*. Pattern Classification and Scene Analysis, 1973, pp. 271-272.

- [PREWITT70] Prewitt, J.M.S; *Object Enhancement and Extraction*. Picture Processing and Psychopictorics, 1970, pp. 75-149.

- [LIANG93] Liang, Z. Tissue Classification and Segmentation of MR images. IEEE Engineering in Medicine and Biology. March (1993) 81-85.

- [MARR80] Marr D., Hildreth E.: Theory of edge detection. Proc Roy Soc London. 27:187-217, 1980.

- [BOMANS90] Bomans M., Hohne KH., Tiede U., Riemer M.: 3-D segmentation of MR images of the head for 3-D display. IEEE Trans MI 9:177-183, 1990.

- [SAHOO88] Sahoo P.K., Soltani S., Wong A.K., Chen Y.C.: A survey of thresholding techniques. *Comput Niston Graph Image Proc* 41:233-260, 1988.

- [CHOW72] Chow, C.K. y Kaneko, T. Automatic boundary detection of the left ventricle from cineangiograms. *Computers and Biomedical Research*, 5(4):388-410, 1972.

- [PREWITT66] Prewitt, J.M.S., and M.L. Mendelson. The analysis of cell images, *Ann. N.Y. Acad. Sci.* 128:1035-1053 (1966).

- [PICHUMANI97] Pichumani, R., Construction of three-dimensional geometric model for segmentation and visualization of cervical spine images. Tesis Doctoral. 1997.

- [DUDA73] R.O. Duda and P.E. Hart. *Pattern Classification and Scene Analysis*. John Wiley and Sons, New York, 1973.

- [COLEMAN79] Coleman, G. B. and H. C. Andrews. Image segmentation by clustering. *Proc. IEEE*, 67(5): 773-785, 1979.

- [ESCALERA01] De la Escalera, A., “Visión por Computador”. Prentice-Hall. 2001.

•[KASS87] M. Kass, A. Witkin and D. Terzopoulos. “Snakes: Active contour models”. Proc. 1st Int. Conference on Computer Vision, London, 1987, pp. 259-268.

•[MARAIS96] Marais P., Guillemaud R., et al. 1996. Visualising cerebral asymmetry. In Proc Visualisation in Biomedical Computation (VBC'96) in Lecture notes in computer science 1131. K.H. Hohne & R. Kikinis, Eds: 411-416. Springer Verlag, Hamburg.

•[AMINI88] Amini, A.A; Tehrani, S; Weymouth, T.E. *Using Dynamic Programming for Minimizing the Energy of Active Contours in the Presence of Hard Constraint*. Proceedings of the Second International Conference on Computer Vision, IEEE Computer Soc. Press, Tampa, Florida. 1988, pp. 95-99.

•[AMINI90] Amini, A; Weymouth, T; Jain, R. *Using dynamic programming for solving variational problems in vision*. IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 12, nº 9, 1990, pp. 855-867.

•[COUGHLAN98] Coughlan, J; Snow, D; English, C; Yuille, A. *Efficient optimization of a deformable template using dynamic programming*. In Proceedings Computer Vision and Pattern Recognition. CVPR'98. Santa Barbara. California, EE.UU., 1998, pp. 747-752.

- [WILLIAMS92] Williams, Donna J; Shah, Mubarak. *A Fast Algorithm for Active Contours and Curvature Estimation*. Computer Vision Graphics and Image Processing, Vol. 55, N° 1, Enero 1992, pp. 14-26.

- [COHEN91] Cohen, Laurent D. *On Active Contour Models and Balloons*. In Computer Vision, Graphics, and Image Processing: Image Understanding, vol. 53, n° 2, pp. 211-218, Marzo 1991. Aparecido en RR INRIA 1075 en Agosto de 1989.

- [COHEN93] Cohen, Laurent D; Cohen, Isaac. *Finite Element Methods for Active Contour Models and Balloons for 2D and 3D Images*. IEEE Trans. On Pattern Analysis and Machine Intelligence, vol. 15, n° 11, Noviembre 1993, pp. 1131-1147.

- [MARIN99] Marín-Hernández, Antonio; Ríos-Figueroa, Homero V. *EELS: Electric Snakes*. Computación y Sistemas Vol. 2, Nos. 2-3, febrero 1999, pp. 87-94.

- [WEIDNER71] Weidner R.T; Sells R.L. *Elementary Classical Physics*. Editorial Allyn and Bacon, USA, Vol.2, 1971.

- [FEYNMAN64] Feynman, R.P., Leighton, R.B., Sands, M., *The Feynman Lectures on Physics, Mainly Electromagnetism and Matter*, Volume II, Addison-Wesley Massachusetts, 1964.

- [PRICE85] Price, R.H., Crowley, R.J., *The Lighting-rod Fallacy*, American Journal of Physics, september, pp. 843, 1985.

- [HOWING00] Höwing, Frank; Dooley, Laurence S; Wermser, Diederich. *Linguistic Contour Modelling through Fuzzy Snakes*. IEEE Proceedings on Vision, Image and Signal Processing, vol. 147, nº 4, Agosto 2000, pp. 323-330.

- [GUNN94] Gunn, S.R; Nixon, M.S. *A Dual Active Contour*. British Machine Vision Conference, Septiembre 1994, York, U.K, pp. 305-314.

- [LAI94] Lai, K.F; Chin, R.T. *On Regularization, Formulation and Initialisation of the Active Contour Model (Snakes)*. Proc. 1st Asian Conf. on Computer Vision, 1994, pp. 542-545.

- [MCINERNEY95] McInerney, Tim; Terzopoulos, Demetri. *Topologically Adaptable Snakes*. Proceedings of the Fifth International Conferences of Computer Vision (ICCV'95), Cambridge. Junio 1995, pp- 840-845.

- [MCINERNEY00] McInerney, Tim; Terzopoulos, Demetri. *T-Snakes: Topology Adaptive Snakes*. Medical Image Análisis, vol. 4, nº 2, 2000, pp. 73-91.

- [COXETER34] Coxeter, H. S. M. *Discrete groups generated by reflections*, Ann. Of Math., vol. 6, 1934, pp. 13-29.

- [FREUDENTHAL42] Freudenthal, H. *Simplizialzerlegungen von beschränkter Flachheit*, Ann. of Math., vol. 43, 1942, pp. 580-582.

- [MCINERNEY99] McInerney, T.; Terzopoulos, D. *Topology adaptive deformable surfaces for medical image volume segmentation*. IEEE Transactions on Medical Imaging, Vol. 18, nº 10, Octubre 1999, pp. 840-850.

- [IVINS94a] Ivins, Jim; Porrill, John. *Statistical Snakes: Active Region Models*. Fifth British Machine Vision Conference (BMVC'94; York, England): vol. 2, 1994, pp 377-386.

- [LEHMANN01] Lehmann, Thomas; Bredno, Jörg; Spitzer, Klaus. *Texture-Adaptive Active Contour Model*. Proceedings of the Second International Conference on Advances in Pattern Recognition, vol. 2013, 2001, pp. 387-396.

- [HARALICK73] Haralick, R; Shanmugam K; Dinstein, I. *Texture features for image classification*. IEEE Trans. Systems, Man and Cybernetics. vol. 3, 1973, pp. 610-621.

- [HAKEN91] Haken, H. *Synergetic Computers and Cognition*. Editorial: Springer-Verlag New York, 1991, 225 páginas.

- [GALVIN99] Galvin, B; McCane, B; Novins, K. *Virtual Snakes for Occlusion Analysis*. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, 1999, pp. 294-299.

- [BRIGGER98] Brigger, Patrick; Engel, Robert; Unser, Michael. *B-Splines Snakes and a JAVA Interface: An Intuitive Tool for General Contour Outlining*. International Conference on Image Processing, 1998, pp. 277-281.

- [MONTAGNAT98] Montagnat, Johan; Delingette, Hervé. *Globally Constrained Deformable Models for 3D Object Reconstruction*. Signal Processing, volume 71, nº 2, Diciembre 1998, pp. 173-186.

- [DELINGETTE00] Delingette, Hervé; Montagnat, Johan. *Topology and Shape constraints on Parametrics Active Contours*. Technical Report nº 3880, INRIA, Enero 2000.

- [MONTAGNAT00] Montagnat, Johan; Delingette, Hervé; Scapel, Nicolas; Ayache, Nicholas. *Representation Shape, topology and evolution of deformable surfaces. Application to 3D medical image segmentation*. Technical Report nº 3954, INRIA, mayo 2000, 69 páginas.

- [FENSTER01] Fenster, Samuel D; Kender, John R. *Sectored Snakes: Evaluating Learned-Energy Segmentations*. IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 23, nº 9, Sep 2001, pp 1028 – 1034.

•[NGUYEN03] Nguyen, Hieu T; Worring, Marcel; van den Boomgaard, Rein. Watersnakes: energy-driven watershed segmentation. IEEE Transaction on Pattern Analysis and Machine Intelligence, 2003, Volume 25, nº 3, pp. 330-342.

•[VINCENT91] Vincent, L; Soille, P. *Watershed in digital spaces: an efficient algorithm based on immersion simulations*. IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 13, nº. 6, 1991, pp. 583-598.

•[BEUCHER92] Beucher, S; Meyer, F. *The morphological approach of segmentation: the watershed transformation*. In *Mathematical Morphology in Image Processing*, Editorial E. Dougherty, New York, 1992, capítulo 12, pp. 433-481.

•[OSHER88] Osher, S; Sethian, J.A. *Fronts Propagating with Curvature-Dependent Speed: Algorithms Based on Hamilton-Jacobi Formulations*. Journal of Computational Physics, vol. 79, 1988, pp. 12-49.

•[SETHIAN06] Sethian, J.
https://math.berkeley.edu/~sethian/2006/Explanations/level_set_explain.html
on-line, último acceso 27-09-2015.

•[SETHIAN96b] "Level Set Method: An Act of Violence", American Scientist, 1996.

- [DUBROVIN84] Dubrovin, B.A; Fomenko, A.T; Novikov, S.P. *Modern Geometry-Methods and Applications I*. Editorial: Springer-Verlag, 1984.

- [CHAN01] Chan, T.F; Vese, L.A. *Active Contours without edges*. IEEE Transactions on Image Processing, vol. 10, nº 2, 2001, pp. 266-277.

- [MUMFORD89] D. Mumford; Shah J. *Optimal approximation by piecewise smooth functions and associated variational problems*. Comm. Pure Appl. Math., vol. 42, 1989, pp. 577-685.

- [LI08] Li, C., Kao, C.Y., Gore, J.C., Ding, Z. "Minimization of region-scalable fitting energy for image segmentation". En: IEEE Transactions on Image Processing vol 17. no. 10 (oct. de 2008), pags. 1940-1949.

- [DIETENBECK10] Dietenbeck, T., Alessandrini, M., Friboulet, D., Bernard, O. "Creaseg: a free software for the evaluation of image segmentation algorithms based on level-set". Hong Kong, China, 2010.

- [BERNARD09] O. Bernard, D. Friboulet, P. Thevenaz, and M. Unser "Variational B-Spline Level-Set: A linear filtering approach for fast deformable model evolution". En: IEEE Transactions on Image Processing, vol. 18-6 (jun. de 2009), pags. 1179-1191.

- [SHI08] Y. Shi y W. C. Karl. "A real-time algorithm for the approximation of level-set based curve evolution". IEEE Trans. Image Process., volumen 17, no. 5, pp.645-656, 2008.

- [LANKTON08] S. Lankton y A. Tannenbaum. "Localizing region-based active contours". IEEE Trans. Image Process., volume 17, no 11, pp.2029-2039, 2008.

- [YEZZI02] Yezzi JA, Tsai A, Willsky A. "A fully global approach to image segmentation via coupled curve evolution equations". J Vis Comm Image Rep Mar; 2002 13(1):195–216.

- [VILARIÑO00] Vilariño, D.L; Brea, V.M; Cabello, D; Pardo, J.M. *Pixel-Level Snakes*. Proceedings of the 15th International Conference on Pattern Recognition, Volume 1, September 2000, pp. 640-643.

- [VILARIÑO03] Vilariño, D.L; Cabello, D; Pardo, J.M; Brea, V.M. *Cellular Neural Networks and Active Contours: A Tool for Image Segmentation*. Image and Vision Computing, vol. 21, nº 2, 2003, pp. 189-204.

- [RUIZ-MARTIN07] Ruiz-Martín M.M.1, Fernández-Prieto A.2, Viéitez-Vázquez J.1, Mosquera-Lezcano R., "Penetrating orbitocranial injury after trivial eyelid trauma: a case report" [on-line]. 2007. <http://scielo.isciii.es/scielo.php?pid=S0365-66912007001200015&script=sci_arttext> [último acceso: 25 de septiembre de 2015]

