VI. Resultados: Evaluación y comparativa

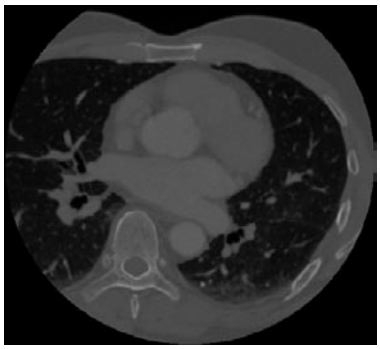
1. Imágenes empleadas para la evaluación

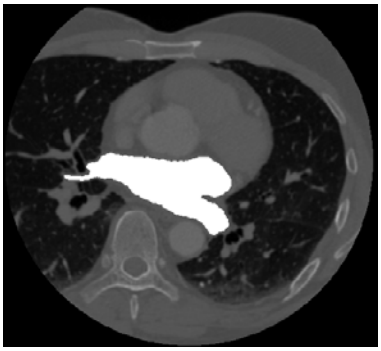
Cada una de las tres imágenes empleadas para la evaluación de los algoritmos forma parte de una serie de imágenes *DICOM* procedente de *TACs* de pacientes distintos. Estas series fueron en primer lugar anonimizadas, y en segundo cargadas en la aplicación *ITK-SNAP*. Esta aplicación, construida sobre ITK, permite visualizar archivos *DICOM*, así como llevar a cabo procesos de segmentación, tanto manuales como semiautomáticos. Con la ayuda de esta herramienta, y por cada una de las 3 series, se generaron los siguientes archivos:

- *ImagenN_TAC.png*. Se trata de una imagen de una de las capas (*slices*) de la serie cargada, exportada en formato *png*, en escala de grises, usando tanto codificación tanto de 16 bits (*ImagenN_TAC_16bits.png*) como de 8 (*ImagenN_TAC_8bits.png*). Esto se realizó con objeto de evaluar las diferencias entre los algoritmos empleando tanto 256 niveles de gris como 65536, para evaluar posibles diferencias. No obstante, salvo el tiempo de ejecución, el resto de variables evaluadas lo fueron solamente con codificaciones de pixel de 16 bits; dada la alta pérdida de calidad de la imagen que supuso el paso a 8 bits, las segmentaciones eran deficientes en general independientemente del algoritmo usado, haciendo fútil el empleo de profundidad de 8 bits para la segmentación de la aurícula izquierda. También es importante añadir que la elección de la capa objeto de la segmentación (de entre todas, en cada una de las tres series de *TACs* empleadas) fue aleatoria, teniendo como única precaución que estuvieran presentes la mayor parte de las estructuras interesantes de la aurícula izquierda, ya que algunas de estas estructuras sólo aparecen en algunos y no en todos los cortes debido a su localización y tamaños, respecto del cuerpo de la aurícula izquierda).
- *ImagenN_GroundTruthSegm.png.* Es una imagen binaria que representa la segmentación manual (o *ground truth*) que se realizó del corte seleccionado, realizada con la herramienta *ITK-SNAP*. Esta segmentación manual fue posteriormente refinada y validada con la ayuda de un experto en Radiología y otro en cardiología.

A continuación se muestran, para cada una de las tres imágenes, tanto la imagen *TAC* original como la segmentación manual que representa la verdadera reconstrucción del objeto (*ground truth*), la aurícula izquierda en este caso. Asimismo, se muestran diversos datos de la serie de imágenes *DICOM* de la cual fueron extraídas.

Imagen 1 – Corte transversal

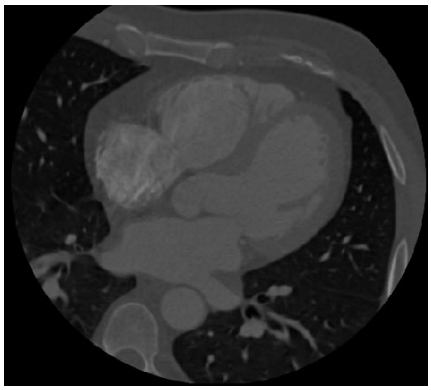




Dimensiones	512 x 512 x 299
Espaciado	0.49 x 0.49 x 0.50
Número de cortes	299
Origen	-103, -32. 1715.150
Byte Order	*Little endian*



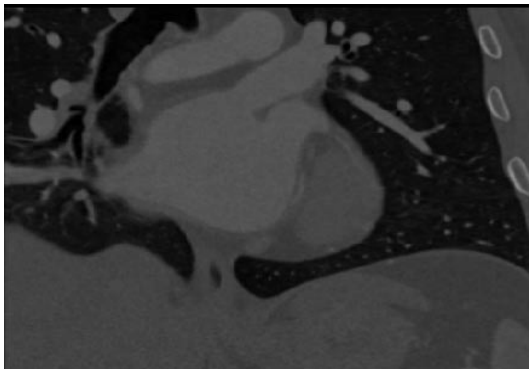
Imagen 2 – Corte transversal





Dimensiones	512 x 512 x 319
Espaciado	0.43 x 0.43 x 0.45
Número de cortes	319
Origen	-116, 2, 1506.4
Byte Order	Little endian

Imagen 3 - Corte coronal





Dimensiones	512 x 512 x 252
Espaciado	0.49 x 0.49 x 0.50
Número de cortes	252
Origen	-114,1.2,1603
Byte Order	Little endian

2. Herramientas empleadas para la evaluación

El conjunto de herramientas empleadas para realizar la evaluación de los algoritmos estudiados en la sección anterior puede resumirse en la Tabla 3:

Herramienta	Función
CMake v2.8.0.	Se trata de la herramienta recomendada por ITK para las tareas de generación, testeo y empaquetado (*build, test & package*) de aplicaciones implementadas sobre *ITK*. *Cmake* es un "*cross-platform, open-source build system*" que permite controlar el proceso de compilación e instalación de software usando archivos de configuración simples e independientes del compilador empleado. Cmake genera *makefiles* y *workspaces* que pueden usarse en el entorno de compilación elegido.
ITK	*Cross-platform Framework* diseñado para el desarrollo de aplicaciones en el campo de la segmentación y el registro de imágenes. La sección Plataforma ITK – Insight ToolKit de este proyecto trata más en profundidad sobre la misma.
ITK- SNAP	Aplicación construida sobre *ITK* y diseñada para visualizar archivos *DICOM*, entre otras, realizar tareas de segmentación 2D y 3D, estructuras en imágenes médicas 3D. En el ámbito particular del proyecto, esta aplicación se usó para llevar a cabo la segmentación manual de las imágenes seleccionadas.
Microsoft Visual C++ Express Edition	*IDE* (*Integrated Development Environment*) escogido para la compilación de *ITK* y de los algoritmos de segmentación, así como de la aplicación CalcularPrecisiónySensibilidad.exe.
Matlab R2009A	Se trata del software usado para la generación de las gráficas mostradas en este proyecto.
Macros MS-DOS	Se emplearon para programar la ejecución en bloque de los bancos de pruebas de los que consta la evaluación de cada algoritmo con cada una de las tres imágenes en todo su espacio paramétrico, proporcionando los parámetros de entrada y salida adecuados en cada caso. También se emplearon para el cálculo de los tiempos de ejecución de los distintos algoritmos evaluados.
Microsoft Office Excel	Empleado para calcular los estadísticos fundamentales (como máximo, mínimo, media, varianza, intervalo de confianza) de las variables evaluadas (precisión, sensibilidad, tiempo de cómputo).
Aplicación ***Calcular Precisión y Sensibilidad.exe***	Aplicación implementada sobre *ITK* específicamente para este proyecto. A partir de dos imágenes segmentadas que se le proporcionen como entrada, lleva a cabo la comparación entre la primera (segmentación *ground truth*) y la segunda (segmentación a evaluar) y genera las variables necesarias con las estadísticas usadas para la evaluación de los distintos algoritmos (falsos positivos, falsos negativos, verdaderos positivos, y por último precisión y sensibilidad). También escribe los resultados en el archivo que recibe como tercer argumento. Adicionalmente se escriben en la salida especificada otras variables usadas, como el número de píxeles blancos y negros de cada imagen.

Tabla 3: Herramientas empleadas

3. Metodología empleada para la evaluación

A la hora de diseñar la metodología de evaluación a emplear, se contaba con los siguientes requisitos de partida:

- Elección de algoritmos a evaluar. Esta elección se basó en diversos criterios expuestos en la sección Algoritmos de segmentación evaluados.
- Elección del banco de imágenes sobre el que realizar la evaluación. Estas han sido expuestas en la sección Imágenes empleadas para la evaluación.
- Segmentaciones *ground truth* de las imágenes escogidas. Supervisión por un experto en radiología.
- Elección de una métrica para la evaluación y comparación de los algoritmos. La métrica ha sido expuesta en la sección

- Especificación de la métrica empleada en la evaluación, y como se ha visto está esencialmente basada en las variables precisión y sensibilidad.
- Implementación de una aplicación sobre *ITK* para calcular las variables necesarias para evaluar las métricas sobre los distintos algoritmos. Esta aplicación, a partir de las dos imágenes que recibe como parámetros de entrada, calcula, esencialmente, la precisión y la sensibilidad de la segunda (la segmentación a evaluar) con respecto a la primera (*ground truth*). Para no distorsionar las medidas de tiempo, el cálculo del tiempo de ejecución de los algoritmos no se ha incluido el tiempo empleado por el sistema en calcular la precisión y la sensibilidad. No obstante, el tiempo de esta aplicación en ejecutarse se midió también durante los distintos *tests*, obteniendo tiempo relativamente estables y pequeños, siempre dentro del margen [0.1-0.2] segundos, tiempo despreciable en comparación con los tiempos de ejecución de la mayoría de los algoritmos evaluados. Esta aplicación tenía como requisito escribir los datos de salida de una manera conveniente para su procesado posterior (en Matlab y Excel).
- Definición del proceso de evaluación para cada algoritmo. Para evitar tiempos de computación demasiado elevados, para cada algoritmo se llevó a cabo una evaluación de tanteo previa a las evaluaciones que se exponen en las siguientes sub-secciones. Estas pruebas sirvieron para delimitar un rango de valores del/ de los parámetro/s del algoritmo en los cuales la salida es distinta de la trivial. La salida trivial de un algoritmo, en el contexto de la evaluación realizada de este proyecto, sería una imagen "todo a cero" (es decir, sub-segmentación total), o "todo a uno", (sobre-segmentación total). En estos casos, los valores de la precisión y sensibilidad son triviales ([p=0, s=1] en el primer caso -imagen sub-segmentada, y [p=1. S=0] en el segundo caso, imagen sobre-segmentada). De esta manera, sólo se evaluaron combinaciones de parámetros que no producían esta salida trivial. Este proceso inicial de tanteo, gracias al cual se diseñó la estrategia de evaluación de parámetros, disminuyó el tiempo de cálculo sensiblemente con respecto a los que se hubieran manejado con el método "por la fuerza bruta". En otras palabras, las gráficas de precisión y sensibilidad que se presentan sólo contienen información sobre los parámetros que no producen salidas triviales.

La metodología para la evaluación de los distintos algoritmos queda resumida en la Ilustración 18.

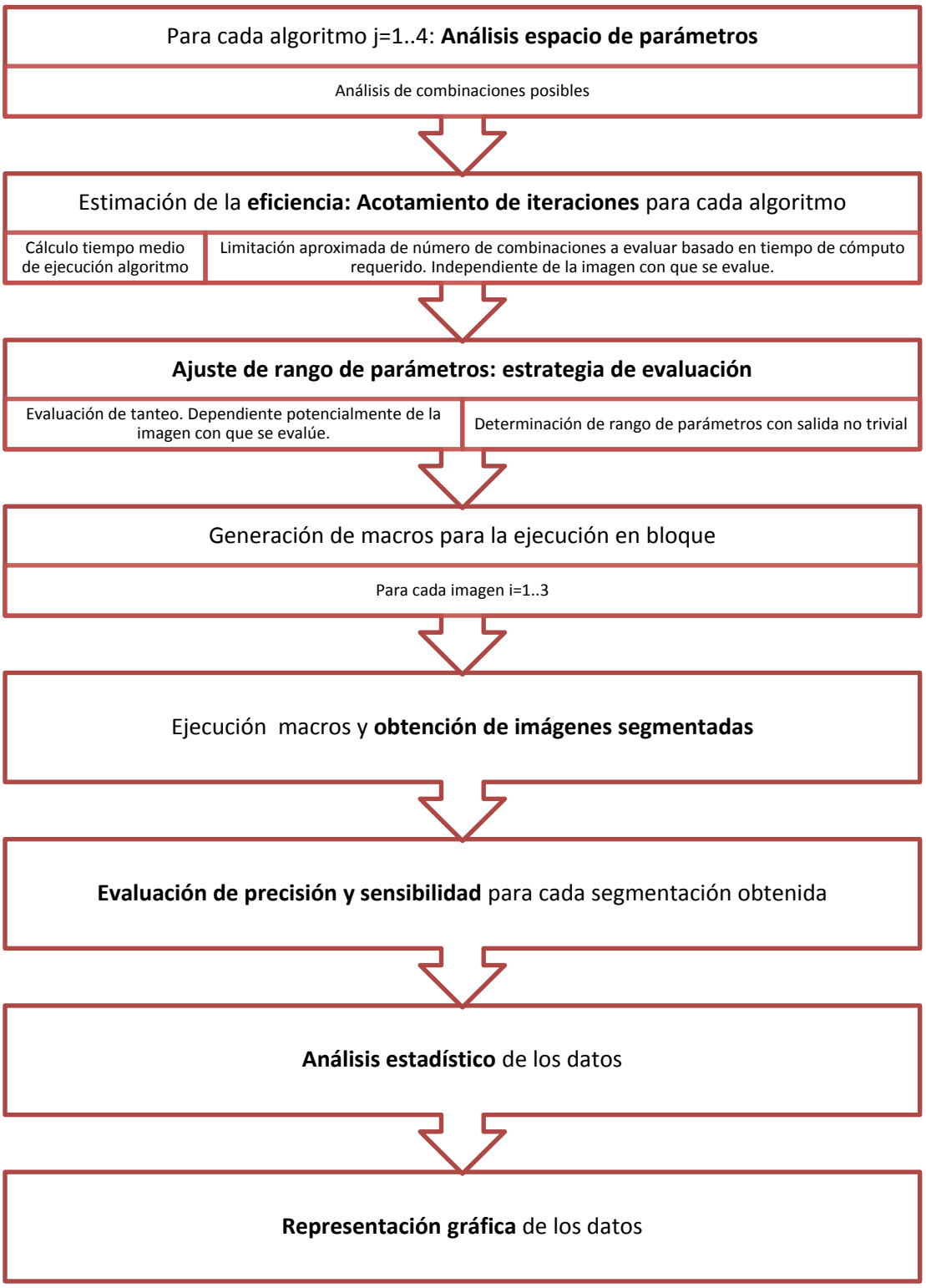


Ilustración 18: Diagrama de flujo empleado para la evaluación de los distintos algoritmos

A continuación se detallan los resultados obtenidos en la evaluación de cada algoritmo. La presentación de los mismos se hará siguiendo los pasos descritos en la Ilustración 18.

4. Algoritmo ConfidenceConnected. Evaluación

1. Análisis de espacio paramétrico

Conviene remarcar que durante la evaluación de los diversos algoritmos las referencias al término parámetros excluirán dos parámetros de entrada comunes a todos los algoritmos de segmentación evaluados: nombre del archivo con la imagen a segmentar y nombre de la imagen segmentada de salida.

El algoritmo ConfidenceConnected recibe dos parámetros de entrada: *SeedX SeedY*. Se trata de la semilla, es decir, dos números enteros que representan las coordenadas de un pixel en la imagen a segmentar que está dentro del área que ocupa el objeto que se pretende segmentar.

El número de semillas posibles que el operador podría potencialmente utilizar sería igual al número de píxeles que posee el objeto a segmentar, la aurícula izquierda, en la imagen. O, lo que es lo mismo, el número de píxeles en blanco de la imagen segmentada manualmente o *ground truth*. La aplicación CalcularPyS.exe proporciona estos datos, que se resumen en siguiente tabla (Tabla 4):

Imagen	Píxeles pertenecientes a la aurícula izquierda
1	9766
2	9746
3	15763

Tabla 4: área de las imágenes a segmentar

Así, el número de posibles semillas para la imagen A sería el número correspondiente indicado en la tabla, 9766. Semejantemente podemos establecer el número potencial de semillas para B y C.

2. Estrategia de evaluación

Durante la fase de tanteo se llevaron a cabo pruebas con el algoritmo que determinaron que el tiempo de ejecución era independiente de la semilla proporcionada, y siempre se encontraba dentro de los márgenes [0.98, 1.03] segundos. Se observó asimismo que el resultado en cuanto a precisión y sensibilidad era en todos los casos altamente satisfactorio, salvo, en ocasiones, al proporcionar una semilla en la frontera de la aurícula izquierda con otras estructuras.

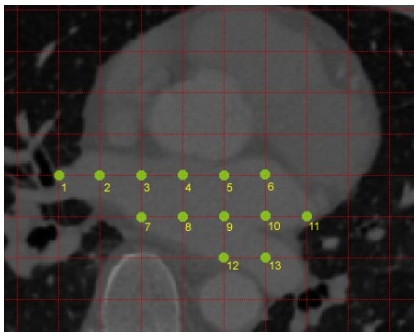
Dada la estabilidad que se observó en los resultados en cuanto a precisión y sensibilidad, y que la única diferencia en resultado se producía al proporcionar una semilla localizada justo en la frontera con otras estructuras, se decidió llevar a cabo un proceso de reticulado de la imagen (tamaño de retícula: 30 x 30 píxeles) para disminuir el espacio de pruebas.

Podría argumentarse que esta modificación del espacio de pruebas alterará la media de resultados, lo cual es sin duda cierto. Sin embargo, no es menos cierto que la probabilidad de que el operador posicione la semilla en la frontera del objeto es sensiblemente menor que la de que la semilla la posicione en el interior, tendiendo probablemente al centro, del cuerpo de la aurícula izquierda. Las semillas puestas en los bordes se incluyen en la estrategia de evaluación fundamentalmente para cuantificar cuán deficientes llegan a ser las segmentaciones producidas por este algoritmo en dicho caso, aunque sería poco probable un mal ajuste de este parámetro por parte del operador.

3. Resultados de la evaluación de ConfidenceConnected

Imagen 1 – ConfidenceConnected

En la imagen de la derecha se representa la imagen 1 junto con las semillas empleadas, calculadas con ayuda de una retícula equiespaciada, para las cuales se evaluó el algoritmo.



A continuación se representan los resultados obtenidos para las semillas posicionadas en las fronteras con otras estructuras cardíacas.

Puede observarse en las dos ilustraciones de la derecha, de izquierda a derecha, las salidas del algoritmo para la semilla 11 (localizada en la frontera de la aurícula izquierda) y para la semilla 1 (localizada en la frontera opuesta a la semilla anterior), respectivamente. Como puede verse, los resultados obtenidos son pobres cuando la semilla no se encuentra dentro de la estructura, sino en los bordes con otras estructuras.





En cambio, con cualquier otra semilla se obtienen resultados bastante satisfactorios. Para ilustrar esto se muestra a la derecha la salida al especificar la semilla 8 (interna a la aurícula izquierda) como entrada al algoritmo (imagen segmentada en blanco), junto con la segmentación *ground truth* (imagen segmentada en rojo).





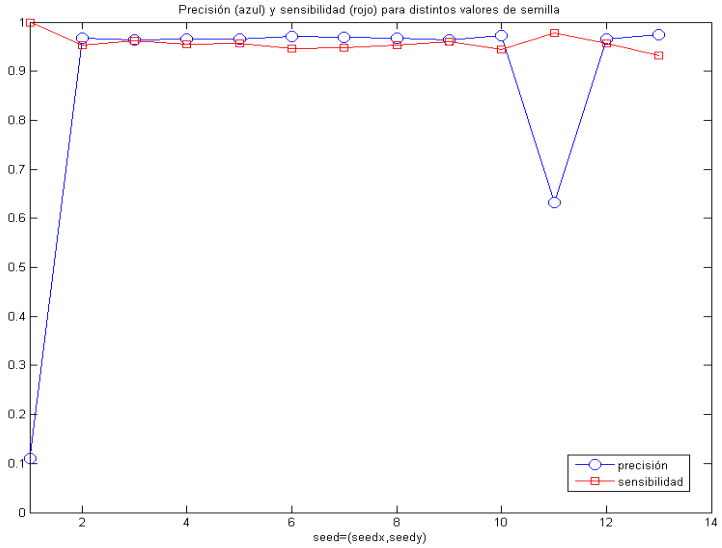
La tabla de la derecha representa los cálculos de precisión y sensibilidad obtenidos para cada semilla. Se representan sombreadas en gris las semillas para las que se obtuvieron valores de precisión y sensibilidad dentro de la zona estable (p>90% & s>90%).

Asimismo, la tabla de abajo representa los valores estadísticos calculados medios calculados a partir de la tabla de la derecha.

seed	precisión	sensibilidad
1	0,110203	1
2	0,967159	0,952898
3	0,963096	0,962011
4	0,966321	0,954843
5	0,96587	0,956277
6	0,970088	0,946447
7	0,96894	0,9487
8	0,9672	0,954127
9	0,963849	0,960987
10	0,972272	0,944297
11	0,63229	0,978087
12	0,965617	0,957608
13	0,973814	0,932931

Precisión media	0.967
Sensibilidad media	0.952
Rango de variación de la precisión	[0.963, 0.974]
Rango de variación de la sensibilidad	[0.933, 0.962]

La imagen de la derecha representa la precisión (en azul) y la sensibilidad (rojo) obtenidas en función de la semilla especificada como entrada.



En la gráfica de la derecha se ha resaltado con trazo verde la ***zona estable*** del algoritmo en su evaluación con la primera de las imágenes. Escogiendo el parámetro de entrada dentro de esta zona (semilla dentro de la aurícula izquierda), la precisión y la sensibilidad obtenidas están dentro de un rango aceptablemente bueno.

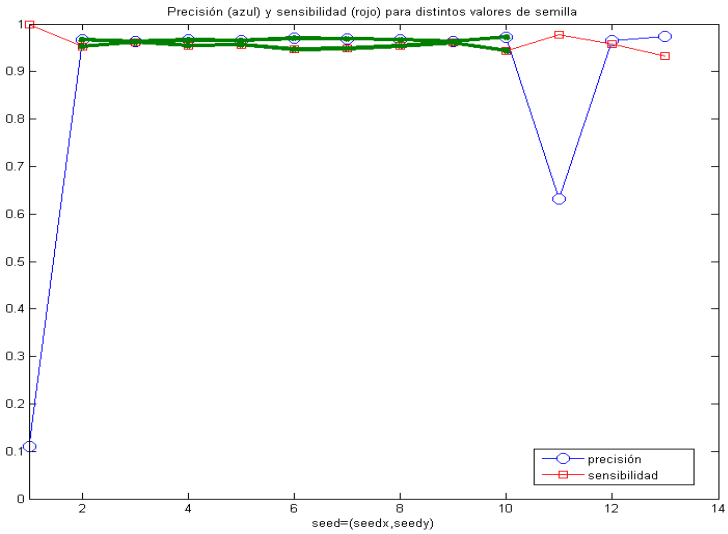
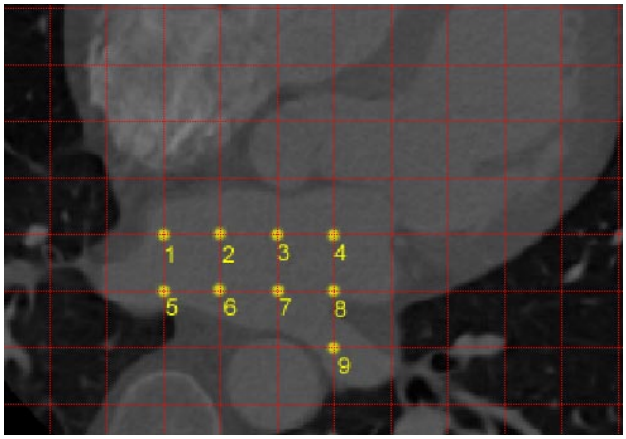


Imagen 2 – ConfidenceConnected

Análogamente a como se procede con la imagen 1, pasamos a definir la estrategia de evaluación del algoritmo *Confidence Connected* con la segunda imagen. En la imagen de la derecha se representa la retícula construida para este fin.



En la tabla de la derecha se muestran los valores de precisión y sensibilidad obtenidos para las distintas semillas ensayadas, y en la tabla de abajo se representan los estadísticos medios de los valores de la tabla de la derecha.

seed	precisión	sensibilidad
1	0,43283	0,97127
2	0,439538	0,952904
3	0,432357	0,972604
4	0,431366	0,975067
5	0,432424	0,972399
6	0,431563	0,974759
7	0,432304	0,972707
8	0,43734	0,962138
9	0,430929	0,975272

Precisión media	0.433
Sensibilidad media	0.9699
Rango de variación de la precisión	[0.43, 0.439]
Rango de variación de la sensibilidad	[0.9529, 0.975]





La imagen de la izquierda (la segmentada en blanco) muestra la mejor de las salidas obtenidas con este algoritmo para la imagen 2. Puede verse claramente que el resultado dista en mucho de ser satisfactorio, si se compara con la segmentación ground truth (segmentada en rojo). El algoritmo ha considerado aurícula y ventrículo izquierdos como parte de una sola estructura.

El resultado de ConfidenceConnected con la imagen 2 resulta bastante más deficiente que el obtenido con la imagen 1. Estos valores de precisión baja y sensibilidad muy alta obtenidos son característicos de un escenario de sobre-segmentación. En este caso esta sobresegmentación ocurre para cualquier valor del parámetro semilla, como puede deducirse claramente de la baja precisión (puede consultarse tanto en la tabla numérica de arriba como en la representación gráfica de estos valores de la página siguiente, trazado azul – precisión).

Analizando la imagen 2 en comparación con la 1 o la 3, puede observarse que el nivel de intensidad en la cámara adyacente a la aurícula izquierda (ventrículo izquierdo) es mucho más semejante al existente en ésta en la imagen 2 que en el resto. Este hecho, unido a la continuidad existente entre ambas cámaras (a través de la válvula mitral), claramente patente en el corte transversal escogido, explica lo poco satisfactorio del resultado obtenido con este algoritmo. El algoritmo ConfidenceConnected funciona con analizando los niveles de intensidad de los áreas vecinas a la estructura considerada. En los casos en los que una cámara adyacente a la aurícula izquierda, como en este caso el ventrículo izquierdo, no está bien separada por la cámara de interés bien una frontera bien delimitada, bien con unos niveles de intensidad distinguibles de la anterior (en el caso de imagen 2, estas diferencias están prácticamente ausentes), el algoritmo ConfidenceConnected produce unos resultados bastante mediocres.

La gráfica de la derecha representa los valores de precisión y sensibilidad obtenidos con ConfidenceConnected para la imagen 2. Puede verse que no existe zona estable en la que la precisión y sensibilidad se mantengan simultáneamente por encima del 90%.

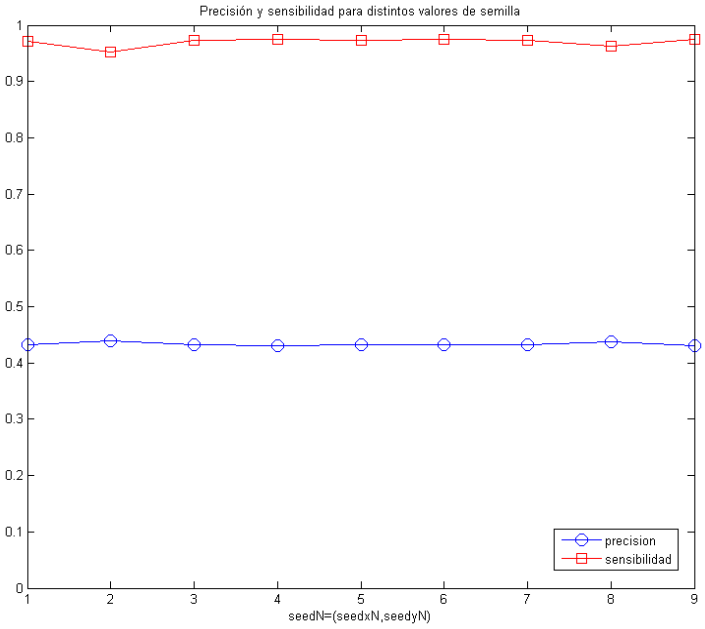
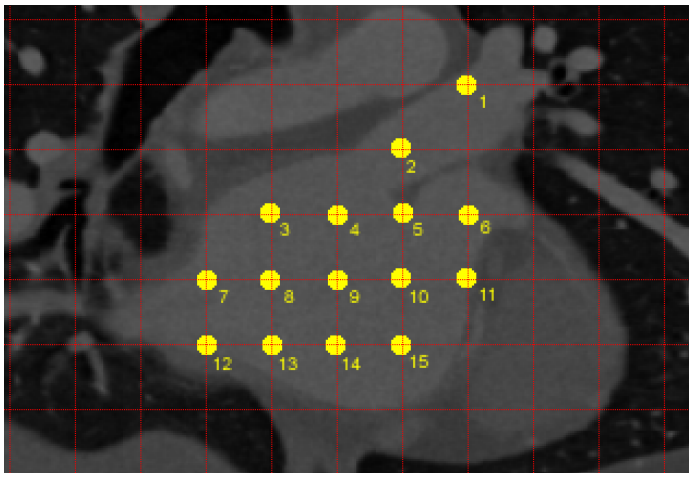


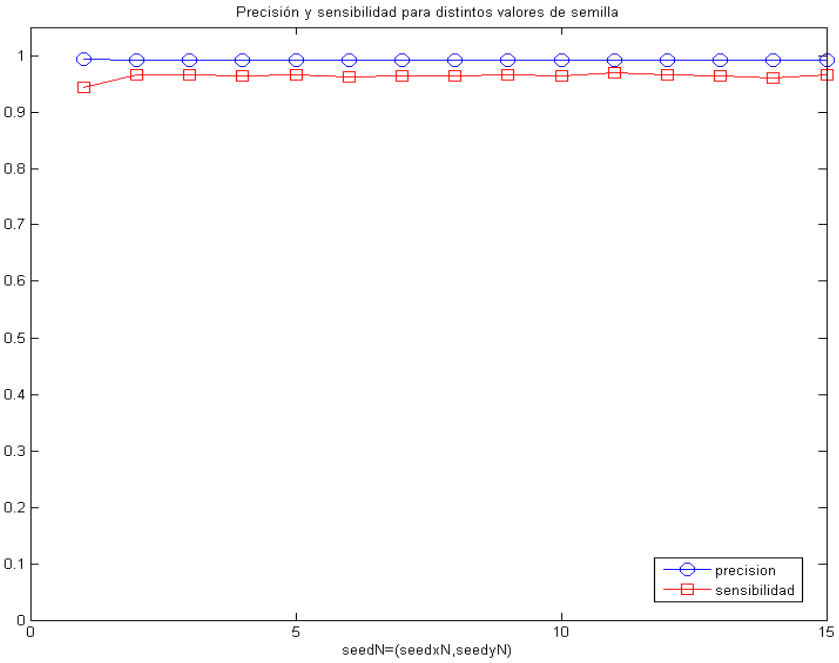
Imagen 3 – ConfidenceConnected

En la imagen de la derecha pueden verse las semillas escogidas para evaluar el algoritmo ConfidenceConnected con la imagen 3.



En la tabla de abajo se detallan los valores de precisión y sensibilidad obtenidos con las segmentaciones producidas con ConfidenceConnected para cada una de las semillas representadas, y a su derecha se representa la misma información gráficamente.

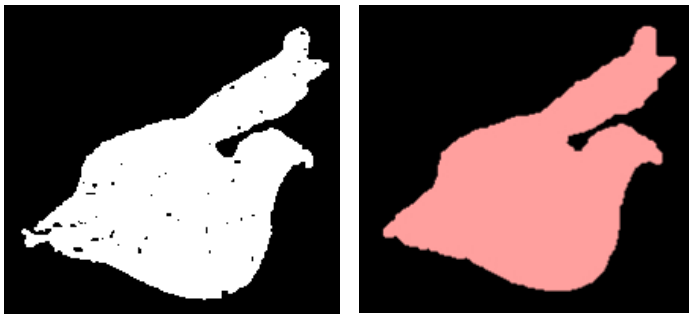
Semilla	Precisión	Sensibilidad
1	0,993392	0,94411
2	0,991337	0,965552
3	0,991331	0,964854
4	0,991513	0,963522
5	0,991336	0,965425
6	0,991758	0,961809
7	0,991391	0,964283
8	0,991513	0,963522
9	0,991336	0,965362
10	0,991513	0,963522
11	0,990787	0,968788
12	0,991337	0,965552
13	0,991388	0,96403
14	0,991804	0,959589
15	0,991336	0,965362



En la gráfica de arriba puede apreciarse una zona estable tan ancha como todo el rango paramétrico del algoritmo, así como unos valores sensiblemente altos de precisión y sensibilidad obtenidos. Y la tabla de la izquierda refleja las medias, máximos y mínimos de los valores de la zona estable.

Precisión media	0.9915
Sensibilidad media	0.9630
Rango de variación de la precisión	[0.9908, 0.9934]
Rango de variación de la sensibilidad	[0.9441, 0.9679]

Las dos imágenes de la derecha ilustran el buen resultado obtenido con prácticamente cualquier semilla empleada. La segmentación en blanco corresponde a la realizada con *ConfidenceConnected* usando la semilla 11 como entrada, y la segmentación en rojo se corresponde con la *groundtruth*.



5. Algoritmo IsolatedConnected. Evaluación

1. Análisis de espacio paramétrico

En el caso del algoritmo *IsolatedConnected*, los parámetros requeridos son 5 en número, aunque conceptualmente son 3 ítems los que se requieren, a saber:

- *SeedIn*: Semilla dentro del objeto a segmentar
- *Threshold*: Umbral inferior de intensidad, bajo el cual no se encuentran los píxeles que ocupan la zona de interés.
- *SeedOut*: Semilla fuera del objeto a segmentar, en una zona que no se desea que forme parte de la imagen segmentada de salida

Al igual que en el análisis que se hizo para el anterior algoritmo, hay tantas posibles semillas como píxeles en la imagen. Y tantas semillas apropiadas *seedIn* como píxeles blancos en la imagen *groundtruth*. Sin embargo, la probabilidad de que el operador indique un pixel de la estructura a segmentar correctamente es por necesidad alta.

No obstante, hay una variabilidad mayor en cuanto a los valores la semilla *SeedOut*. El operador tiene la opción de marcar como *SeedOut* cualquier estructura que tenga frontera con la aurícula izquierda, y probablemente estas distintas estructuras tengan distinta intensidad de gris, con lo que es posible que el algoritmo modifique su resultado a causa de este parámetro.

Y en cuanto al umbral, tenemos un parámetro que puede tomar un valor entre 0 y 65535, si se trabaja con imágenes con 16 bits por píxel, o entre 0 y 255 si se trabaja con imágenes con niveles codificados con 8 bits. El caso que se analiza es el más desfavorable (16 bits), que además es el de utilidad en la práctica clínica para la segmentación de la aurícula izquierda, tarea para la cual se enfoca este proyecto.

2. Estrategia de evaluación

Durante la fase de tanteo, se calculó el tiempo de ejecución, con media 1.17 ± 0.2segundos (variabilidad despreciable en el contexto del análisis realizado), independientemente de los parámetros o del nivel de grises con que se ejecutara el algoritmo.

A continuación se llevaron a cabo las evaluaciones de tanteo con cada una de las tres imágenes. Esta evaluación de tanteo sirvió para determinar, para cada imagen, el umbral de parámetros donde la salida es no trivial. La Tabla 5 recoge los hallazgos:

Imagen	Rango de umbrales con salida no trivial
1	[15000 – 20698]
2	[17750 – 22343]
3	[15800 – 21350]

Tabla 5: Rango de parámetros con salida no trivial. Isolated Connected.

Para evaluar el algoritmo con cada una de las tres imágenes se escogieron distintas semillas. En este párrafo nos referimos a la elección de semilla para el caso de la imagen 1, dado que para el resto extiende fácilmente la metodología. Para la evaluación con la imagen 1 se trabajó con 4 distintas semillas *SeedIn*, todas dentro pero en distintas zonas de la aurícula izquierda (una más central en la aurícula izquierda, y otra más periférica, en la orejuela) y con 4 semillas *SeedOut*, cada una en una estructura anatómica limítrofe con la aurícula izquierda. En particular, se posicionaron *SeedOuts* en localizaciones ventricular izquierda, aórtica descendente, septales y pulmonares. El resultado varió de una manera muy poco significativa para un umbral constante, en cuanto a precisión y sensibilidad, al realizar pruebas con las distintas combinaciones de semillas. Sin embargo, la dependencia con el parámetro umbral se mostró muy elevada.

A la luz de estos resultados, se planeó la estrategia de pruebas específicamente para resaltar y cuantificar esta dependencia.

Se evaluó el algoritmo IsolatedConnected siguiendo los valores de umbral recogidos en la Tabla 6 :

Img	Umbral inicial	Umbral final	Salto (en píxeles)	Número de iteraciones
1	17800	23343	Retícula variable: entre 46 en la zona menos densa, hasta 30 en la zona de más interés. Fuera de este rango: salida trivial.	100
2	15800	21350	Retícula variable: entre 60 en la zona menos densa, hasta 5 en la zona de más interés. Fuera de este rango: salida trivial.	100
3	15000	20860	Retícula variable: entre 500 en la zona menos densa, hasta 40 en la zona de más interés. Fuera de este rango: salida trivial.	100

Tabla 6: Estrategia para la evaluación de IsolatedConnected. Parámetro LowerThreshold.

En todos los casos, *SeedIn* se escoge en el centro de la aurícula izquierda, y *SeedOut* se sitúa en una estructura adyacente cuyo nivel de intensidad se contrasta fácil visualmente del de la aurícula izquierda, puesto que se comprobó previamente la débil dependencia con la semilla especificada, siempre que estas se proporcionaran dentro y fuera de la aurícula izquierda, respectivamente.

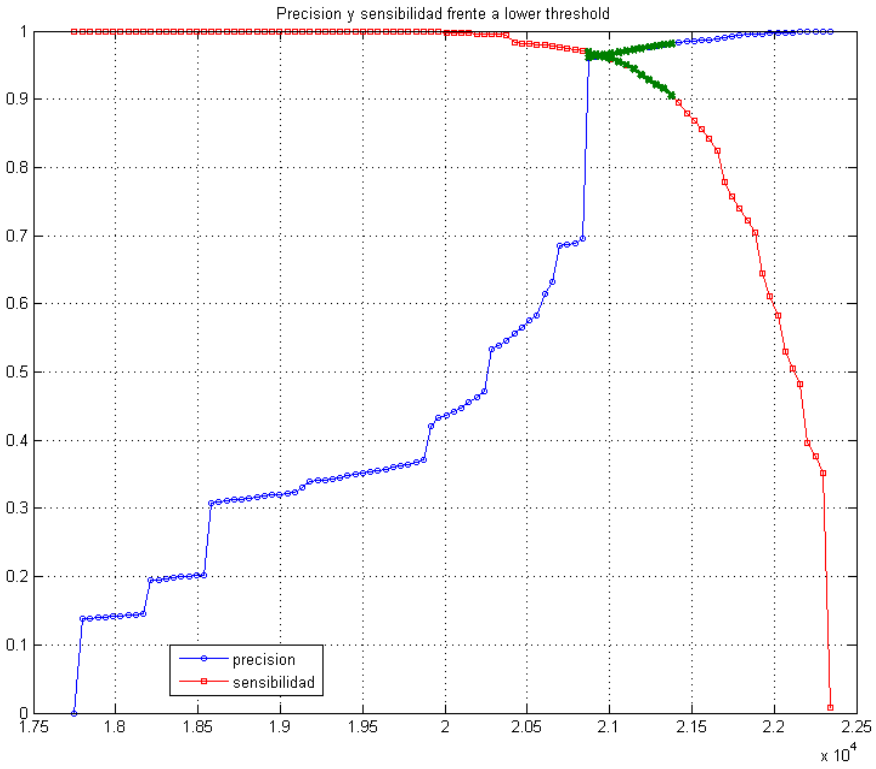
3. Resultados de la evaluación de IsolatedConnected

Imagen 1 – IsolatedConnected

La consideración de la zona estable en este caso, dadas las rápidas fluctuaciones de la precisión y la sensibilidad al cambiar los parámetros, comprende el espacio de parámetros que mantenían tanto la precisión como la sensibilidad por encima del 90%. Estos hallazgos quedan resumidos en la tabla de la derecha.

Zona estable: p>0.9 y s>0.9	Umbral Є [20870, 21380]
Precisión	P Є [0.9609, 0.9807]
Valor medio precisión en zona estable	0.97144
Sensibilidad	S Є [0.9694, 0.9061]
Valor medio sensibilidad en zona estable	0.9427

En la gráfica de la derecha se representan precisión y sensibilidad frente al parámeto lower threshold para las semillas in y out fijadas en la fase de tanteo. Pueden observarse más fácilmente los resultados de la tabla de arriba, al representarse en verde la zona estable del algoritmo. También puede observarse la rapidez de la decadencia de la respuesta del algoritmo conforme el parámetro *lowerthreshold* se aleja de dicha zona estable.



La mejor de las segmentaciones obtenidas se produce para los valores de la tabla de la derecha (en la gráfica de arriba se correspondería con el punto donde se encuentran aproximadamente precisión y sensibilidad). Dicha segmentación óptima tiene el aspecto que se muestra en la imagen de la derecha (segmentación en blanco). En rojo se representa la segmentación verdadera o *groundtruth*.

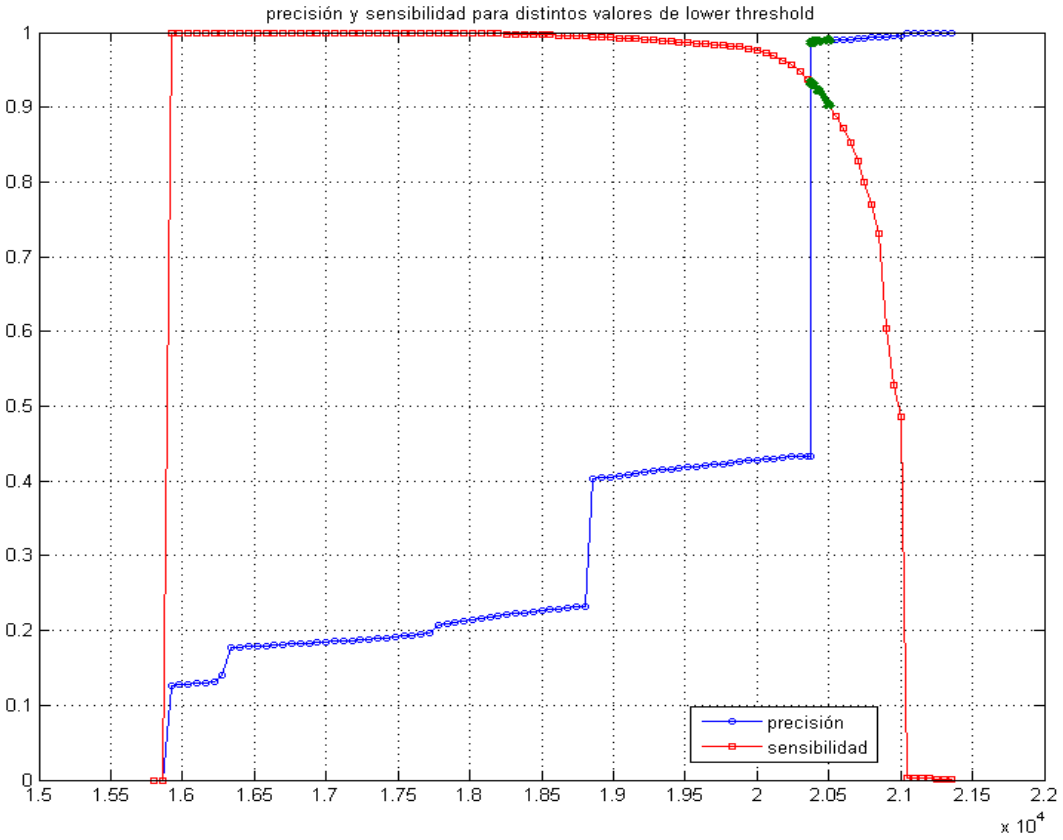
Segment. óptima	
Umbral	20871
Precisión	0.9609
Sensibilidad	0.9693



Imagen 2 – IsolatedConnected

La zona estable en este caso es muy estrecha, dadas las rápidas fluctuaciones de la precisión y la sensibilidad al cambiar los parámetros. Los hallazgos en cuanto a precisión y sensibilidad quedan resumidos en la tabla de la derecha, y representados en la gráfica de abajo. En dicha gráfica puede comprobarse la zona estable del algoritmo, y la extrema rapidez en la decadencia de su respuesta al alejarnos de dicha zona.

Zona estable: p>0.9 y s>0.9	Umbral Є [20375, 20500]
Precisión	P Є [0.9876, 0.9895]
Valor medio precisión en zona estable	0.9881
Sensibilidad	S Є [0.903, 0.9333]
Valor medio sensibilidad en zona estable	0.9242



La mejor de las segmentaciones obtenidas se produce aproximadamente cuando se encuentran precisión y sensibilidad, tomando los valores recogidos en la tabla de la derecha. La imagen segmentada correspondiente a estos valores se representa en la imagen de la derecha (segmentación en blanco). Para su comparación, se muestra a su lado la segmentación verdadera o *groundtruth* (en rojo).

Segment. óptima	
Umbral	20375
Precisión	0.9876
Sensibilidad	0.9333

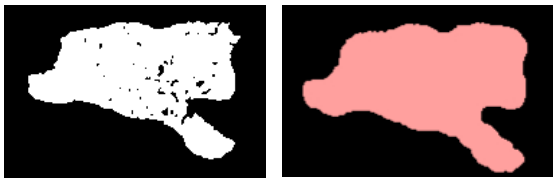
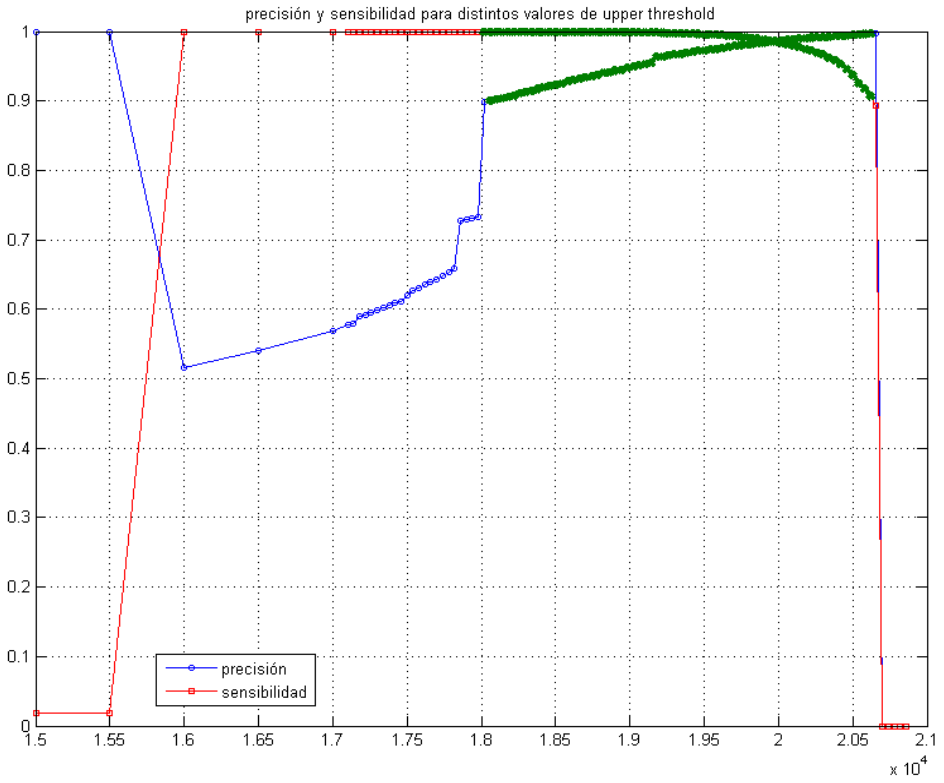


Imagen 3 - IsolatedConnected

Zona estable: p>0.9 y s>0.9	Umbral Є [18060, 20620]
Precisión	P Є [0.9000, 0.9967]
Valor medio precisión en zona estable	0.9591838
Sensibilidad	S Є [0.9070, 1]
Valor medio sensibilidad en zona estable	0.98748

La zona estable en este caso es significativamente mayor que en el caso de la imagen anterior (imagen 2). La gráfica de abajo muestra claramente el amplio rango, en contraste con las anteriores dos imágenes, con el que obtenemos resultados buenos de manera estable. La de arriba recoge los estadísticos correspondientes al valor medio, máximo y mínimo de la precisión y la sensibilidad en esta zona estable.

En la gráfica de la derecha se representan precisión y sensibilidad frente al parámetro de entrada lower threshold. Puede comprobarse la anchura de la zona estable del algoritmo, y la rapidez de la decadencia de su respuesta al alejarnos de dicha zona.



Umbral	19980
Precisión	0.9861
Sensibilidad	0.9861

La mejor de las segmentaciones obtenidas se produce aproximadamente cuando se encuentran precisión y sensibilidad, tomando los valores recogidos en la tabla de la derecha. La segmentación correspondiente a dicho valor óptima se representa a la derecha (estructura en blanco), y a su lado se muestra la segmentación groundtruth (en rojo).





6. Algoritmo OtsuMultipleThreshold. Evaluación.

1. Análisis del espacio paramétrico

El algoritmo *OtsuMultipleThreshold* posee el espacio paramétrico más simplificado de todos los estudiados. Se trata de un número entero, que representa el número de umbrales en los que basa la segmentación que realiza. El algoritmo calcula $n + 1$ segmentación, es decir, una imagen segmentada por cada uno de los rangos en que el histograma queda dividido, siendo n el número de umbrales que se calcularán en el histograma (parámetro de entrada).

Su principal desventaja, tal como ha quedado probado en la evaluación que a continuación se presenta, es la baja calidad de las segmentaciones que produce, así como el tiempo de cálculo, que crece exponencialmente con el valor de umbral especificado.

2. Estrategia de evaluación

Durante la evaluación de tanteo, se obtuvieron los tiempos medios de ejecución. Estos tiempos fueron independientes de la imagen con que se empleara el algoritmo. Sorprendentemente, los tiempos de ejecución apenas variaron usando las mismas imágenes con 256 niveles. Dichos tiempos se recogen en la tabla de la derecha.

Number Of Thresholds	Tiempo medio de ejecución con 65536 niveles	Tiempo medio de ejecución con 216 niveles
1	0.65	0.62
2	0.67	0.61
3	1.07	1.02
4	13.49	13.55
5	344.7	344.1
6	7734.69 (2h 21min)	7554.3 (2h 6min)

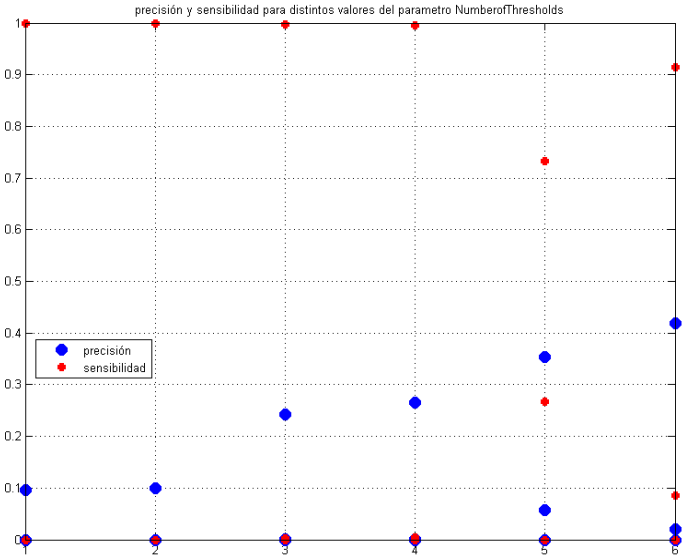
La estrategia de evaluación es simple: se probaron los valores del algoritmo que podrían ser útiles en la práctica clínica para la segmentación de la aurícula izquierda. El tiempo máximo admisible se estimó en base al tiempo del que se dispone a la hora de realizar un procedimiento de ablación.

3. Resultados de la evaluación de OtsuMultipleThreshold

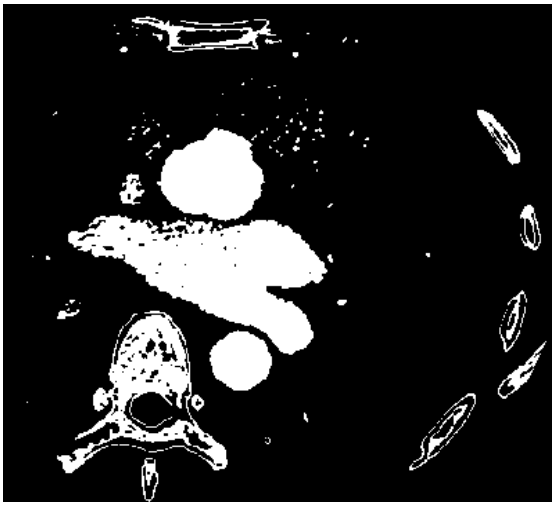
Los resultados que se representan son los obtenidos solamente con las imágenes con 16 bits de profundidad, dados los deficientes resultados obtenidos.

Imagen 1 - OtsuMultipleThreshold

Como ha sido discutido en secciones anteriores, el algoritmo genera N+1 segmentaciones para el valor del parámetro *NumberOfThresholds*=N. De esta manera, como las pruebas incluyen los valores N=1..6, se obtuvieron 27 segmentaciones para la evaluación del algoritmo. En la gráfica de la derecha se representan los valores de precisión y sensibilidad obtenidos con las 27 imágenes de salida totales.



El mejor resultado obtenido se produjo con el valor de umbral *NumberOfThresholds*=6. La imagen de la derecha muestra la mejor de las segmentaciones conseguidas, y más abajo en rojo se muestra la segmentación groundtruth, para que por comparación pueda observarse la pésima calidad de la segmentación obtenida. La tabla de abajo resume los valores de precisión y sensibilidad alcanzados en esta segmentación óptima.



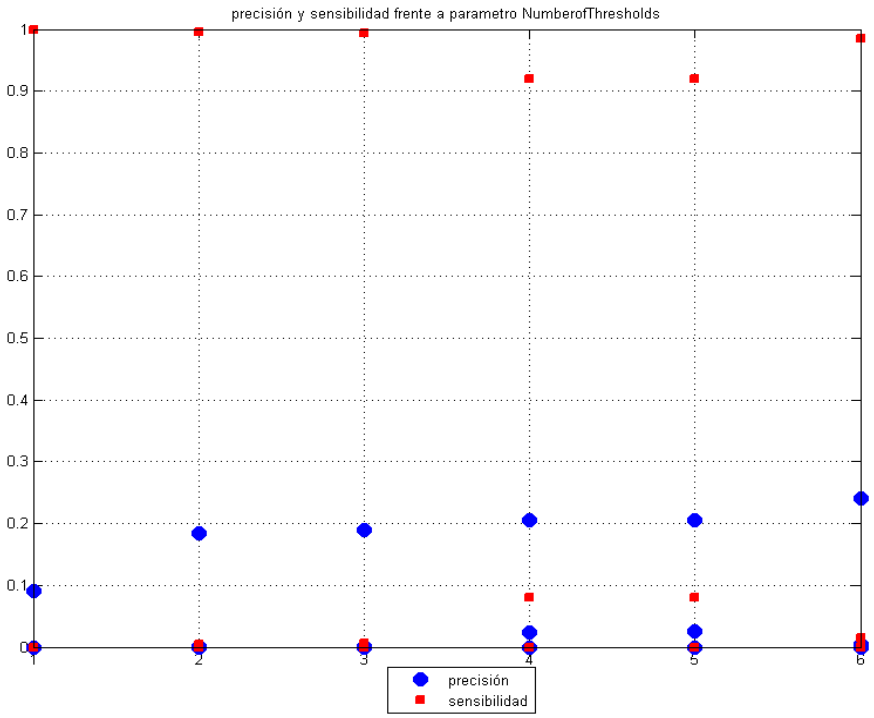
Segment. óptima	
Umbral	6
Precisión	0.42
Sensibilidad	0.91



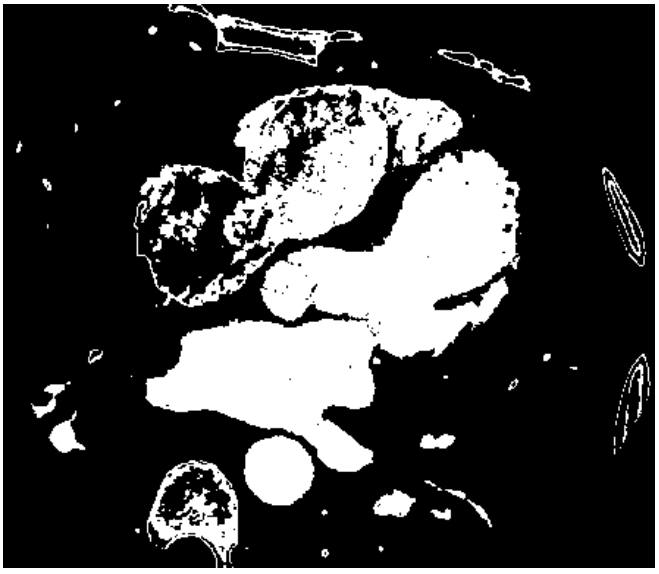
Al no existir zona estable, no se han calculado valores medios para la precisión y la sensibilidad.

Imagen 2 - OtsuMultipleThreshold

Al igual que se ha comentado para la imagen 1, el algoritmo *OtsuMultipleThreshold* genera N+1 segmentaciones para el valor del parámetro NumberOfThresholds= N. De esta manera, como las pruebas incluyen los valores N=1..6, se obtuvieron 27 segmentaciones para la evaluación del algoritmo. En la gráfica de la derecha se representan los valores de precisión y sensibilidad obtenidos con estas 27 imágenes. No se ha trazado la gráfica por tratarse de un conjunto de datos en el que múltiples ordenadas comparten una misma abscisa, es decir, para cada $N = n_i$ (siendo $n_i$el parámetro con que se llama al algoritmo) existen $n_i + 1$ pares de puntos $(n_i, p), (n_i, s)$.



Al no existir zona estable con resultados satisfactorios, no se han definido valores medios para la precisión y la sensibilidad. No obstante, el mejor resultado obtenido se produjo con el valor de umbral *NumberOfThresholds*=6. La imagen de la derecha muestra la imagen segmentada, y abajo se muestra para su comparación la segmentación groundtruth. La tabla de abajo muestra los valores de precisión y sensibilidad asociados a esta segmentación óptima, que como puede verse dista mucho de ser satisfactoria, o siquiera comparable con los otros algoritmos ya estudiados.

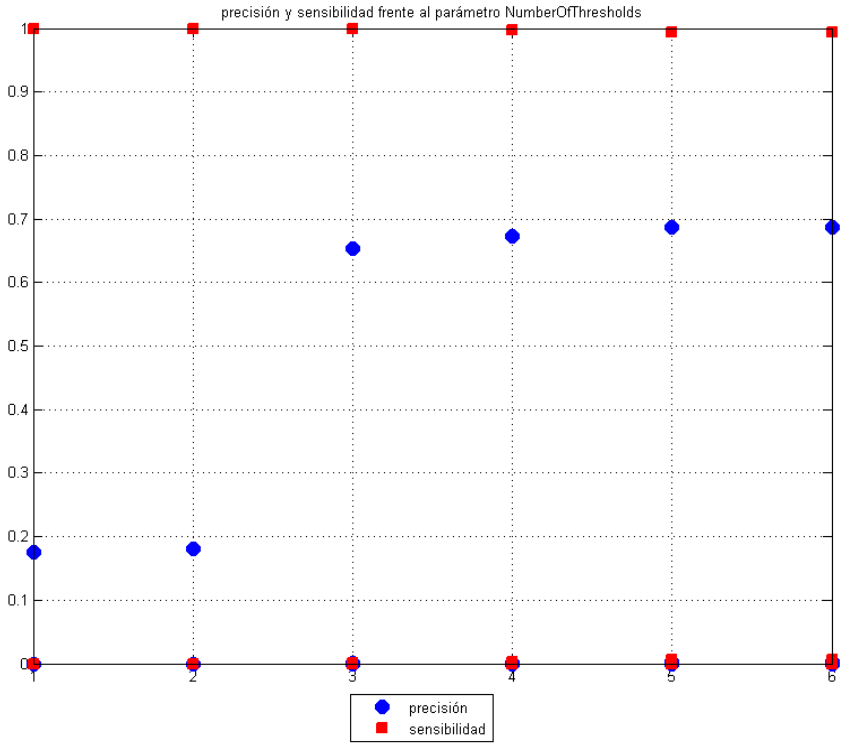


Segment. óptima	
Umbral	6
Precisión	0.24
Sensibilidad	0.9585635



Imagen 3 - OtsuMultipleThreshold

Como se ha expuesto para las imágenes 1 y 2, el algoritmo genera N+1 segmentaciones para el valor del parámetro *NumberOfThresholds*=N. De esta manera, como las pruebas incluyen los valores N=1..6, se obtuvieron 27 segmentaciones para la evaluación del algoritmo. En la gráfica de la derecha se representan los valores de precisión y sensibilidad obtenidos con estas 27 imágenes. No se han trazado las curvas al tratarse de un conjunto de datos multievaluados para la abscisa, al igual que se ha explicado con la imagen anterior.



Al no existir zona estable con resultados satisfactorios, no se han definido valores medios para la precisión y la sensibilidad. El mejor resultado obtenido se produjo con el valor de umbral *NumberOfThresholds*=6, cuya imagen se muestra en la imagen de la derecha. Más abajo se muestra la segmentación *groundtruth* en rojo para su comparación, asimismo los datos de la mejor segmentación se recogen en la tabla de abajo.



Segment. óptima	
Umbral	5 y 6
Precisión	0.686865
Sensibilidad	0.994544



7. Algoritmo ThresholdLevelSet. Evaluación

1. Análisis de espacio paramétrico

Los parámetros que este algoritmo requiere por parte del operador son 5, aunque pueden englobarse en cuatro conceptos:

- Semilla, situada en la estructura objeto de la segmentación.
- *InitialDistance*. Distancia desde la superficie hasta la semilla. Tras probar diversos valores para este parámetro y comprobar la débil dependencia con el mismo, tal y como se expone en la guía de usuario de ITK (8), se usó el valor inicial por defecto recomendado en dicho documento.
- *LowerThreshold*: Umbral inferior de intensidad que posee la zona de interés.
- *Upper Threshold*: Umbral superior de intensidad que posee la zona de interés.

Se calculó el tiempo de cómputo del algoritmo y se comprobó que no dependía de los parámetros arriba mencionados. Asimismo el algoritmo se evaluó con imágenes con profundidades de color de 8 y 16 bits. Los tiempos variaron, aunque poco significativamente. La Tabla 7 muestra dichos tiempos:

Niveles de gris	Tiempo medio
65536	218.77
256	217.14

Tabla 7: Diferencia de tiempos de ejecución 8-16bits resolución gris. Threshold Level Set.

Dada la escasa diferencia y la evidente mayor relevancia clínica de la segmentación con imágenes a 65536 niveles, la evaluación se efectuó solamente con estas últimas.

2. Estrategia de evaluación

Para cada imagen se llevó a cabo un proceso de tanteo para delimitar el espacio de parámetros a la zona no trivial. Los resultados que se obtuvieron se muestran en la Tabla 8:

	Rango de parámetros con salida no trivial
Imagen 1	Lower=[17000, 24000], Upper = [18000, 25000]
Imagen 2	Lower=[15000, 23000], Upper=[15000, 24000]
Imagen 3	Lower=[17000, 23000], Upper=[18000, 24000]

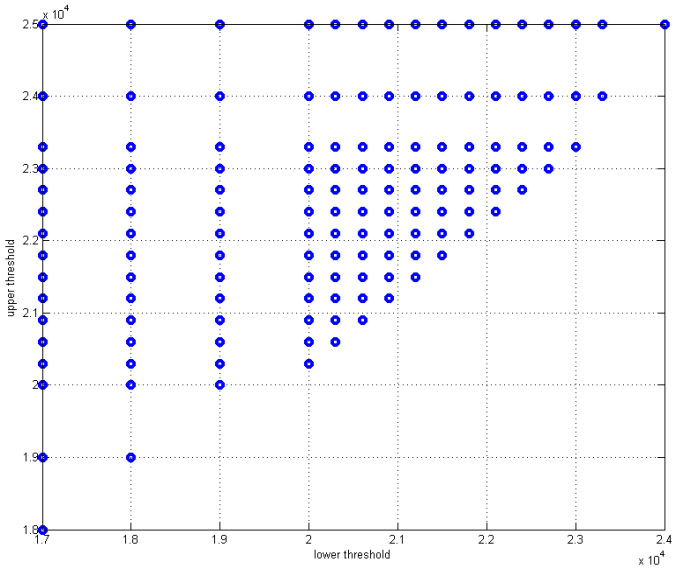
Tabla 8: Rango de parámetros con salida no trivial. ThresholdLevelSet.

3. Evaluación de resultados

Imagen 1 - ThresholdLevelSet

Para la imagen 1, el espacio de parámetros se reticuló dentro de la zona no trivial, ajustando una mayor resolución en la zona con mejores resultados, y cuidando que el tiempo de cómputo total no se disparara. La retícula resultante constó de 136 pares, y requirió un tiempo de cómputo global de unas 9 horas aproximadamente.

En la gráfica de la derecha se representa la retícula que sirvió para evaluar el algoritmo con la imagen 1.



La precisión y la sensibilidad calculadas a partir de las 136 segmentaciones procesadas se representan en la siguiente gráfica (Ilustración 19). En la superficie que se observa, la altura (valor del eje Z) representa el valor de la precisión. Y el color proyectado sobre la misma representa la sensibilidad.

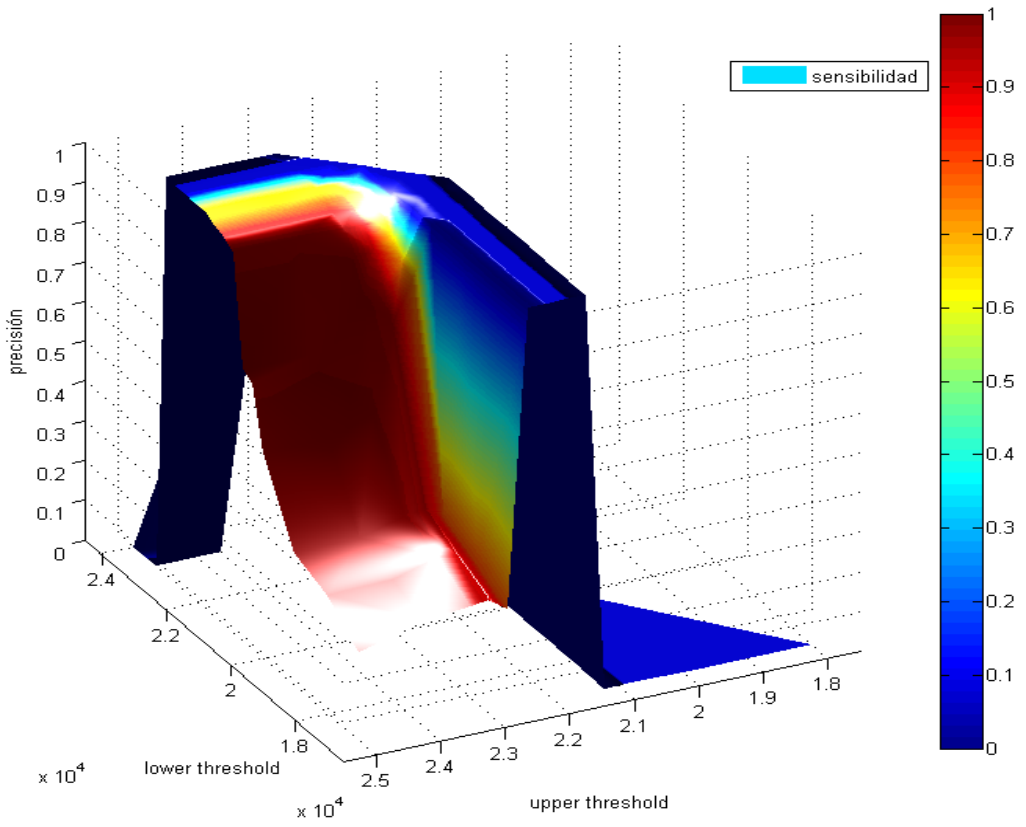


Ilustración 19: Precisión y sensibilidad ThresholdLevelSet. Imagen 1

Como ya se ha ido viendo durante la evaluación del resto de algoritmos, la zona estable del algoritmo es aquella cuyos valores de precisión y sensibilidad cumplen dos criterios a la vez; máxima altura (p>0.9, precisión máxima) y colores naranja hacia rojo (s>0.9, sensibilidad máxima). Si se observa la superficie, y los datos numéricos discretos con los que se construyó, puede verse que es verdaderamente pequeña (en términos de rango de parámetros) la zona de estabilidad. En la Ilustración 20 podemos ver una ampliación de la gráfica anterior (Ilustración 19), en la que pueden distinguirse los valores numéricos discretos de la precisión y la sensibilidad. El resto es la interpolación realizada al representar los datos a través de una superficie.

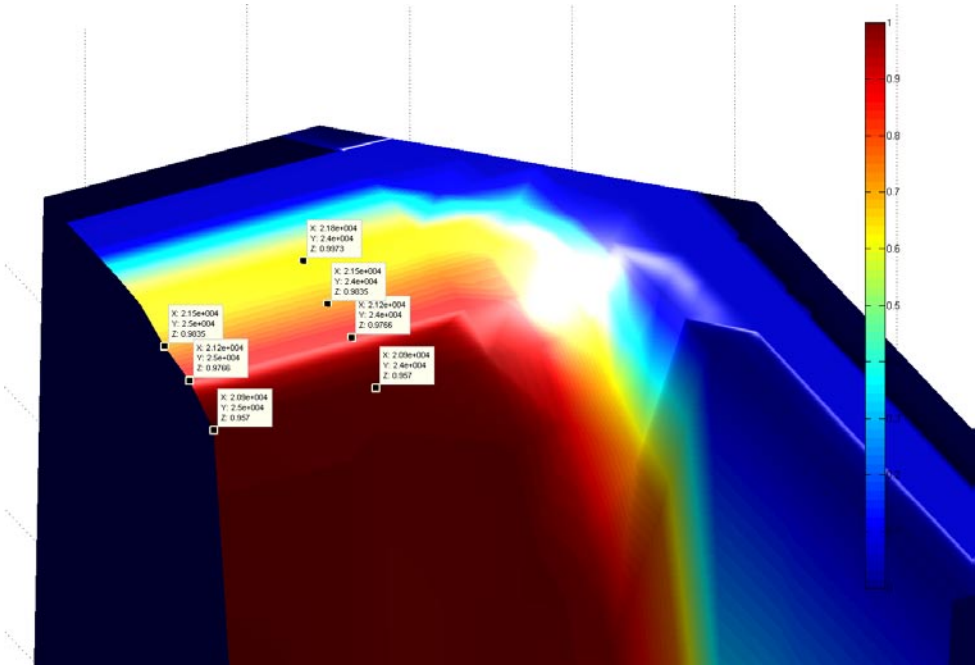


Ilustración 20: Ampliación de la zona estable ThresholdLevelSet. Imagen 1

Como podrá apreciarse, existen tan solo 6 combinaciones de entre las 136 ensayadas con valores propios de la zona estable (precisión y sensibilidad por encima del 90%). Las características de la precisión y sensibilidad en dicha zona se muestran en la tabla de la derecha.

Lower threshold	Upper threshold	Precisión	Sensibilidad
20600	23300	0,901839	0,968974
20900	23300	0,956602	0,956994
20900	**24000**	**0,957046**	**0,967336**
20900	25000	0,957046	0,967336
21200	24000	0,976626	0,902724
21200	25000	0,976626	0,902724

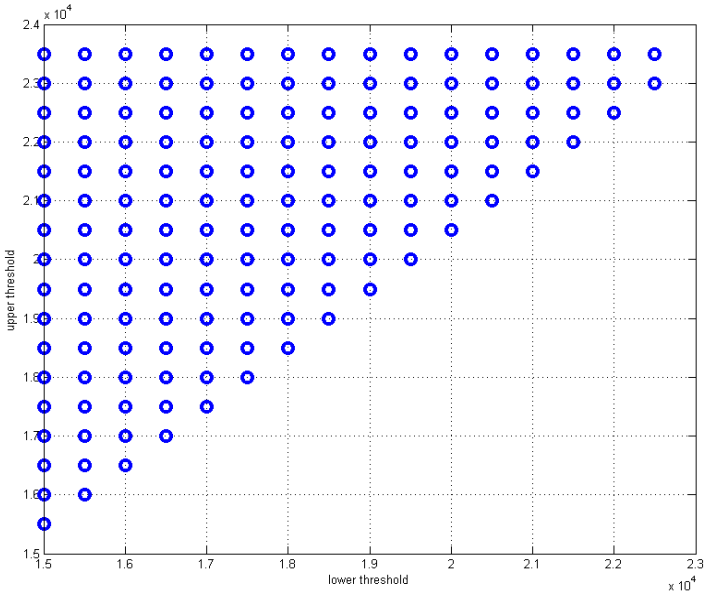
Zona estable	
Precisión máxima	0.9766
Sensibilidad máxima	0.968974
Precisión media	0.95429
Sensibilidad media	0.94434

A fin de poder comparar las características de los algoritmos en sus zonas estables, la tabla de la izquierda detalla unos estadísticos simples a partir de la tabla anterior. El valor de los parámetros con los que se obtuvieron las mejores precisión y sensibilidad conjuntamente se ha resaltado en la tabla superior. Las imágenes de abajo representan la segmentación óptima realizada por el algoritmo *thresholdlevelset* (en blanco), y la segmentación *groundtruth* en rojo.



Imagen 2 - ThresholdLevelSet

Para la imagen 2, el espacio de parámetros se reticuló dentro de la zona no trivial, cuidando que el tiempo de cómputo total no se disparara dadas las numerosas combinaciones posibles. Con la retícula resultante se obtuvieron 152 pruebas distintas del algoritmo, que requirieron un tiempo de cómputo global de unas 10 horas aproximadamente. En la gráfica de la derecha se representa la retícula regular que sirvió para evaluar el algoritmo con la imagen 2.



La precisión y la sensibilidad calculadas a partir de las 152 segmentaciones procesadas se representan en la Ilustración 21. Al igual que para la imagen 1, en la superficie representada, la altura de la misma (valor del eje Z) representa el valor de la precisión. Y el color proyectado sobre la misma representa la sensibilidad. Los valores de la zona estable del algoritmo son aquellos con máxima altura, cercana a uno, y con sensibilidad máxima, es decir, tendiendo al color rojo. Si se observa la superficie, y los datos numéricos discretos con los que se construyó, puede verse que es verdaderamente pequeña (en términos de rango de parámetros) la zona de estabilidad.

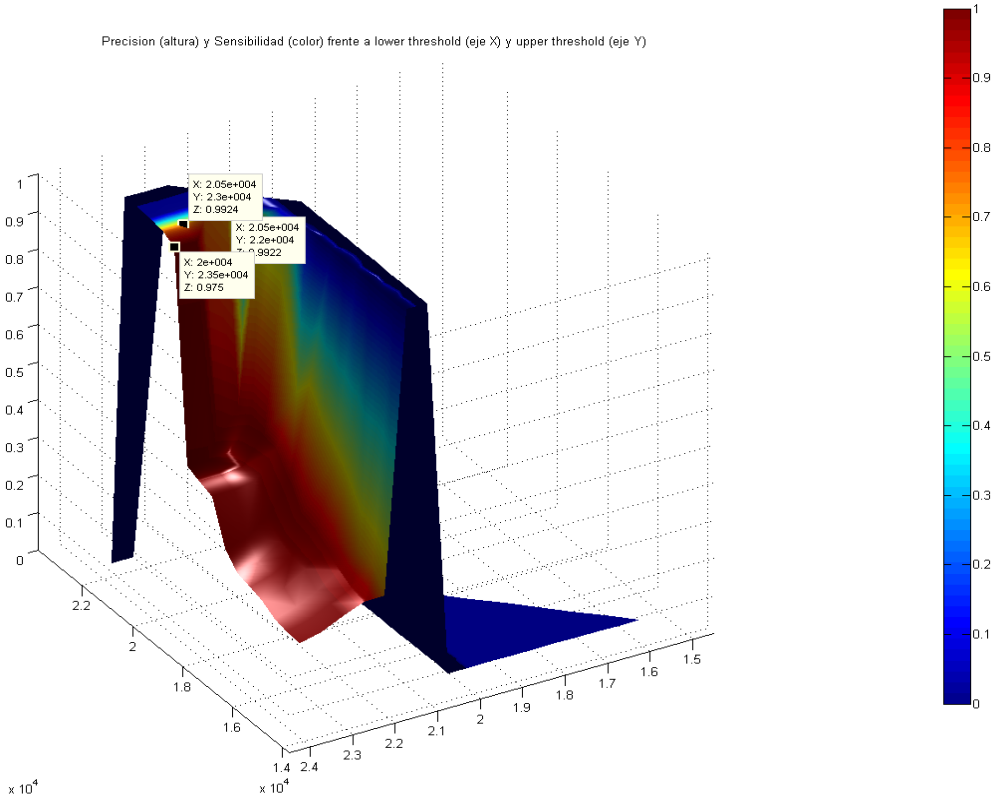


Ilustración 21: Zona estable precisión y sensibilidad. ThresholdLevelSet. Imagen 2

En la Ilustración 22 podemos ver una ampliación de la gráfica anterior, en la que pueden distinguirse los valores numéricos precisión y sensibilidad.

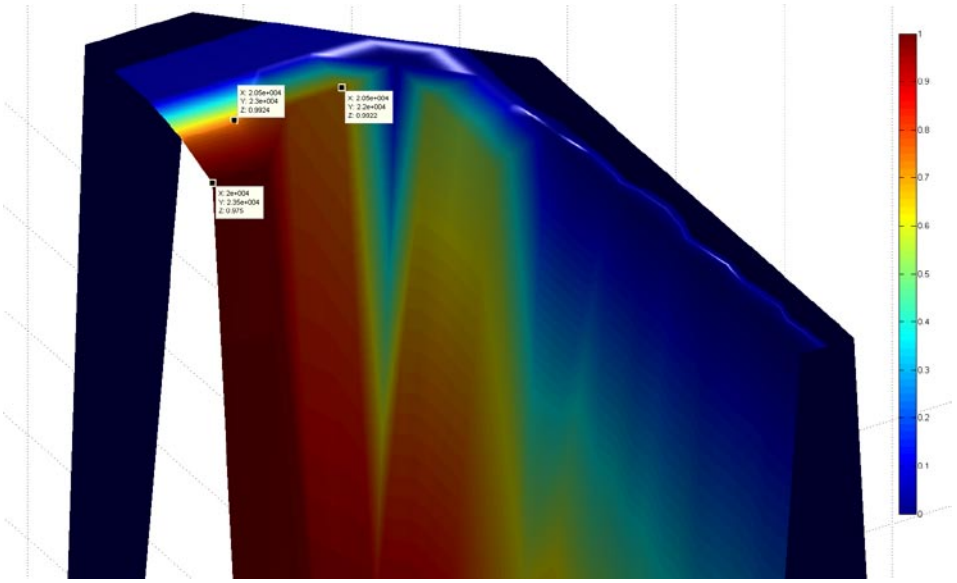


Ilustración 22: Ampliación de la zona estable. ThresholdLevelSet. Imagen 2

Como podrá apreciarse, existen tan solo 2 combinaciones de entre las 152 ensayadas con valores dentro de la zona estable (precisión y sensibilidad por encima del 90%). La tabla de la derecha contiene las características de la precisión y sensibilidad en dicha zona. Se ha resaltado en negrita los valores obtenidos en la segmentación óptima.

Lower threshold	Upper threshold	Precisión	Sensibilidad
20000	23000	0,972925	0,962344
20000	**23500**	**0,974951**	**0,962446**

Zona estable	
Precisión máxima	0.974951
Sensibilidad máxima	0.962446
Precisión media	0.973938
Sensibilidad media	0.962395

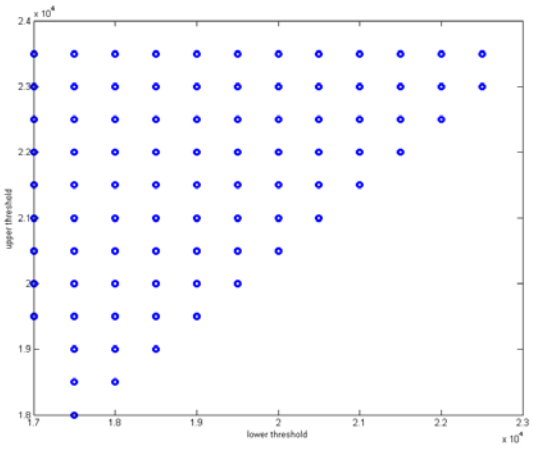
A fin de poder comparar las características de los algoritmos en sus zonas estables, la tabla de la izquierda muestra unos estadísticos simples a partir de la tabla anterior:

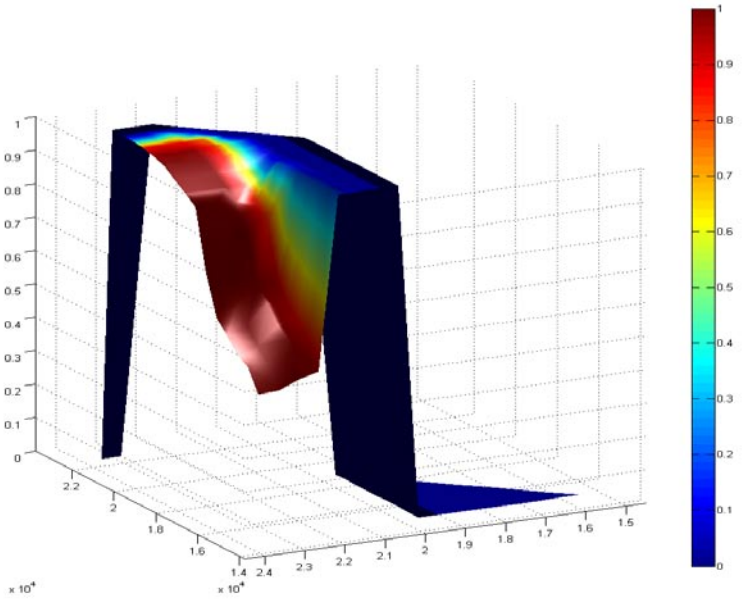
Debajo se muestran la segmentación óptima realizada con *ThresholdLevelSet* (en blanco) y la segmentación *groundtruth* (en rojo).



Imagen 3 - ThresholdLevelSet

Para la imagen 3, el espacio de parámetros se reticuló dentro de la zona no trivial, con la precaución de que el tiempo de cómputo total no se disparara, dadas las numerosas combinaciones posibles. La retícula regular resultante constó de 152 pruebas, y requirió un tiempo de cómputo global de unas 10 horas aproximadamente. En la gráfica de la derecha se representa el conjunto de valores que conforma dicha retícula.





La precisión y sensibilidad calculadas a partir de las 152 segmentaciones procesadas se representan en la gráfica de la izquierda. En la superficie representada, la altura de la misma (valor del eje Z) representa el valor de la precisión. Y el color proyectado sobre la misma representa la sensibilidad. Los valores de la zona estable del algoritmo son, al igual que en las evaluaciones presentadas para las imágenes 1 y 2, aquellos con máxima altura, cercana a uno, y con sensibilidad máxima, es decir, tendiendo al color rojo. Si se observa la superficie, y los datos numéricos discretos con los que se construyó, puede verse que es bastante pequeña (en términos de rango de parámetros) la zona de estabilidad.

Entre los datos obtenidos existen 12 pares de entre los 152 ensayados con valores de precisión y sensibilidad que podemos considerar satisfactorios (precisión y sensibilidad por encima del 90%). La tabla de la derecha recoge las características de la precisión y sensibilidad en la zona estable. En negrita y subrayado se representa la segmentación óptima de entre todas las que conforman dicha zona.

Zona estable	Precisión	Sensibilidad
Máxima	0.9866	0.9991
Media	0.959	0.9866

Lower threshold	Upper threshold	Precisión	Sensibilidad
18500	22500	0,926258	0,990484
18500	23000	0,926133	0,998224
18500	23500	0,925922	0,999112
19000	22500	0,950449	0,988073
19000	23000	0,950115	0,996828
19000	23500	0,950091	0,997526
19500	22500	0,973856	0,985409
19500	23000	0,973558	0,992704
19500	23500	0,973062	0,994544
20000	22500	0,986688	0,968661
20000	23000	0,986595	0,975829
20000	**23500**	**0,986538**	**0,976274**

A fin de poder comparar las características de los algoritmos en sus zonas estables, se presentan en la tabla de la izquierda unos estadísticos simples sobre la zona estable.

8. Comparativa de los algoritmos estudiados

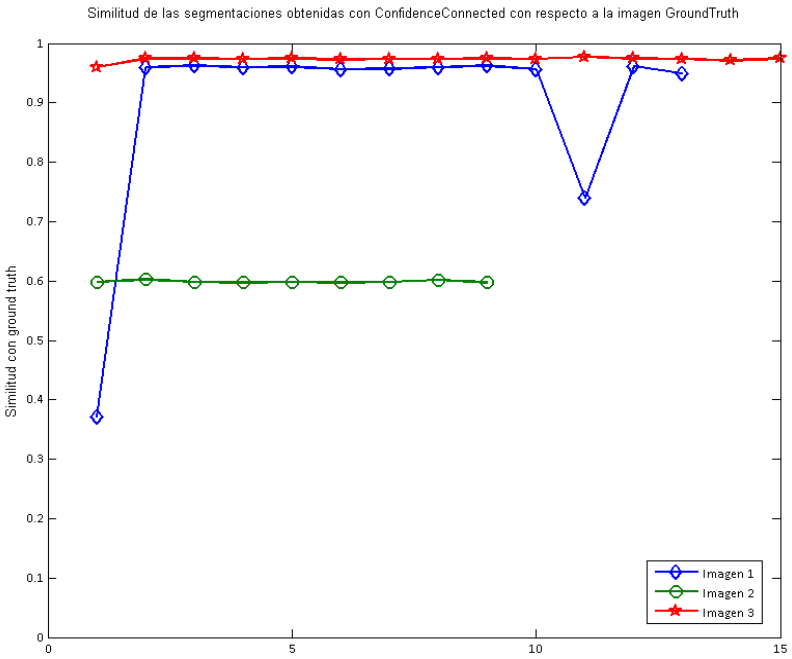
1. Compilación de resultados

En esta sección se lleva a cabo la comparativa de los algoritmos cuya evaluación se ha mostrado durante las secciones anteriores. Para ello, dada la cantidad importante de datos, se hace necesario en primer lugar compilar y condensar toda la información obtenida con las distintas evaluaciones realizadas. La Tabla 9 presenta una compilación de los resultados obtenidos.

		Confiden Connected	**Isolated Connected**	**Otsu Multiple Threshold**	**Threshold LevelSet**
	Parámetros	*SeedIn*	*SeedIn* *SeedOut* *Threshold*	*NumberOfThresholds*	*SeedIn* *Lower thr* *Upper thr*
	Segmentaciones obtenidas	37	300	18 (81 segmentaciones)	430
Imagen 1	¿Zona estable?	Sí	Sí	No	Sí
	Anchura zona estable (sobre eje de pruebas)	84% (11/13)	13%	--	4.7% (6/126)
	Número pruebas	13	100	6. (27 segmentaciones)	126
	Medias (p, s) zona estable	(0.967, 0.952)	(0.97144, 0.9427)	--	(0.95429, 0.94434)
	Similitud media zona estable	0.9588	0.9547	--	0.9491
	(p,s) máximas	(0.964, 0.96)	(0.9609, 0.9693)	(0.42, 0.91)	(0.957046, 0.967336)
	Similitud máxima	0.9619	0.9648	0.5850	0.9618
	Tiempo medio iteración (segs)	1 seg	1.2 seg	Exponencial con parámetro number of thresholds	218.77 segs
	Tiempo global	14 segs	130 segs	2h 30min	7h 40min
	Complejidad operador	1	2	1	3
Imagen 2	¿Zona estable?	No. Zona estable redefinida: p>0.4, s>0.9	Sí	No	Sí
	Anchura zona estable (sobre eje de pruebas)	0/9	4%	--	1.3% (2/152)
	Número de pruebas	9	100	6 (27 segmentaciones)	152
	Medias (p, s) zona estable	(0.433, 0.9699)	(0.9881, 0.9242)	--	(0.973938, 0.962395)
	Similitud media zona estable	0.5985	0.9457	--	0.9624
	(p,s) maximas	(0.439, 0.95)	(0.9876, 0.9333)	(0.24, 0.958)	(0.974951, 0.962446)
	Similitud máxima	0.6017	0.9520	0.4618	0.9681
	Tiempo medio iteración	1 seg	1.2 segs	Exponencial	218.77 segs
	Tiempo global	10 segs	130 segs	2h 30 min	10h aprox.
	Complejidad operador	1	2	1	3
Imagen 3	¿Zona estable?	Sí	Sí	No	Sí
	Anchura zona estable (sobre eje de pruebas)	100% (15/15)	44%	--	7.8% (12/152)
	Número de pruebas	15	100	6 (27 segmentaciones)	152
	Medias (p, s) zona estable	(0.9915, 0.9630)	(0.959, 0.987)	--	(0.959, 0.9866)
	Similitud media	0.9732	0.9694	--	0.9695
	Mejor segmentación alcanzada	(0.9917, 0.9618)	(0.9861, 0.9861)	(0.6868, 0.9945)	(0.986538, 0.9756274)
	Similitud máxima	0.9724	0.9861		0.9803
	Tiempo medio iteración	1	1.2	Exponencial	218.77
	Tiempo global	15 segs	130 segs	2h 30 min	10h aprox.
	Complejidad operador	1	2	1	3

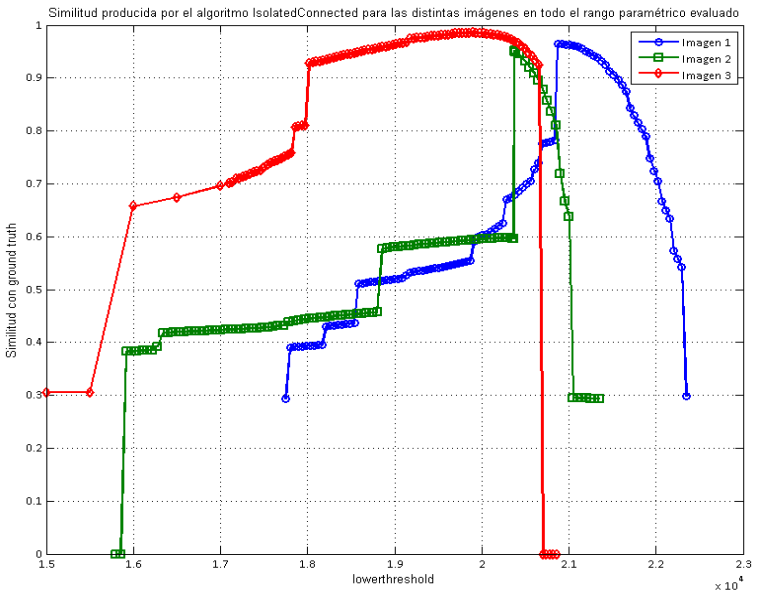
Tabla 9: Compilación de resultados de la evaluación

ConfidenceConnected



En esta primera gráfica de la derecha se representa la similitud de las segmentaciones obtenidas con el algoritmo **ConfidenceConnected** con cada una de las 3 imágenes empleadas para la evaluación. La utilidad de la similitud es agrupar precisión y sensibilidad en una sola variable, de manera que se pueden comparar fácilmente los resultados obtenidos por un algoritmo con más de una imagen. En este caso, se puede apreciar la zona de estabilidad conseguida con cada imagen para los distintos parámetros ensayados. Pueden extraerse las conclusiones que se enunciaron en los apartados anteriores correspondientes, pero en una sola gráfica. A modo de ejemplo, de la ilustración de la derecha se puede obtener de manera directa leyendo la línea verde, (etiquetada como perteneciente a la imagen 2 en la leyenda de la gráfica) que el algoritmo ConfidenceConnected no consiguió en ningún caso buenos valores de similitud con ninguna semilla probada. En contraste, pueden verse la buena similitud obtenida con la imagen 3 para cualquier valor de semilla especificado.

IsolatedConnected

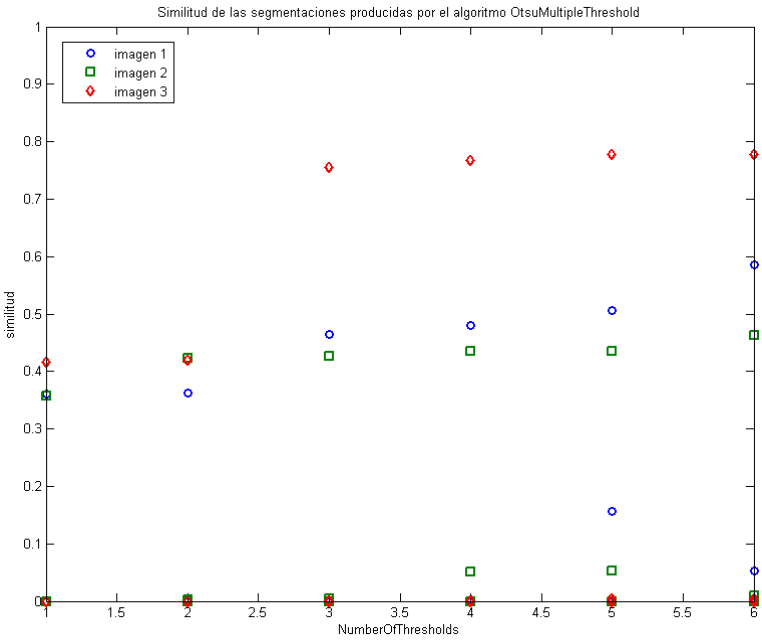


A la derecha se representan las gráficas de similitud calculadas con las segmentaciones producidas por el algoritmo IsolatedConnected. Pueden apreciarse las zonas de estabilidad (puesto que similitud > 0.9 se corresponde aproximadamente en el plano (p,s) con (p>0.9 & s>0.9)) obtenidas con cada imagen, y el ancho de las mismas. Puede inferirse por ejemplo que se pueden conseguir mejores resultados que con ConfidenceConnected para la imagen 3, aunque para un rango del parámetro *LowerThreshold* muy reducido.

OtsuMultipleThreshold

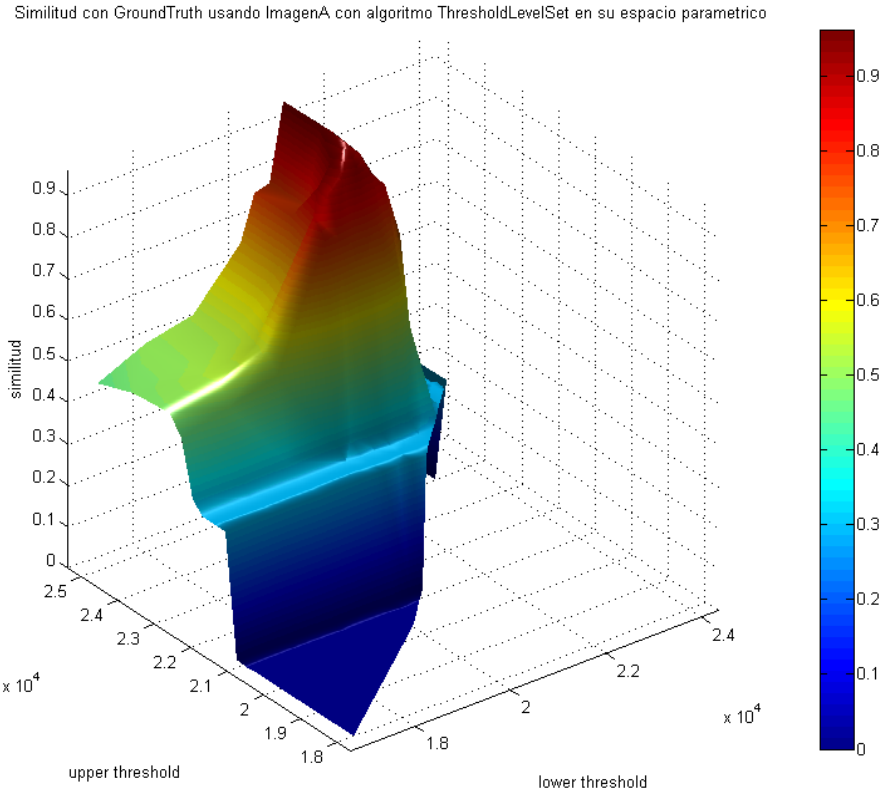
La figura de la derecha representa la similitud de las segmentaciones obtenidas con el algoritmo *OtsuMultipleThreshold* para cada una de las 3 imágenes con que se ha evaluado. Puede leerse que los resultados no son satisfactorios para ninguna imagen, independientemente del parámetro especificado.

No se han trazado curvas sobre los puntos dado que a una misma abscisa le corresponden varios valores de ordenadas.



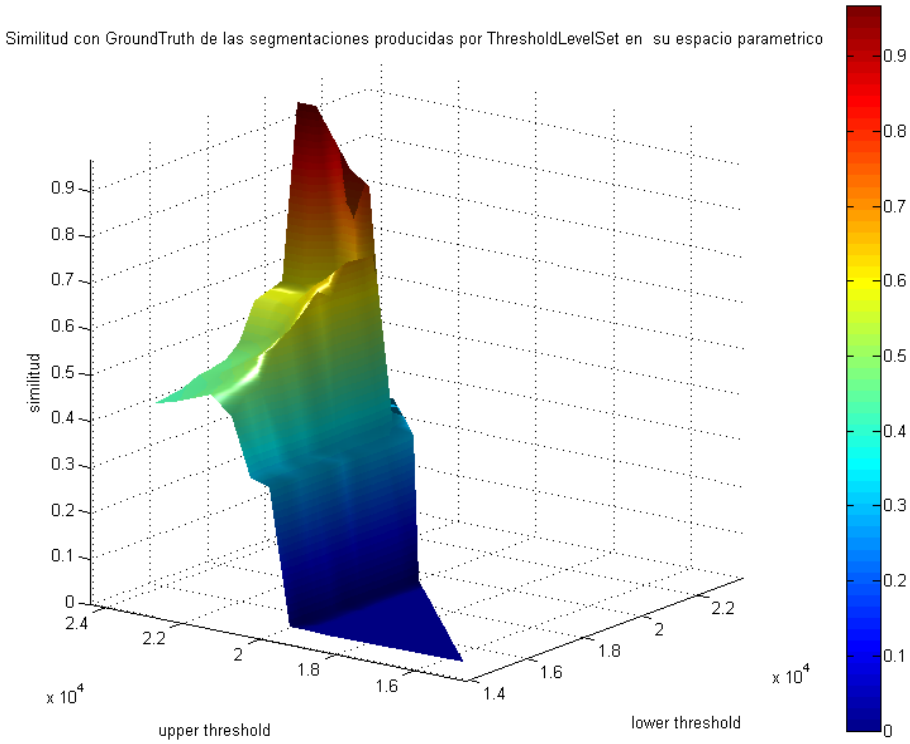
ThresholdLevelSet – Imagen 1

En la figura de la derecha se representa la similitud del algoritmo *Theshold Level Set* obtenido con la imagen 1. Puede verse el abrupto pico rojo, indicando que se consiguen buenos resultados para muy pocos valores de umbral. En este caso, tanto el color como la altura de la superficie representan el valor de la similitud.



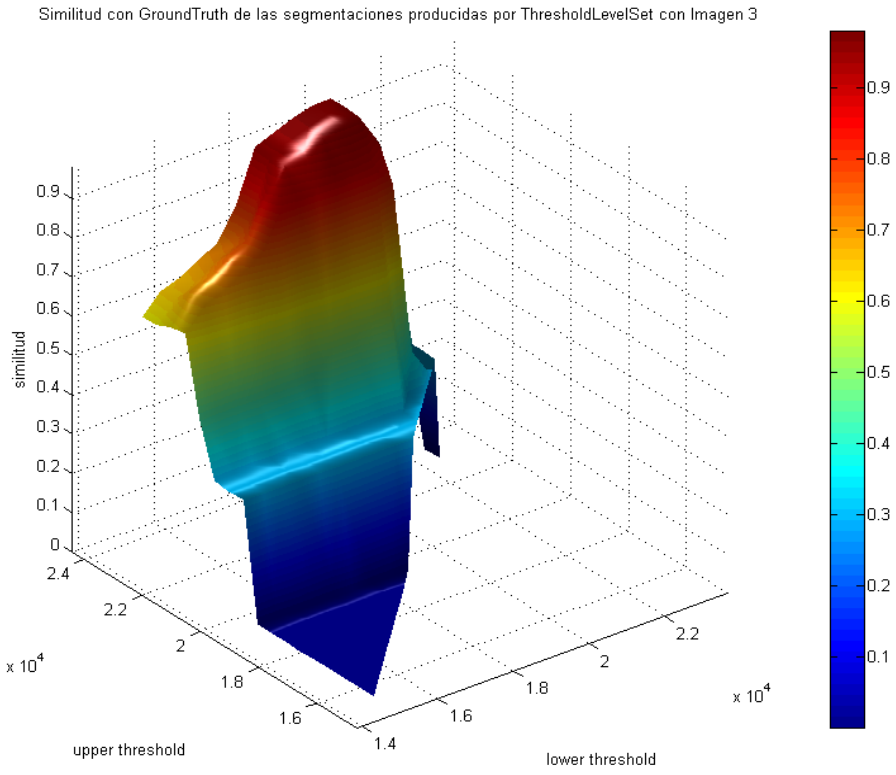
ThresholdLevelSet – Imagen 2

En la figura de la derecha se ha representado la similitud del algoritmo *Threshold Level Set* a partir de las segmentaciones obtenidas para la imagen 2. Puede verse un pico muy alto pero muy abrupto en la similitud, indicando un buen resultado obtenidos solamente para un rango muy pequeño de parámetros de entrada.



ThresholdLevelSet – Imagen 3

En la figura de abajo sin embargo, que representa la similitud del algoritmo *Threshold Level Set* para la imagen 3, puede verse un pico menos abrupto que en los otros casos, indicando así que la zona estable es algo mayor que con las otras dos imágenes probadas.



2. Presentación de segmentaciones óptimas. Comparativa visual.

A continuación, como paso previo a la discusión y comparativa de los distintos algoritmos evaluados, se representa el conjunto de segmentaciones óptimas conseguidas con cada algoritmo, secuencialmente para cada una de las 3 imágenes usadas para la evaluación, junto con la imagen TAC original correspondiente y la segmentación manual o *groundtruth*. La Ilustración 23 presenta las segmentaciones de la imagen 1, la Ilustración 24 las de la imagen 2, y la Ilustración 25 las de la imagen 3.

Imagen 1

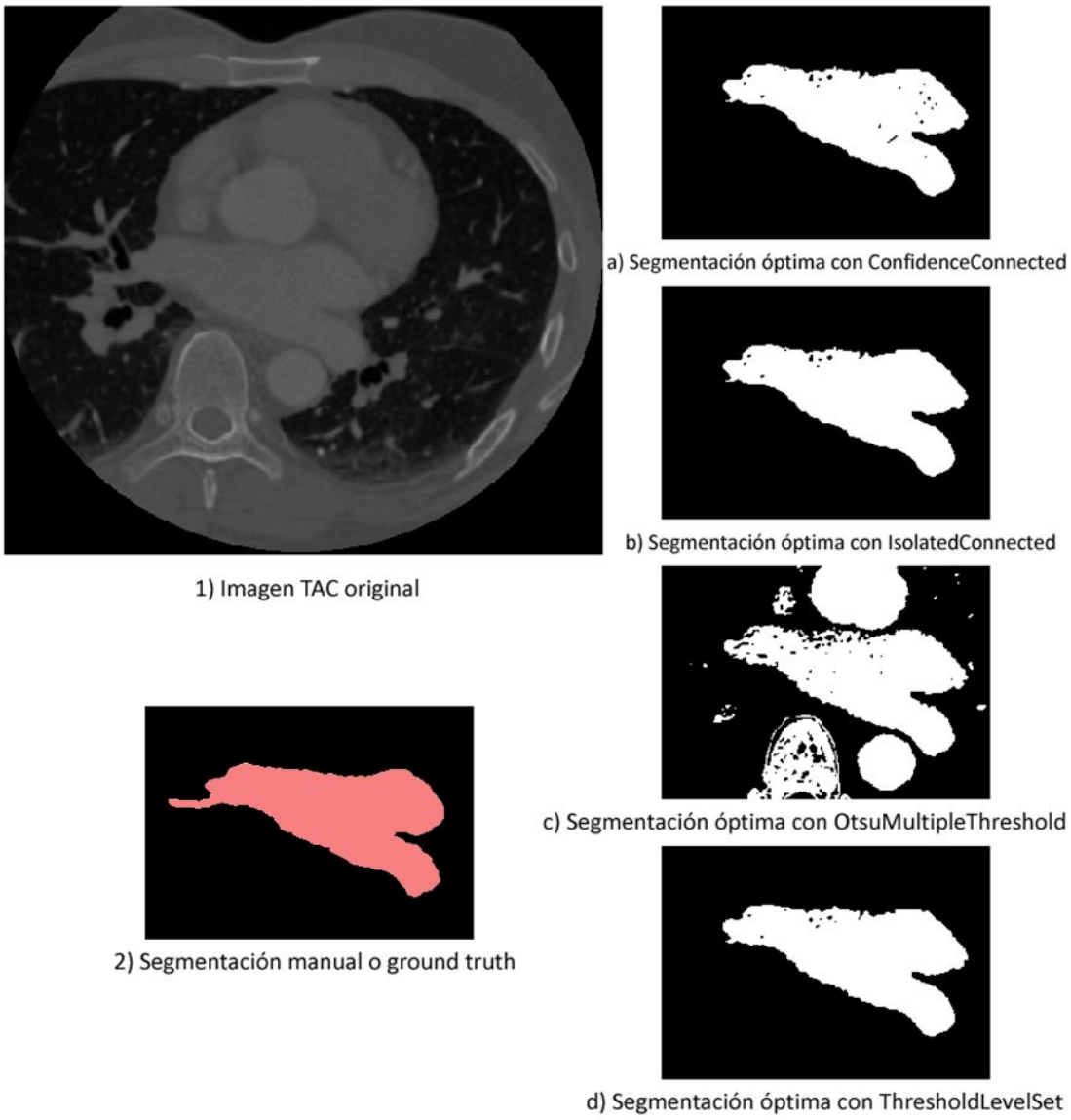


Ilustración 23: Segmentaciones óptimas de los distintos algoritmos para Imagen 1

Imagen 2

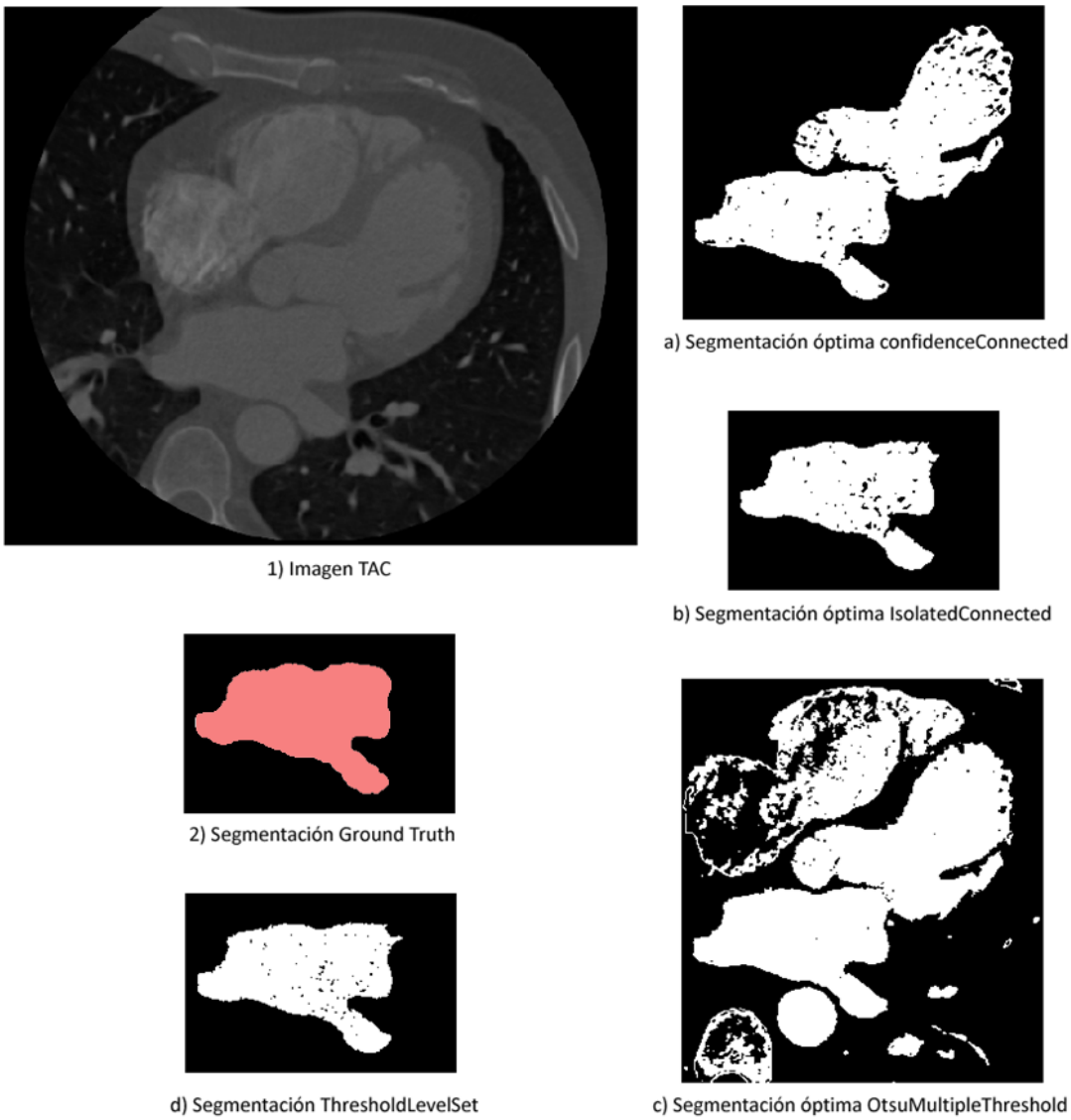


Ilustración 24: Segmentaciones óptimas de los distintos algoritmos para Imagen 2

Imagen 3

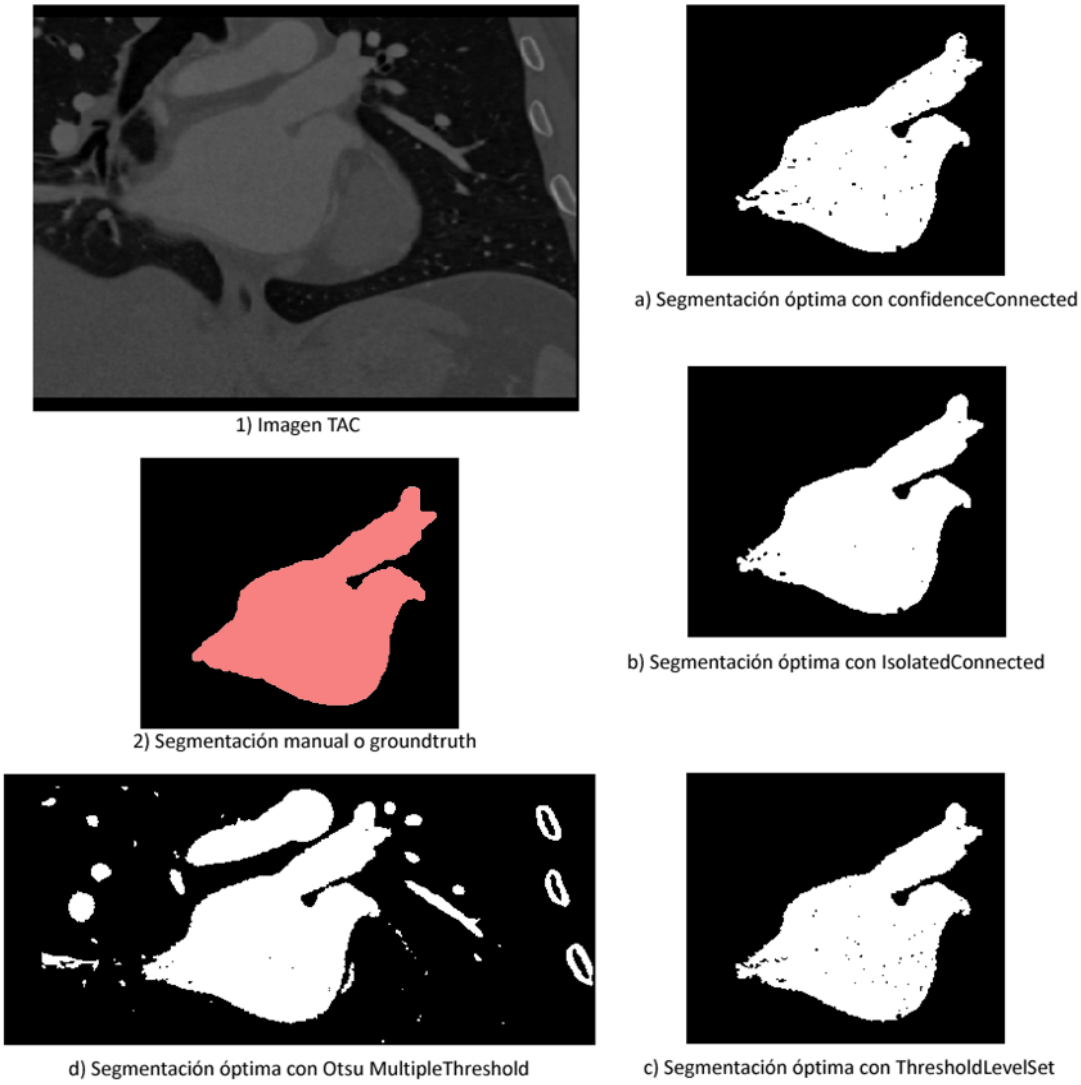


Ilustración 25: Segmentaciones óptimas de los distintos algoritmos para Imagen 3

3. Comparativa. Conclusiones

En la Tabla 10 se muestra el algoritmo que mejor desempeño ha mostrado en la categoría indicada por la fila, y para la imagen determinada por la columna. Cuando la diferencia entre los desempeños ha sido muy pequeña (de menos del 2% sobre el máximo), se ha incluido más de un algoritmo, ordenados de mejor desempeño a peor desempeño. La abreviatura *CC* designa al algoritmo *ConfidenceConnected*, *IC* designa a *IsolatedConnected*, *TLS* designa a *ThresholdLevelSet* y *OT* designa a *OtsuMultipleThreshold*. Como ejemplo, si en una categoría determinada, los algoritmos *CC*, *IC*, *TLS*, *OT* muestran unas similitudes del 98, 97, 96 y 24%, respectivamente, la entrada correspondiente de la tabla recogería el valor siguiente (en el mismo orden): *CC* (0.98) – *IC* (0.97) – *TLS* (0.97).

	Imagen 1	**Imagen 2**	**Imagen 3**
Similitud máxima	IC (0.9648) CC (0.9619) TLS (0.9618)	TLS (0.9681) IC (0.952)	IC (0.9681) TLS (0.9803) CC (0.9724)
Similitud media (en zona estable)	CC (0.9588) IC (0.9427) TLS (0.9491)	TLS (0.9681) IC (0.952)	CC (0.9732) TLS (0.9695) IC (0.9694)
Ancho zona estable	CC	IC	CC
Eficiencia	CC (1s) IC (1.2s)	IC (1.2s)	CC (1s) IC (1.2s)
Complejidad	De – a + complejidad: CC IC TLS	De – a + complejidad: CC IC TLS	De – a + complejidad: CC IC TLS

Tabla 10: Comparativa global

Para identificar fácilmente el mejor algoritmo en cada categoría, se ha asignado un color distinto a cada uno cuando han alcanzado el mejor valor en cualquier categoría. Así, cuando **ConfidenceConnected** ha sido el mejor en algún aspecto con alguna imagen se ha representado en verde, **IsolatedConnected** en rojo, y **ThresholdLevelSet** en azul. A continuación se pasa a discutir el desempeño global de cada algoritmo, en orden de mejor a peor desempeño global.

ConfidenceConnected

Puede concluirse que se trata de un algoritmo globalmente bastante apropiado para la tarea, no exento de algunos inconvenientes. Entre sus grandes ventajas, independientemente de la imagen empleada, están su gran eficiencia, la mayor de todos los algoritmos evaluados (seguida muy de cerca por *IsolatedConnected*), y su gran estabilidad, cuando ha existido zona estable. También ha demostrado alcanzar los mejores valores medios de similitud (y por tanto de precisión y sensibilidad) con la imagen groundtruth (seguido de cerca por *IsolatedConnected* y *ThresholdLevelSet*) dentro de su zona estable, **cuando ésta ha existido**.

Y esa precisamente ha demostrado ser su gran desventaja. Cuando ha habido cámaras adyacentes mal separadas (bien en niveles de intensidad, o bien por escasa frontera anatómica o frontera no continua del todo), el algoritmo no ha podido establecer segmentaciones aceptables independientemente de la semilla escogida.

IsolatedConnected

Ha demostrado un desempeño muy similar a *ConfidenceConnected*, perdiendo en similitudes medias frente a *ConfidenceConnected*, pero ganándole en valores máximos. La eficiencia también es muy similar (diferencia despreciable). Por si esto fuera poco, ha demostrado no tener el problema más importante que se observó con *ConfidenceConnected*. *IsolatedConnected* siempre ha sido capaz de alcanzar zona estable, por estrecha que fuera.

A cambio, *IsolatedConnected* añade una bastante moderada dificultad al proceso de segmentación, pues requiere especificar *SeedOut* y *Lowerthreshold* (en comparación con *ConfidenceConnected*), y necesariamente hace más estrecha la zona estable, puesto que el algoritmo es bastante sensible al parámetro *LowerThreshold* (aunque no tanto como *ThresholdLevelSet*).

ThresholdLevelSet

Es un algoritmo que no destaca por su eficiencia (220 veces menos eficiente que *ConfidenceConnected*), y que aumenta sensiblemente la dificultad al proceso de segmentación, puesto que el operador necesita especificar *Lower* y *Upper Thresholds*, siendo el resultado bastante dependiente del buen ajuste de estos. La dependencia con el parámetro *Seed* es inapreciable. Con todas las imágenes ha producido una muy estrecha zona estable pero con buenos valores medios y máximos de similitud, aunque no los mejores. En el único caso en que este algoritmo mostró mejor desempeño en similitud fue con la imagen 2. En este caso, tanto la similitud media como máxima fue obtenida con éste, aunque seguido de cerca por *IsolatedConnected*.

Sus puntos débiles son la eficiencia pequeña y la estrecha estabilidad. Su punto fuerte es conseguir el mejor resultado con imágenes más complicadas para el resto de algoritmos.

OtsuMultipleThreshold

Este algoritmo no ha podido competir con los otros en ningún aspecto, quedando descartado como primera opción para la segmentación de la aurícula izquierda en las condiciones evaluadas en este proyecto.

Valoración global y conclusiones.

Por su simplicidad y eficiencia, el algoritmo *ConfidenceConnected* es aconsejable para una segmentación rápida y sencilla. Para los casos en los que ConfidenceConnected no consigue buenos resultados, (como por ejemplo cuando aurícula y ventrículo aparecen con una separación mal definida) IsolatedConnected sería una mejor opción a considerar. Se tendría una segmentación algo más compleja, por tener que ajustar el parámetro *LowerThreshold*, pero se llegarían a buenos resultados. La buena eficiencia de este algoritmo permite también agilidad en el proceso de ensayo-error para el ajuste de parámetros.

En el contexto de este proyecto, con el sistema de evaluación usado, y con las imágenes ensayadas, no se han detectado casos en los que el uso de *ThresholdLevelSet* sea la primera opción a manejar. La diferencia de desempeño con IsolatedConnected es ínfima, y sin embargo este último algoritmo rebaja significativamente el tiempo del proceso de segmentación y la complejidad del ajuste de parámetros que implica *ThresholdLevelSet*. El algoritmo *OtsuMultipleThreshold* es el que ha mostrado peor desempeño para su uso en la segmentación de la aurícula izquierda, no habiendo conseguido en ningún caso resultados aceptables.