

# 5. Resultados

En este capítulo indicaremos las bases de datos que hemos utilizado para hacer el proyecto, el programa y mostraremos los resultados obtenidos tras aplicar los algoritmos de segmentación.

## 5.1 Bases de datos

Para demostrar la validez de los algoritmos de segmentación diseñados realizaremos las pruebas en imágenes provenientes de Tomografía Axial Computerizada (TAC).

La creciente utilización de sistemas de adquisición y tratamiento digital de imágenes médicas ha hecho necesaria la adopción de estándares que posibiliten el intercambio de éstas tanto dentro de las propias instituciones como fuera de ellas, por lo que trabajaremos con imágenes TAC en formato Dicom.

El estándar DICOM 3.0 nace en el año 1993, a partir de un rediseño completo de la publicación normalizada No 300-1988 de ACR-NEMA (fusión de American College of Radiology, ACR y el National Electrical Manufacturers Association, NEMA) y pertenece al campo de la Informática Médica.

DICOM (Digital Imaging and Communications in Medicine) es el estándar industrial para transferencia de imágenes digitales e información médica entre computadoras. Dicom permite la comunicación digital entre equipos de diagnóstico, terapéuticos y entre sistemas de diferentes fabricantes. Los objetivos de este estándar son:

- Logran un interfaz común para todos los dispositivos de imágenes (tomografía, resonancia magnética, ultrasonido, rayos x, etc).
- Debe ser aplicable a toda la esfera de las imágenes médicas, desde su transmisión hasta su tratamiento e impresión.

Se ve entonces, la gran importancia de este estándar, ya que da la posibilidad de interconectar sistemas informáticos de diferentes fabricantes y hace posible que se comuniquen entre sí, lo que en un hospital, donde los aparatos médicos son de muchas marcas diferentes, debido a la especialización, es tremendamente interesante y necesario.

La estandarización de archivos médicos hace posible que, mediante una transmisión segura, los datos de los pacientes puedan viajar de departamento en departamento, de hospital en hospital, lo que hace que esa información pueda ser vista remotamente de la zona de adquisición de las imágenes. Esto permite que los médicos puedan diagnosticar desde su casa, buscar diferentes opiniones de otros médicos expertos de una forma sencilla y rápida, un orden y estructura de los datos más efectivo y seguro.

## 5.2 Programa

La opción escogida para realizar las segmentaciones será el entorno Matlab (MATrix LABoratory) debido a sus principales características versátiles y fáciles de utilizar y a su vez robusto y potente, con el objetivo de poder desarrollar cualquier aplicación de procesado de imagen sin limitación alguna.

La característica fundamental de este entorno es la facilidad que aporta su estructura matricial de datos, conocida por sus aplicaciones en el procesado de señal e imagen. Todas las operaciones las realiza a partir de matrices (tipo base `mxArray`), es lo que convierte a Matlab en una herramienta idónea para trabajar con imágenes.

Matlab representa las imágenes como un conjunto de píxeles con información de color o intensidad de gris. Así almacena la mayoría de tipos de imágenes que soporta como matrices bidimensionales, donde cada elemento de la matriz hace referencia a un píxel de la imagen representada. La palabra *Pixel* es un anglicismo que proviene de la abreviación *picture element*, y normalmente representa un punto de luz en el monitor.

A su vez, Matlab aporta un amplio conjunto de herramientas para la creación de interfaces gráficas. La creación de código resulta sencilla, ya que no es necesario preocuparse por reserva de memoria o compatibilidad de formatos entre cada una de las variables. Estos fueron los principales motivos que nos llevaron a optar por este lenguaje de programación.

Para comenzar a desarrollar las herramientas, se dispuso de la versión 7.0 de Matlab que incorpora nuevas herramientas como la opción de tratamiento de imágenes con formato DICOM, factor clave para el diseño de los algoritmos de segmentación desarrollados.

A continuación comentaremos algunas funciones de Matlab utilizadas en el proyecto:

**Min:** Devuelve el valor mínimo de la matriz pasada como parámetro.

**Max:** Devuelve el valor máximo de la matriz de información dada como parámetro.

**Dicomread:** Permite lectura de archivos con formato Dicom, almacenando su información en una variable.

**Dicominfo:** Permite ver información de los archivos Dicom, como datos del Paciente, Historia, tipo de estudio, formato, etc.

**Dicomwrite:** Permite guardar una figura con formato Dicom y almacenar información en ella. Con ello podemos guardar los cambios realizados en las Dicom originales tras la segmentación y el posprocesado.

**Imhist:** Visualiza el histograma de la figura, con ello podremos analizar los rangos de los valores de intensidad de los distintos píxeles asociados a cada región de la imagen.

**Imadjust:** Permite expandir el histograma de valores de intensidad de los píxeles de la imagen aumentando así el contraste de las distintas regiones.

**Double:** Convierte los datos a double precisión, por defecto en Matlab los datos se crean en este tipo de precisión. Así la función `I=double(x)` devuelve datos de tipo double para los elementos de la matriz X. Si X ya es una matriz de elementos double, la función no tiene efectos sobre ella.

**Impixel:** Es una función de Toolbox de Procesado de Matlab. Determina el valor de los píxeles seleccionados con el ratón o con el teclado, y los almacena en un vector. Gracias a esta opción podemos conseguir los valores de intensidad y coordenadas del píxel que se seleccione de entrada desde el ratón.

## 5.3 Resultado de imágenes

A fin de evaluar las prestaciones del esquema de segmentación propuesto, se procesaron algunas imágenes TAC de prueba.

En primer lugar realizamos un estudio de los histogramas de las figuras. Al principio investigamos el valor de nivel de gris para poder segmentar cada parte del cuerpo. Tras este estudio pudimos comprobar que:

1) Para zonas óseas de **piernas y miembros inferiores** el valor que teníamos que utilizar es de **0.73** de valor de propiedad pues si utilizáramos otros valores no segmentaba la región objeto de interés. En la siguiente gráfica podemos verlo:

Rangos/ Casos	0.70	0.73	0.75	0.78	0.80	0.82	0.83
PierM1	Mal	Regular	Regular	Regular	ok	ok	ok
PierM25	Regular	ok	ok	ok	ok	ok	ok
PierM71	ok	ok	Regular	Regular	Mal	Mal	Mal
PierMiguellez1	Regular	ok	ok	Regular	Mal	Mal	Mal
PierMiguellez2	Regular	ok	ok	Regular	Mal	Mal	Mal
PierMiguellez3	ok	ok	ok	ok	Regular	Regular	Regular
Pier184005IM1	Regular	Regular	Regular	Regular	Mal	Mal	Mal
Pier184005IM18	Regular	ok	ok	ok	ok	Regular	Regular
Pier184005IM28	Regular	ok	Regular	Regular	Regular	Mal	Mal
Pier184005IM38	Regular	ok	ok	ok	ok	Regular	Regular
Pier184005IM48	Regular	ok	ok	ok	ok	Regular	Regular

Figura 19: Esquema de umbrales para segmentación de hueso de la pierna

Veamos el caso Pier184005IM28, cuando ponemos umbral fijado en 0.70 podemos observar que al segmentar se introduce en otra región fuera de la de interés pues ese valor de intensidad se corresponde con otras zonas de la figura. En este caso además de la zona ósea, segmenta parte de la estructura metálica donde se encuentra el paciente como muestra la siguiente figura:

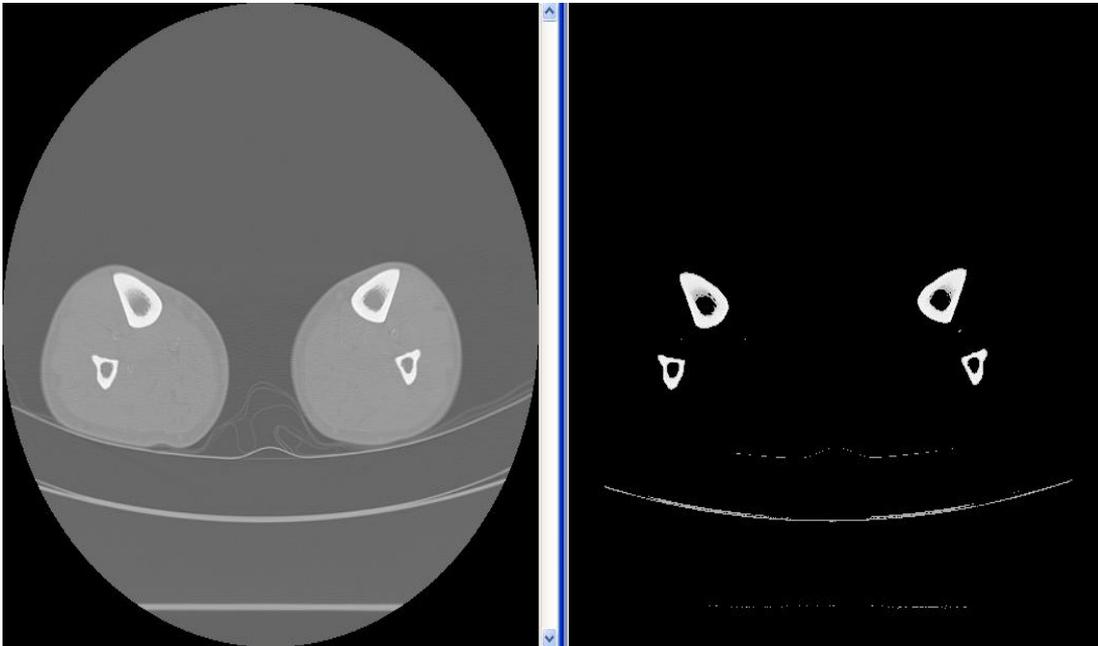


Figura 20: Segmentación de hueso de la pierna con umbral fijo 0.70

Si escogemos un umbral demasiado grande como podría ser 0.85, lo que ocurre es que al realizar la reconstrucción la región está incompleta, como podemos observar en la siguiente figura, falta hueso:

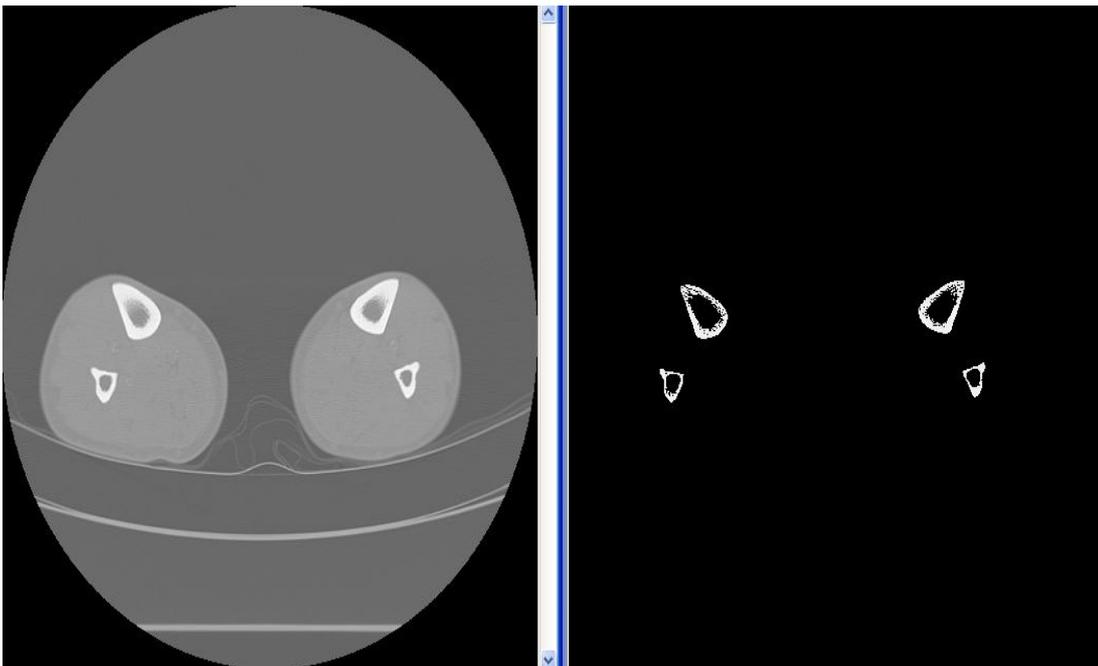


Figura 21: Segmentación de hueso de la pierna con umbral fijo 0.85

Finalmente usando el umbral escogido para segmentación de hueso correspondientes a piernas y miembros inferiores 0.73, quedaría de la siguiente forma:

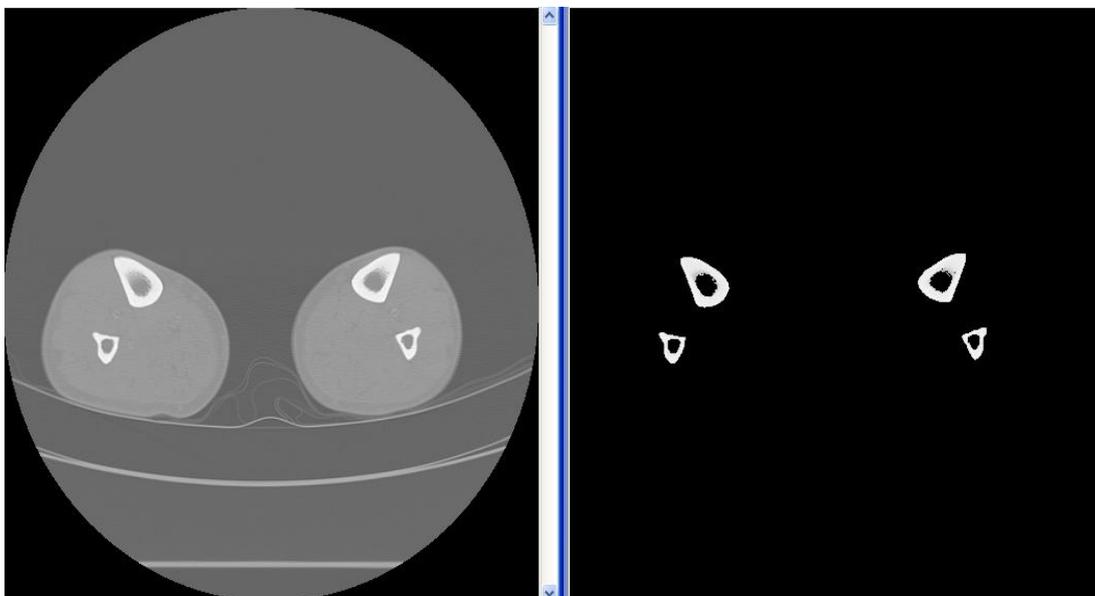


Figura 22: Segmentación de hueso de la pierna con umbral fijo 0.73

2) Para las zonas óseas correspondientes a la **cabeza**, el valor inicial que tenemos que utilizar para poder segmentar y reconstruir el modelo es de **0.75**

Rangos/Casos	0.70	0.72	0.73	0.74	0.75	0.80
CabMiguel1	Ok	ok	ok	ok	ok	Regular
CabIM1	Mal	Mal	Regular	Regular	ok	Regular
CabIM42	Mal	Mal	Mal	Regular	ok	ok
CabIM75	ok	ok	ok	ok	ok	Regular
CabIM96	ok	ok	ok	ok	ok	Regular
CabIM113	Regular	ok	ok	ok	ok	Regular
Cab184005IM1	Mal	Mal	Regular	Regular	ok	Regular
Cab184005IM25	Mal	Regular	ok	ok	ok	Regular
Cab184005IM35	Regular	Regular	ok	ok	ok	ok
Cab184005IM90	Regular	ok	ok	ok	ok	Regular
Cab184005IM104	Regular	ok	ok	ok	ok	Regular
Cab184005IM83	Regular	ok	ok	ok	ok	Regular

Figura 23: Esquema de umbrales para segmentación de hueso de cabeza

Veamos el caso Cab184005IM1 cuando utilizamos el umbral 0.70, como el umbral escogido no se corresponde con el umbral óptimo para segmentar hueso segmentará otras regiones además del hueso:

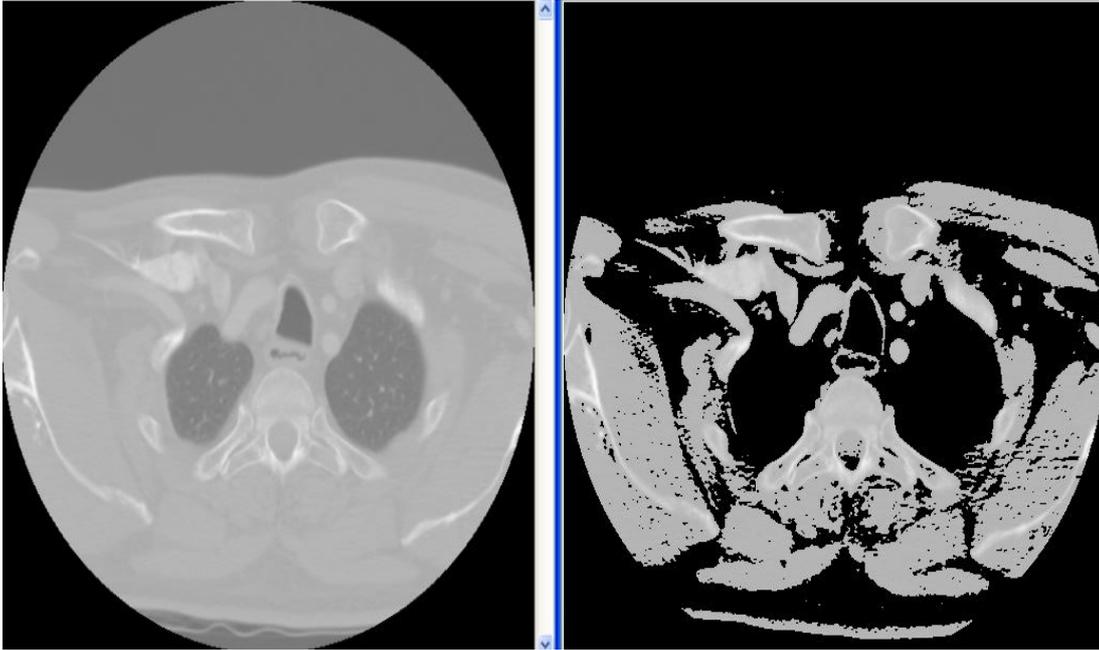


Figura 24: Segmentación de hueso de cabeza con umbral fijo 0.70

Si utilizamos el umbral 0.80 el resultado tampoco será el esperado, pues como podemos comprobar en la siguiente figura falta zona ósea:

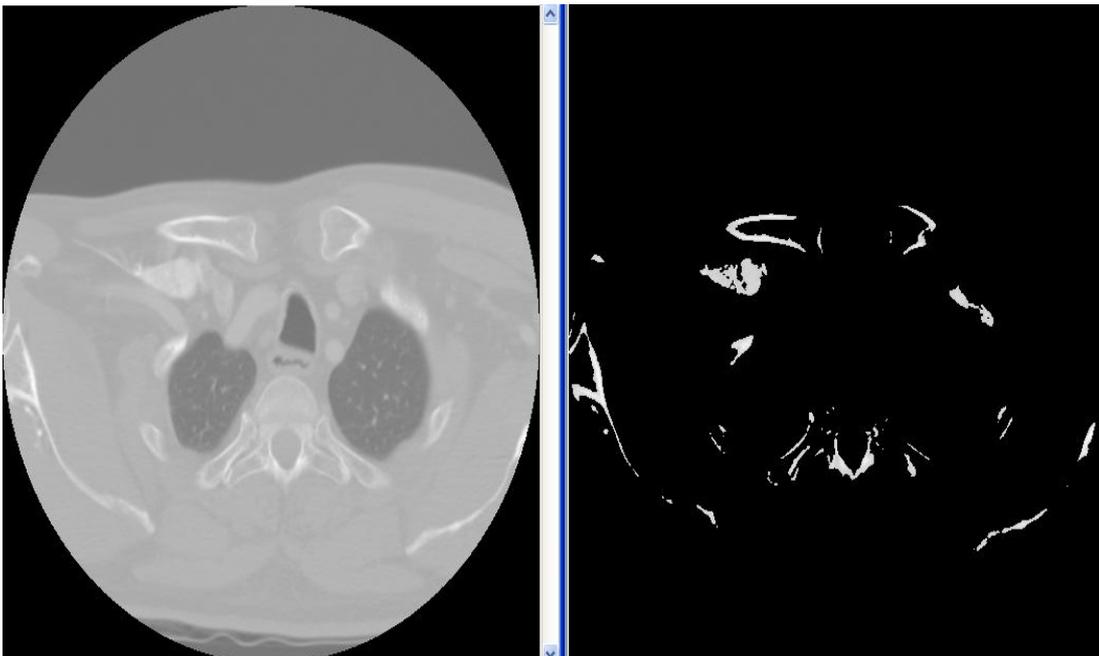


Figura 25: segmentación de hueso de cabeza con umbral fijo 0.80

Sin embargo cuando utilizamos el umbral fijado para cabeza de 0.75 el resultado es muy distinto a los anteriores. Como podemos observar en la siguiente figura se segmenta bien el hueso:

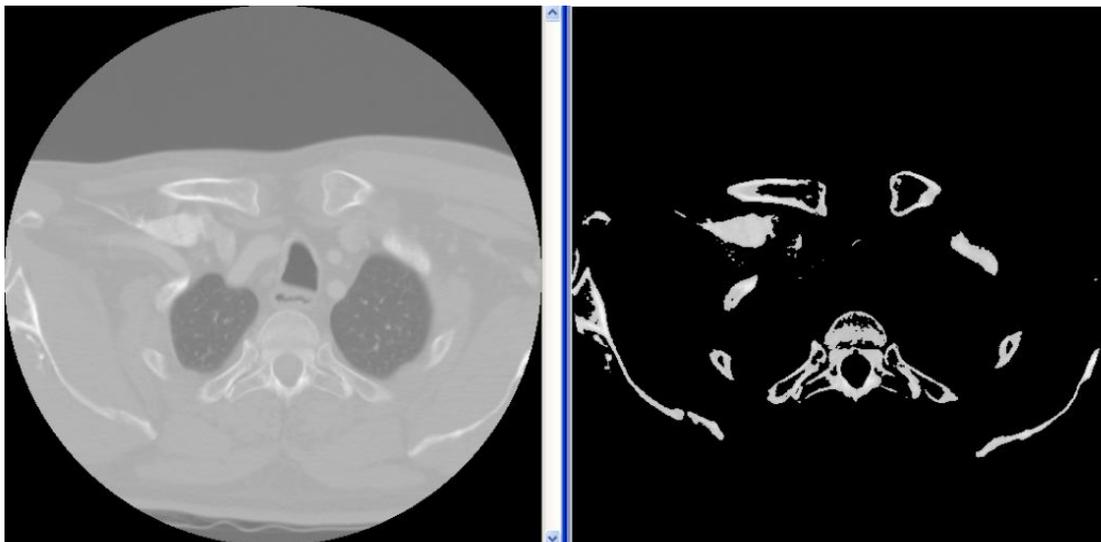


Figura 26: Segmentación de hueso de cabeza con umbral fijo 0.75

3) Para zonas óseas correspondientes a la tráquea el valor más adecuado para que el algoritmo segmente bien es de 0.67 como podemos comprobar en la siguiente tabla:

Rangos/Casos	0.63	0.65	0.67	0.70	0.73	0.75
104114987IM1	ok	ok	ok	Regular	Mal	Mal
104114987IM2		ok	ok	Regular	Mal	Mal
104114987M3	ok	ok	ok	Regular	Mal	Mal
104114987IM4	ok	ok	ok	Regular	Mal	Mal
104114987IM5	Mal	Regular	ok	Regular	Mal	Mal
104114987IM6	Mal	Regular	ok	Mal	Mal	Mal
104114987IM7	Mal	Mal	ok	ok	Mal	Mal
104114987IM8	Mal	Mal	Regular	Regular	Regular	Mal
104114987IM9	Mal	Mal	Regular	Regular	ok	Regular
104114987IM10	Mal	Mal	ok	Regular	Regular	Mal
104115284IM1	ok	ok	ok	ok	Regular	Mal
104115284IM2	Regular	ok	ok	Regular	Regular	Mal
104115284IM3	Regular	ok	ok	ok	Regular	Mal
104115284IM4	Regular	Regular	ok	ok	Regular	Regular
104115284IM5	Regular	Regular	ok	Regular	Mal	Mal
104115284IM6	Regular	Regular	ok	ok	ok	Regular

Figura 27: Esquema de umbrales para segmentación de hueso de la tráquea

Veamos el caso 104114987IM5 cuando utilizamos el umbral fijado de 0.63. Al no ser este umbral el óptimo para segmentar cabeza, podremos observar que tras realizar la reconstrucción de la imagen se segmenta otras zonas próximas al hueso respecto a niveles de propiedad en los píxeles pero que no se corresponde con la región de interés.

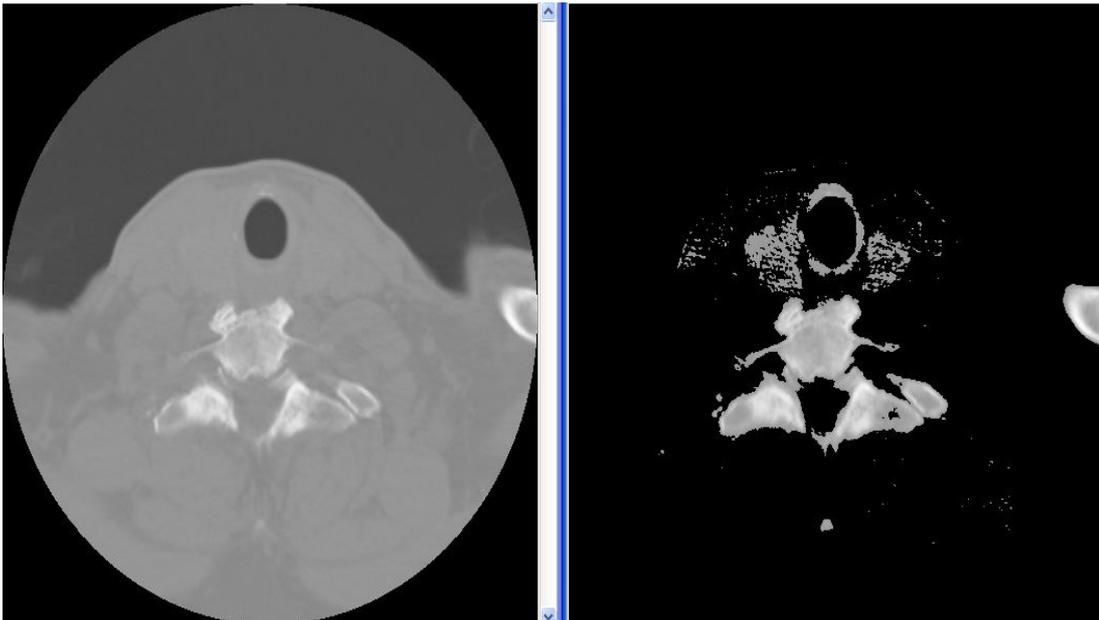


Figura 28: Segmentación de hueso de tráquea con umbral fijo 0.63

Si por el contrario utilizamos un umbral de 0.75 podemos comprobar que la región segmentada es inferior a la región de interés:

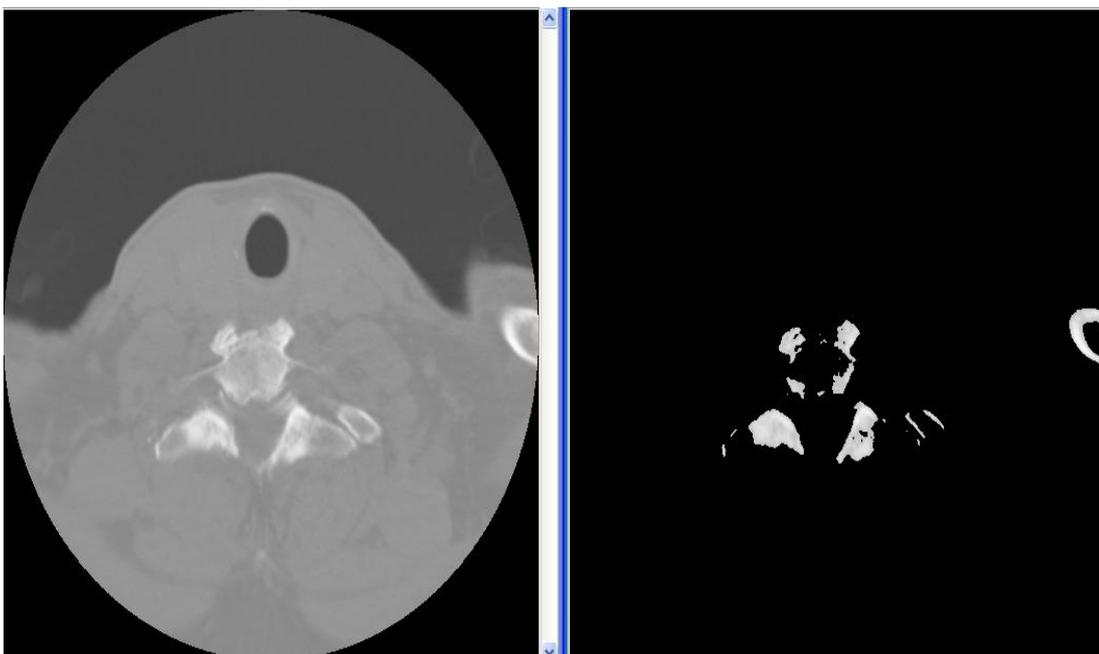


Figura 29: Segmentación de hueso de tráquea con umbral fijo 0.75

La segmentación adecuada se realizará con el umbral óptimo que tras resultados experimentales fijamos en 0.67, como se puede apreciar en la siguiente figura:

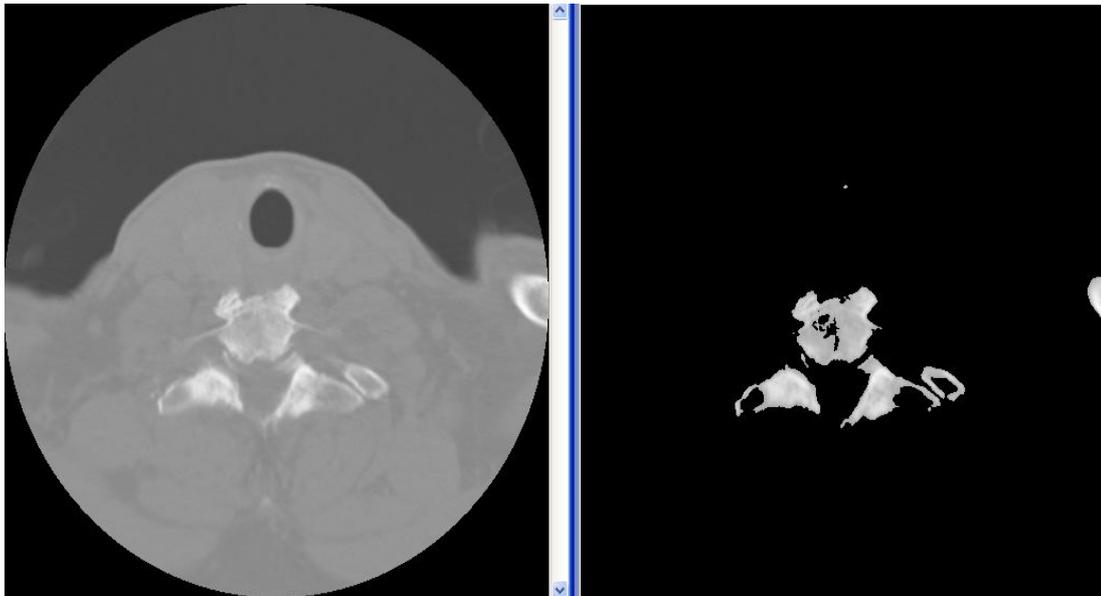


Figura 30: Segmentación de hueso de tráquea con umbral fijo 0. 67

Al hacer esto nos dimos cuenta de que teníamos que utilizar umbrales diferentes para cada zona del cuerpo y para cada tejido que quisiéramos segmentar, como nuestro objetivo es realizar una segmentación semiautomática nos interesaría utilizar otra opción que no dependiera de parámetros fijos por lo que el siguiente paso fue introducir en nuestro algoritmo el valor inicial de forma interactiva.

Con la función `impixel`, Matlab nos permite introducir interactivamente con el ratón, el valor de intensidad inicial correspondiente a la zona a segmentar independientemente de la zona del cuerpo que muestra la imagen Dicom cargada. Los tres algoritmos diseñados tendrán esta funcionalidad inicial.

### 5.3.1 Segmentación por umbralización aplicada a tejidos óseos

Comenzaremos mostrando los resultados obtenidos con el algoritmo de umbralización para tejidos óseos. El primer punto crítico sería fijar la condición de inclusión, es decir fijar el umbral:

$$|\text{Semilla}-I(x,y)| < \text{Umbral}$$

Tenemos que tener cuidado al fijar este valor pues si ponemos un valor demasiado pequeño la zona segmentada será muy pequeña y no nos reconstruirá la región de interés completa y sin embargo, si ponemos un umbral demasiado grande la zona segmentada sobrepasará la región seleccionada e inundará otras regiones. Mostraremos algunos ejemplos que nos llevarán a elegir la mejor opción.

Comenzamos poniendo un umbral de 0.10. Las imágenes cuyas zonas óseas estaban claramente definidas con valores de intensidades próximos se segmentan bien cómo podemos ver en el caso primero, lo que ocurre es que para la mayoría de los casos los niveles de intensidad de las zonas óseas no se encuentran tan uniformes, por lo que tendremos que aumentar el umbral.

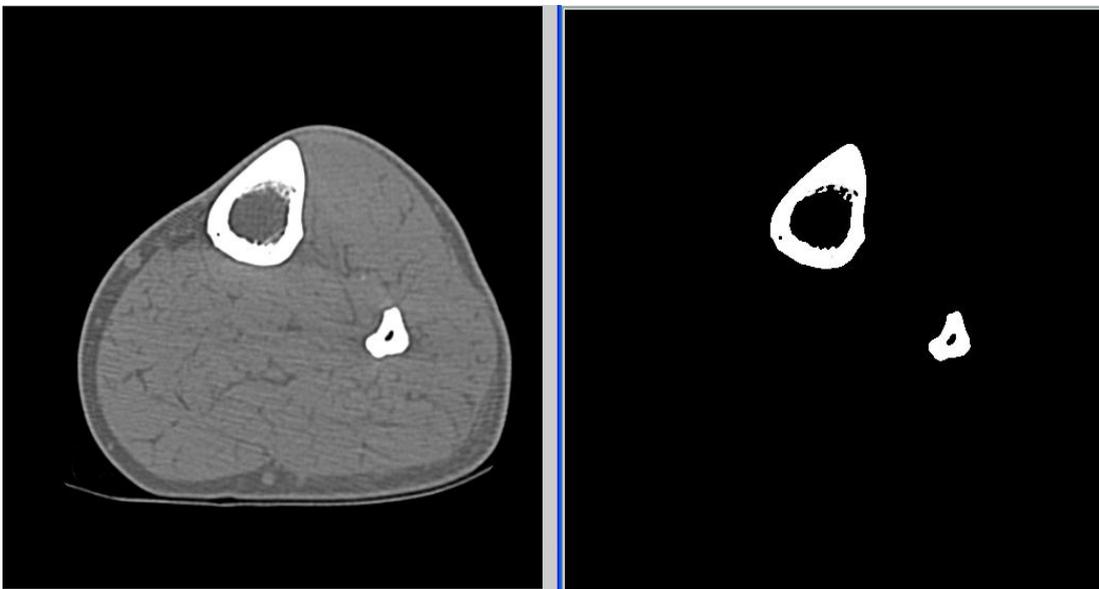


Figura 31: Caso 1. Segmentación de hueso por umbrales con rango 0.10

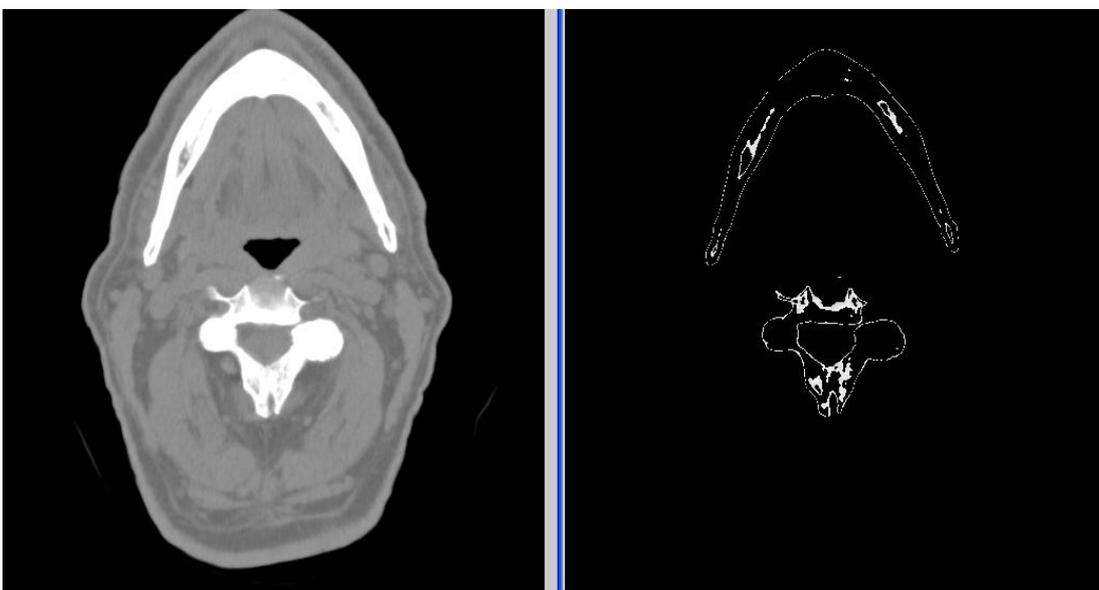


Figura 32: Caso 2. Segmentación de hueso por umbrales con rango 0.10

Para evitar esta situación, aumentaremos el rango, por lo que la siguiente opción es utilizar umbral de 0.15:

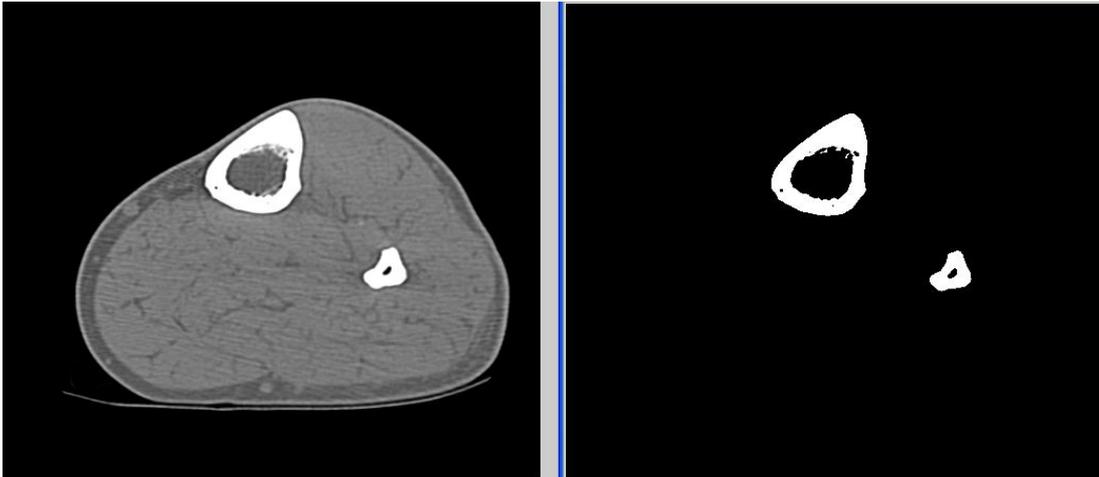


Figura 33: Caso 1. Segmentación de hueso por umbrales con rango 0.15

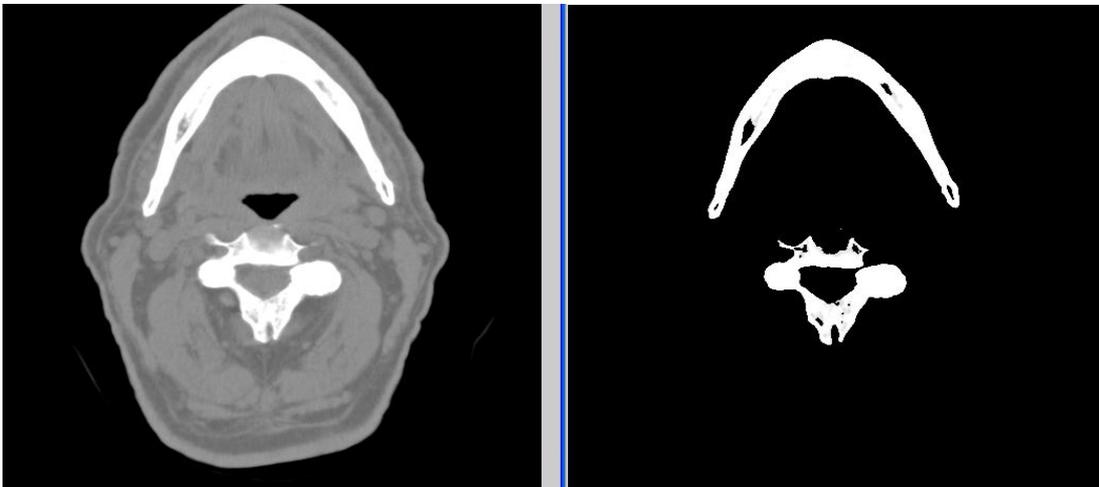


Figura 34: Caso 2. Segmentación de hueso por umbrales con rango 0.15

La imagen resultante se ve mejorada sobre todo en el caso 2, si lo comparamos con la imagen cuyo rango era 0.10, aunque todavía se muestran puntos que faltan un poco de relleno.

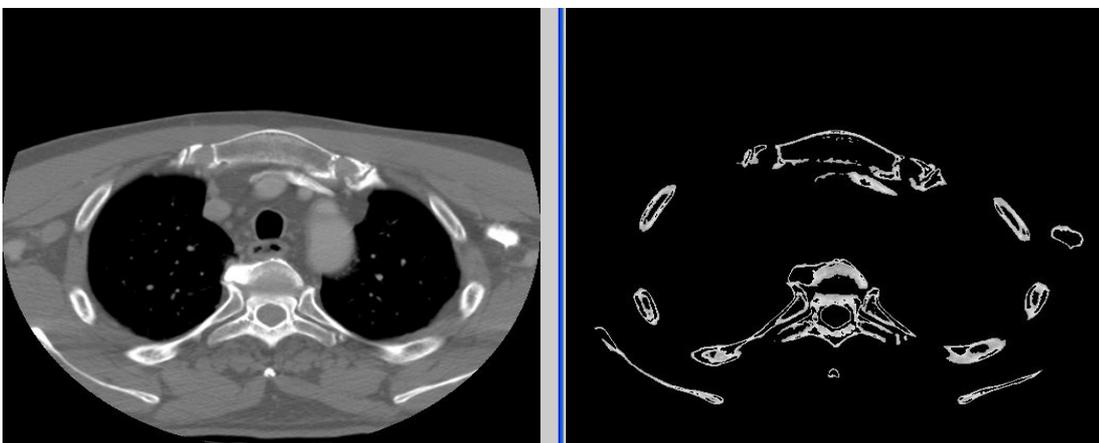


Figura 35: Caso 3. Segmentación de hueso por umbrales con rango 0.15

En este tercer caso se puede comprobar que como la zona ósea se presenta en un rango menos uniforme el umbral 0.15 es demasiado pequeño para poder reconstruir bien la zona segmentada, por lo que tendremos que aumentar el intervalo. El siguiente paso será comprobar si es válido el umbral de 0.20:

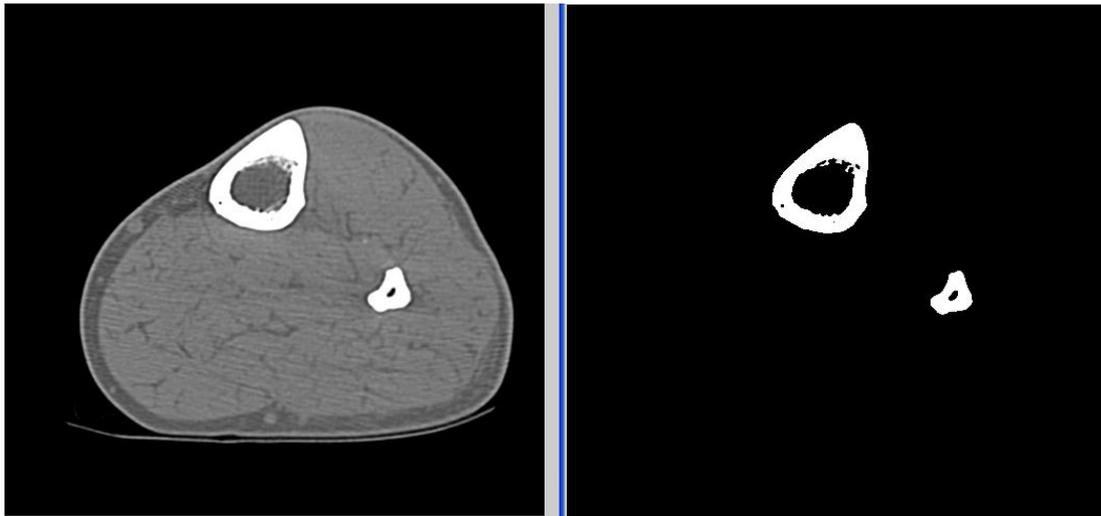


Figura 36: Caso 1. Segmentación de hueso por umbrales con rango 0.20

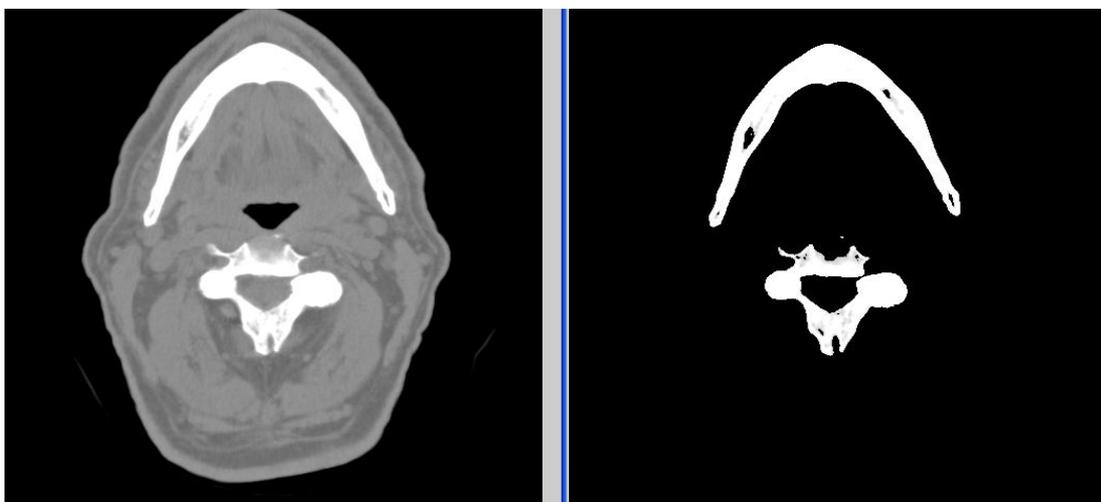


Figura 37: Caso 2. Segmentación de hueso por umbrales con rango 0.20

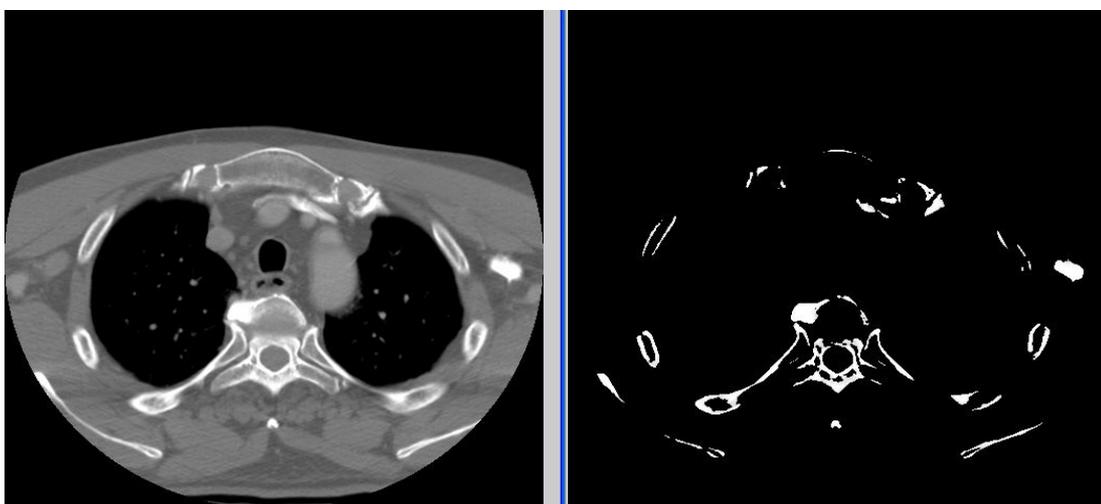


Figura 38: Caso 3. Segmentación de hueso por umbrales con rango 0.20

En los tres casos se puede apreciar una mejora con este aumento del intervalo de nivel de gris aceptado, sobre todo en el tercer caso pues con el umbral 0.15 no se reconstruía bien el hueso.

Tras la vista de los resultados podemos sospechar que el umbral que vamos a fijar para realizar el algoritmo de segmentación por umbralización para huesos está próximo al valor umbral 0.20.

Para solventar los pequeños huecos existentes en el caso tres imponemos que el umbral sea 0.22 y veamos los resultados para los tres casos:

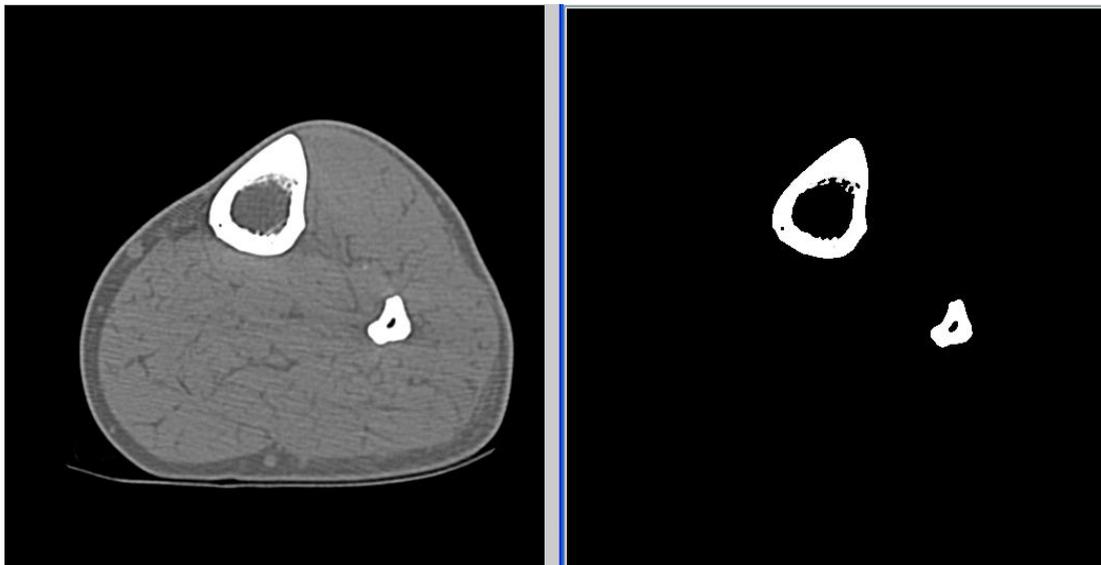


Figura 39: Caso 1. Segmentación de hueso por umbrales con rango 0.22

El resultado obtenido en el primer caso es excelente. Comprobemos que ocurre en el resto de casos:

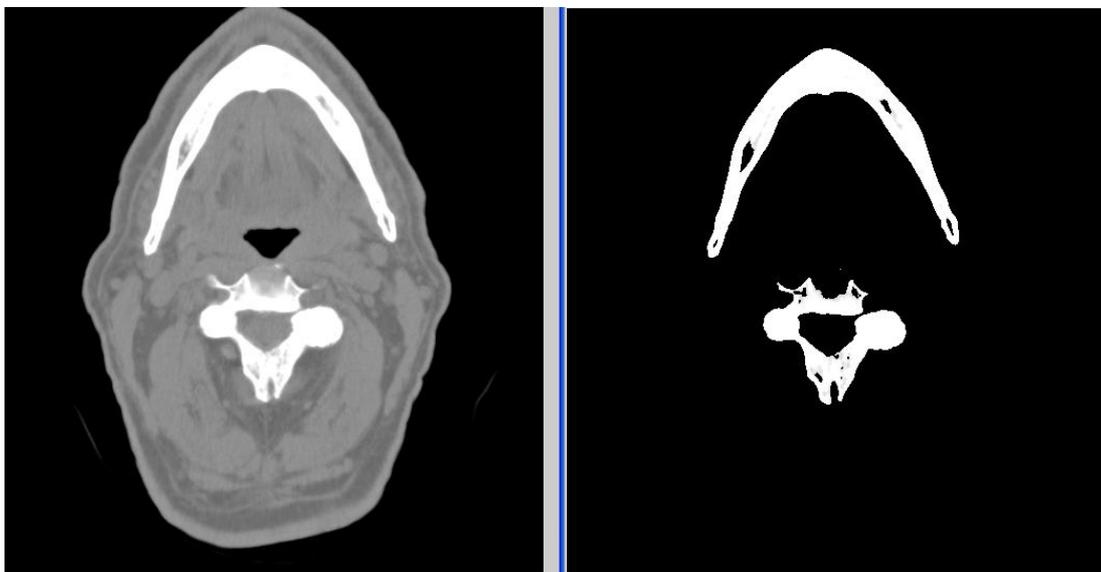


Figura 40: Caso 2. Segmentación de hueso por umbrales con rango 0.22

La reconstrucción del segundo caso se aproxima más a la realidad para el umbral de 0.22 como en el caso uno que vimos anteriormente. Veamos que ocurre en el caso tercero:

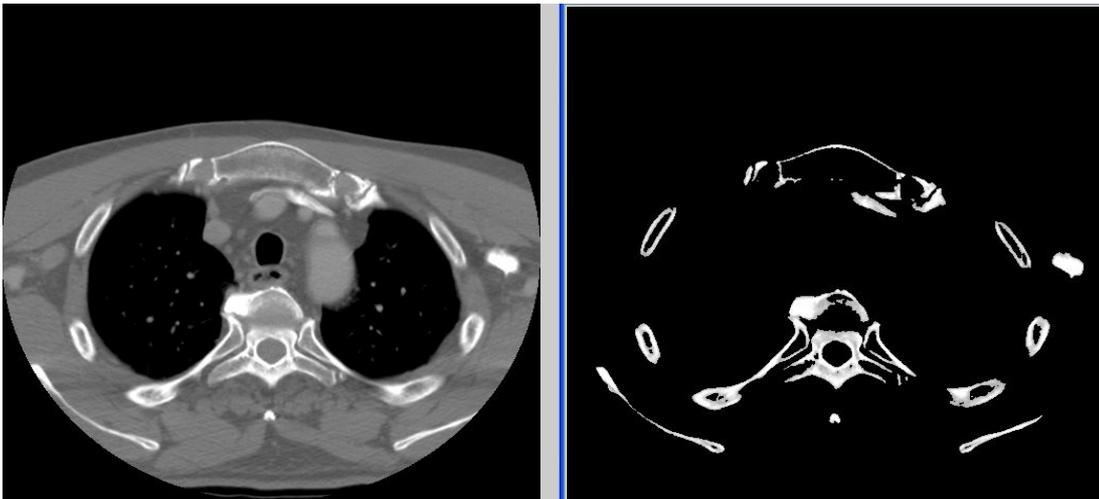


Figura 41: Caso 3. Segmentación de hueso por umbrales con rango 0.22

En este tercer caso se puede apreciar notablemente la mejoría, pues se rellenan zonas que quedaban vacías con los umbrales anteriores. El umbral 0.22 podría ser un candidato idóneo para la segmentación por umbrales de zonas óseas.

Al probar con umbrales mayores de 0.22 nos damos cuenta de que el rango comprendido entre el valor de intensidad del píxel introducido por el usuario ( $I_0$ ) y los valores de intensidad de los píxeles que cumplen la condición de inclusión ( $I(x,y)$  incluidos) es excesivo.

$$|I_0 - I(x,y)| < \text{Umbral}$$

Esto lo podemos demostrar al visualizar las imágenes resultantes de la segmentación, donde se puede visualizar que se reconstruye zonas que no pertenecen a la región ósea.

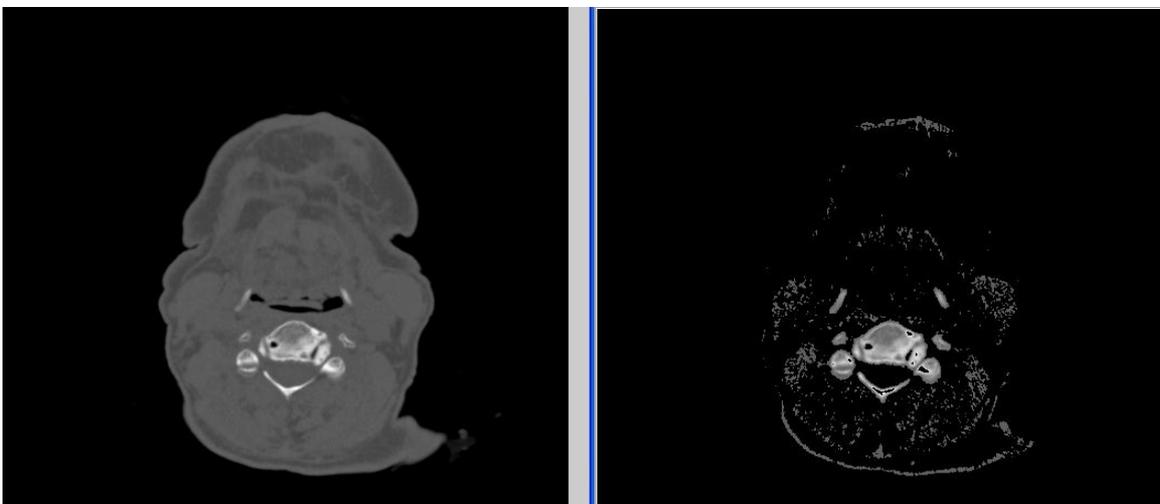


Figura 42: Segmentación de hueso por umbrales con rango excedido

Tras comprobar el umbral de 0.22 sobre 230 imágenes Dicom provenientes de tomografía y de resonancia magnética, obtenemos los siguientes resultados:

- 1 segmenta mal
- 2 segmenta regular
- 227 segmenta bien

A la vista de los resultados decidimos optar por **0.22** como **umbral óptimo** para realizar la segmentación por umbralización de zonas óseas.

Finalmente si realizamos técnicas de pos-procesado para suavizar las regiones segmentadas se eliminarán partes de las zonas óseas como podemos comprobar en las siguientes imágenes, por lo que no realizaremos este último tratamiento.



**Figura 43: Resultado tras pos-procesado de filtro de mediana en segmentación de hueso**

### 5.3.2 Segmentación por umbralización para tejidos exceptuando el tejido óseo

A continuación veremos los resultados obtenidos tras la aplicación del **segundo algoritmo de segmentación** diseñado por umbrales para distintos tejidos. El umbral de la condición de inclusión es fijado por resultados experimentales como hicimos en el algoritmo anterior.

Tras realizar pruebas vemos que a partir de umbral 0.02 empieza a mostrarse los distintos tejidos. Veamos el caso primero de esta segmentación:

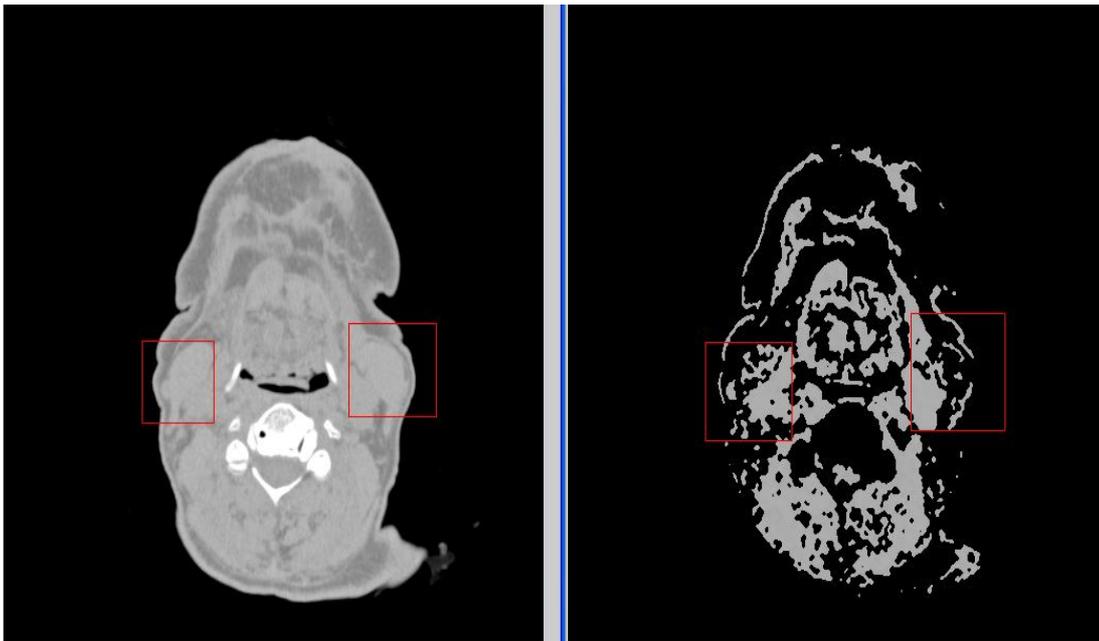


Figura 44: Caso 1. Segmentación de tejidos por umbralización con rango 0.02

Como podemos observar en las imágenes existen zonas que le faltan tejido, esto significa que el umbral escogido para realizar la segmentación por umbralización es pequeño para esta imagen. El siguiente paso será aumentar el umbral a 0.04. Veamos el resultado en el primer caso, aumentando el umbral:

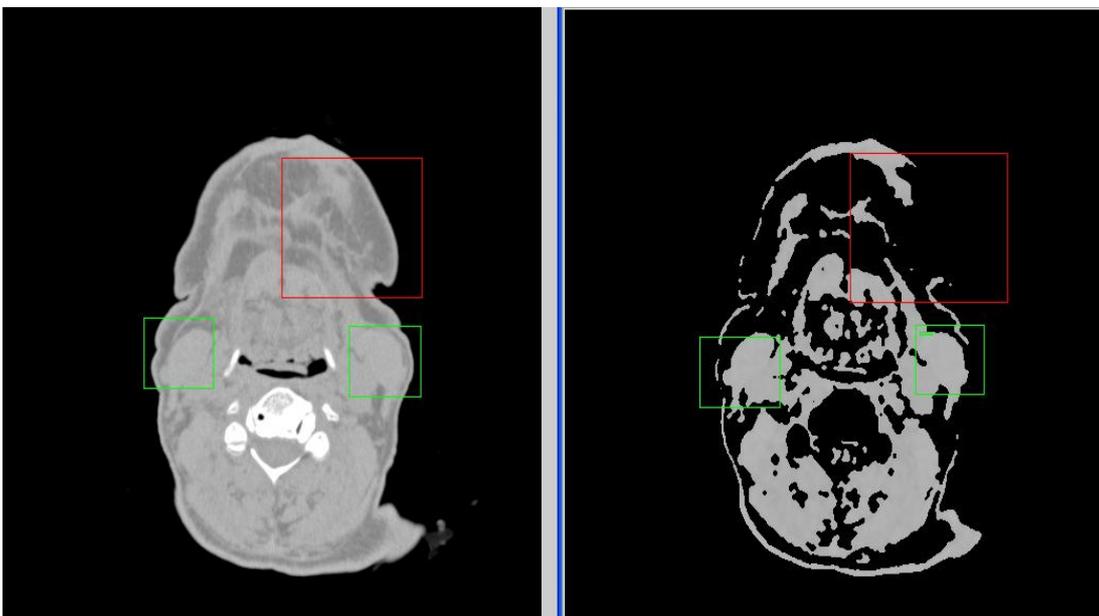


Figura 45: Caso 1. Segmentación de tejidos por umbralización con rango 0.04

Al visualizar la imagen podemos ver que mejora respecto a la segmentación que utiliza el rango de 0.02 (cuadros verdes), sin embargo en el tejido segmentado siguen apareciendo huecos (cuadro rojo), por lo que la imagen no se reconstruye bien. Veamos el resultado en un segundo caso con rango 0.04:

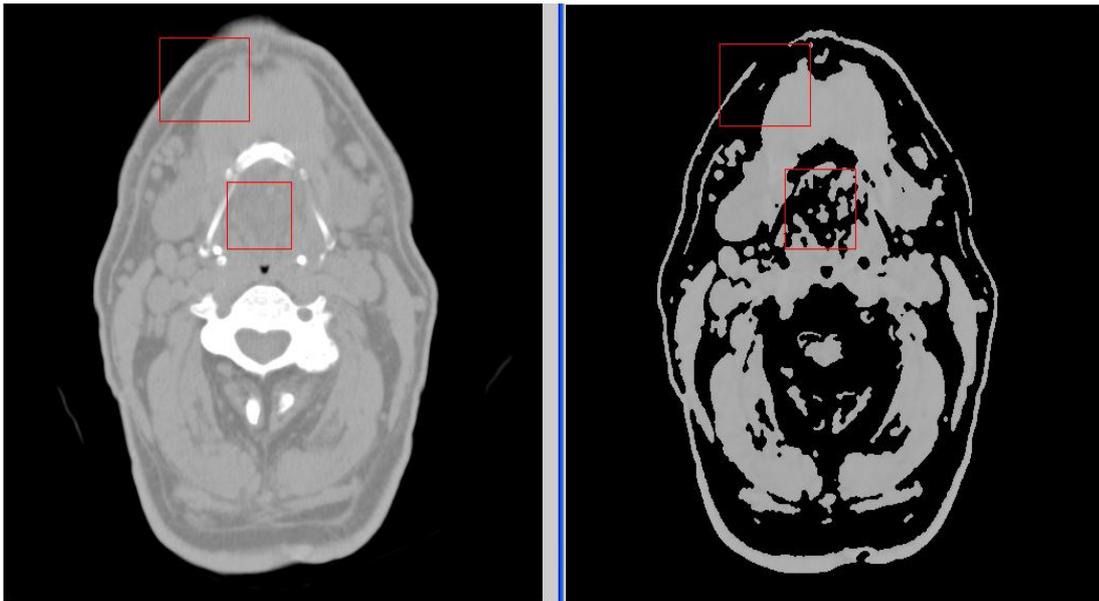


Figura 46: Caso 2. Segmentación de tejidos por umbralización con rango 0.04

El resultado en este segundo caso es bastante sorprendente, pues la zona segmentada cubre casi todo el tejido elegido por el usuario, solamente aparece algunos huecos en las regiones enmarcadas en rojo. Analizaremos qué ocurre cuando aumentamos el rango a 0.06:

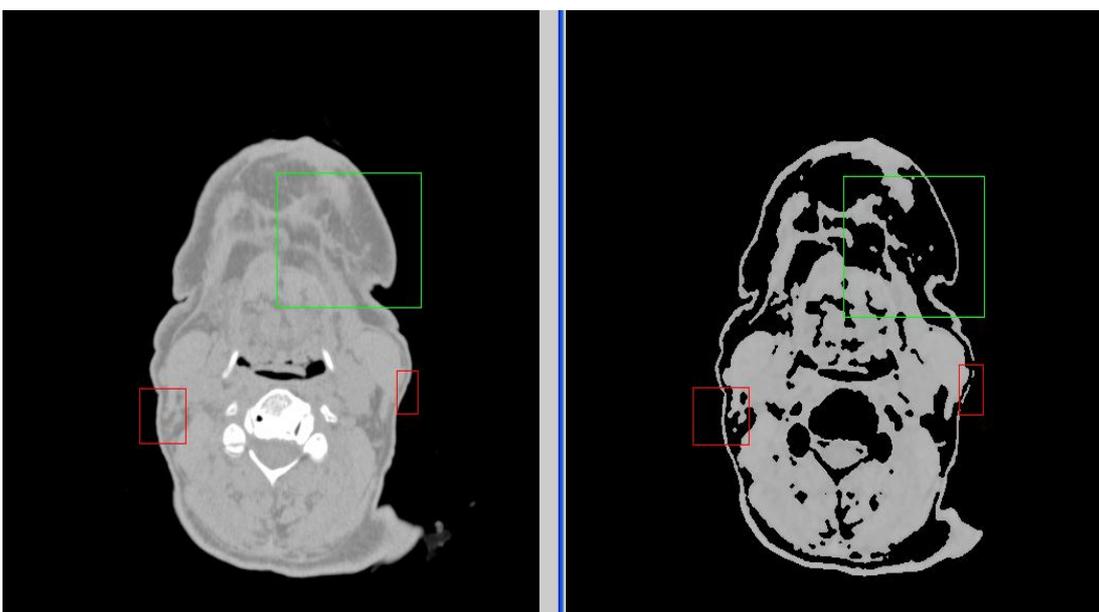


Figura 47: Caso 1. Segmentación de tejido por umbralización con rango 0.06

Se aprecia una notable mejora si aplicamos como umbral 0.04 para el caso primero como podemos observar en las zonas enmarcadas de verde y unos pequeños huecos apenas apreciables en los recuadros rojos que con un pequeño incremento del rango conseguiríamos la reconstrucción exacta del tejido. Veamos el segundo caso:

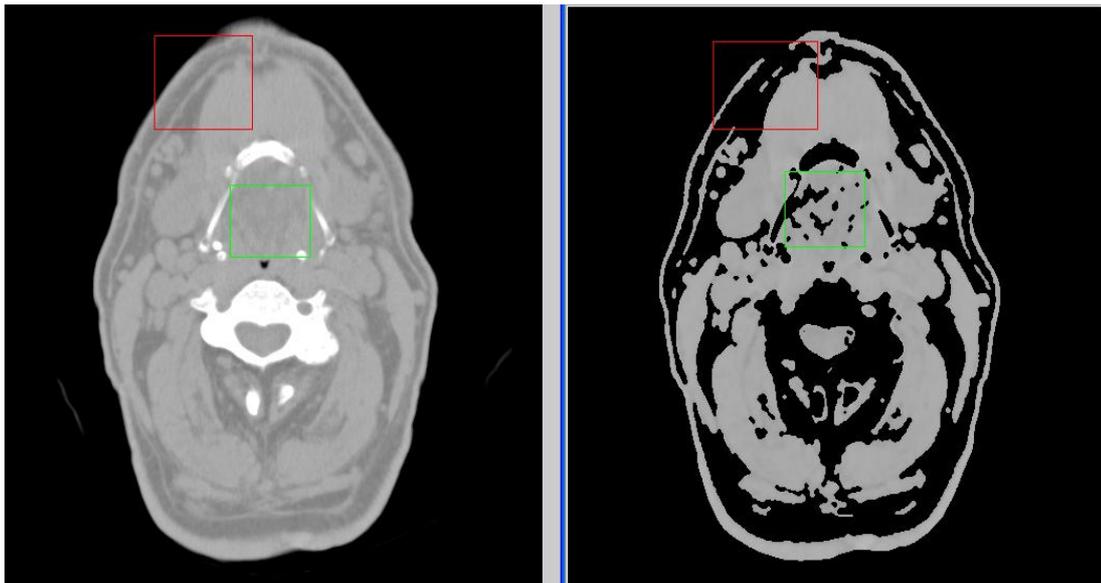


Figura 48: Caso 2. Segmentación de tejido por umbralización con rango 0.06

En las zonas enmarcadas de rojo observamos que sigue faltando contorno en el tejido. Esto se debe a que el rango establecido de intensidades comprendidas entre el valor del píxel introducido por el usuario y el resto de píxeles que cumplen la condición de inclusión debe ser un poco mayor, pero en torno a 0.06, pues el resto de tejido sí se reconstruye bien. Veamos el tercer caso:

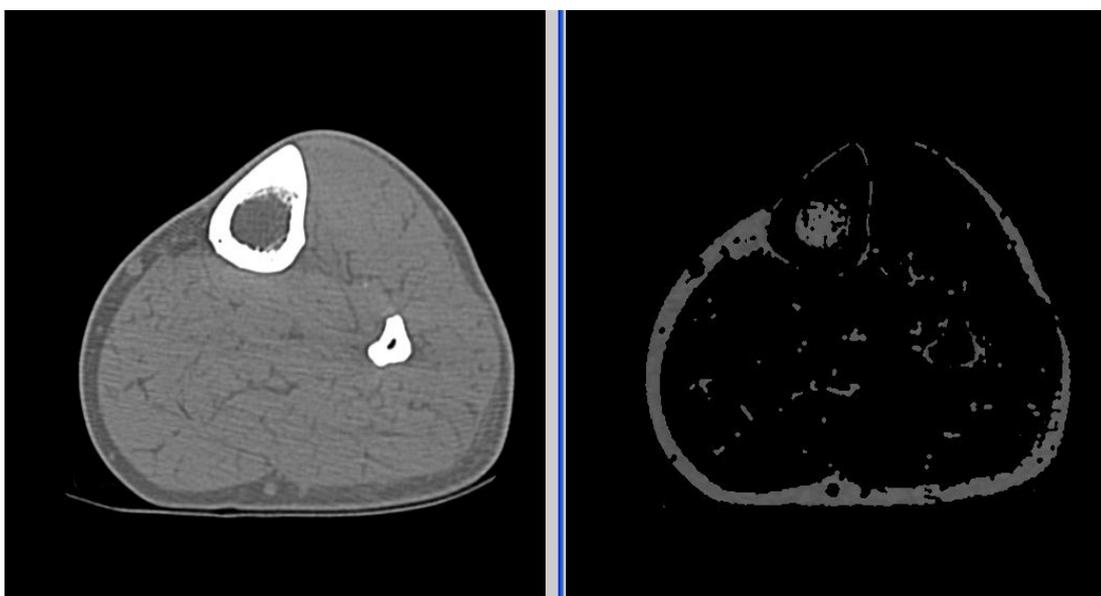


Figura 49: Caso 3. Segmentación de tejido por umbralización con rango 0.06

En los tres casos a los que hemos aplicado el rango de intensidades de 0.06 los resultados son muy parecidos. Para imágenes en la que existe un buen contraste original, podríamos aplicarle un rango de variación de intensidad un poco menos restrictivo. Sin embargo, en la mayoría de los casos las distintas regiones que componen la imagen se muestran como en los casos anteriores, por lo que una posible opción sería utilizar rango 0.06.

Comprobemos qué ocurre cuando aplicamos un rango en torno al 0.06 pero con un ligero incremento para mejorar el primer caso, tomemos el valor 0.065 y lo aplicaremos a los tres casos:

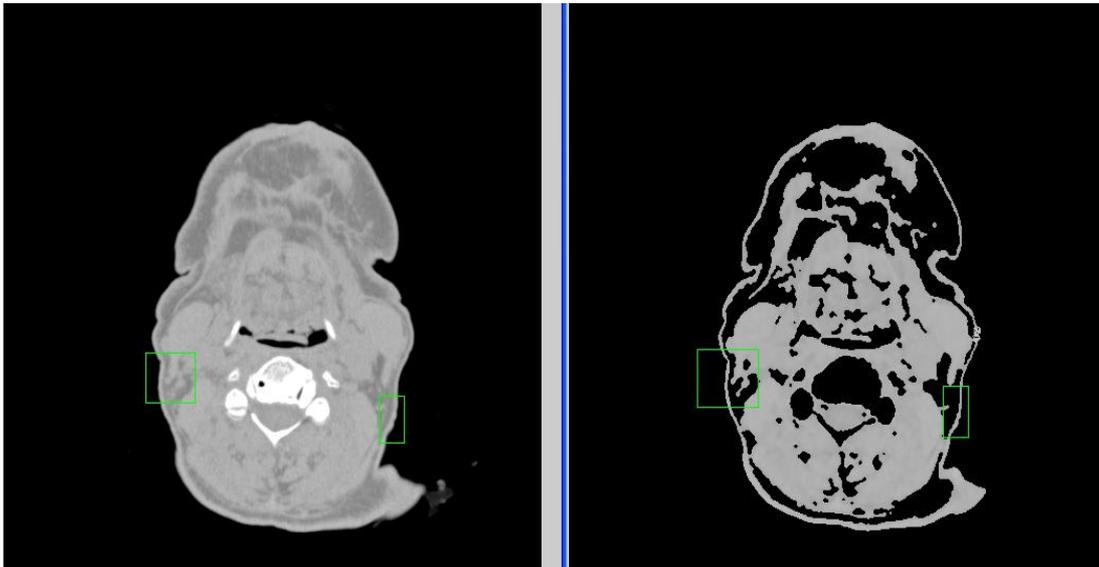


Figura 50: Caso 1. Segmentación de tejido por umbralización con rango 0.065

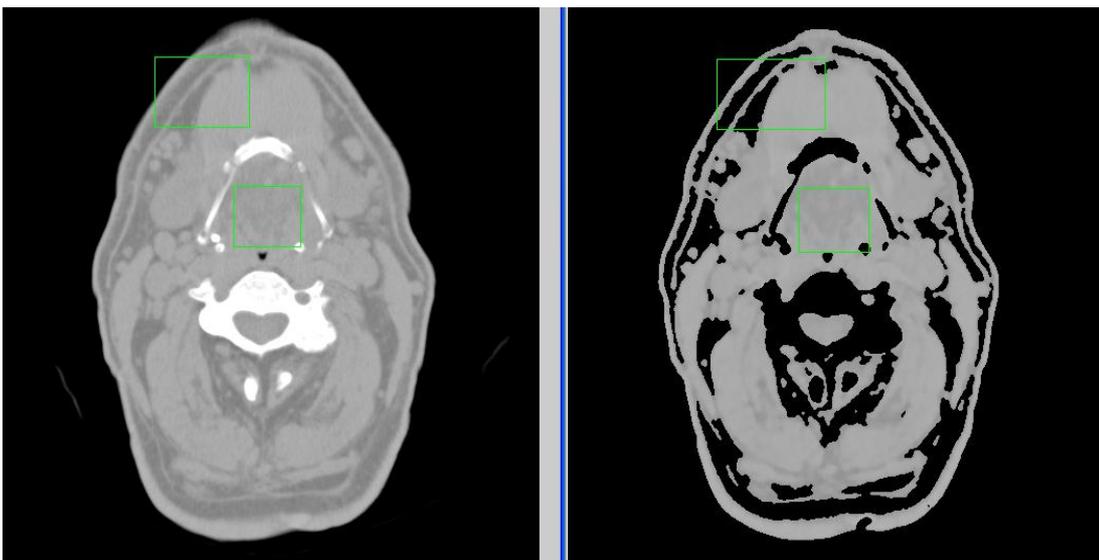


Figura 51: Caso 2. Segmentación de tejido por umbralización con rango 0.065

Como podemos observar en la segmentación con el rango 0.065 de valores de grises permitidos se mejora la reconstrucción del tejido seleccionado, sobre todo en las zonas enmarcadas de verde.

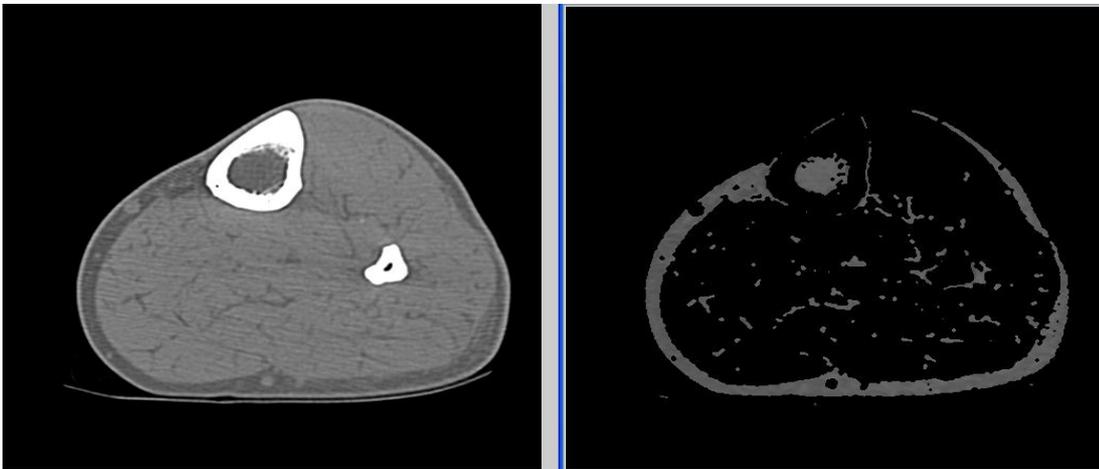


Figura 52: Caso 3. Segmentación de tejido por umbralización con rango 0.065

Veamos qué ocurre cuando aumentamos un poco más el rango de intensidades permitidas:

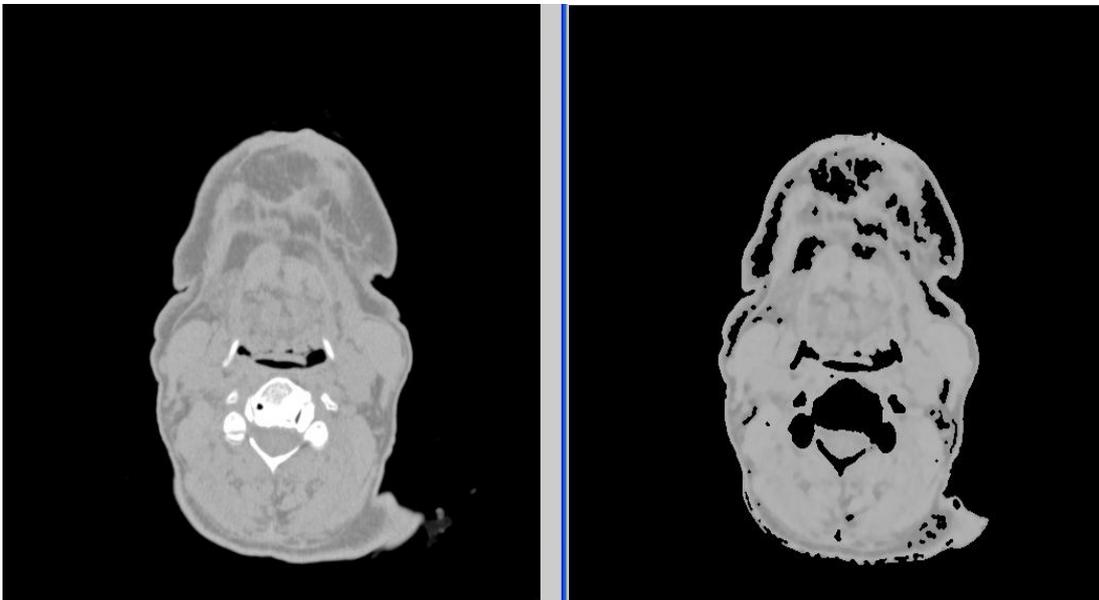


Figura 53: Caso 1. Segmentación de tejido por umbralización con rango 0.07

Como podemos observar el rango de intensidades permitidas es excesivo y hace que se introduzcan píxeles pertenecientes a regiones vecinas correspondientes a otros tejidos. La solución es tomar un rango inferior de intensidades válidas.

La opción escogida es utilizar un umbral de 0.065 para la condición de inclusión del algoritmo de segmentación por umbralización para tejidos exceptuando zonas óseas:

$$|I_0 - I(x,y)| < 0.065$$

De los 260 casos analizados con este algoritmo de segmentación obtenemos los siguientes resultados:

- 2 segmenta mal

- 3 segmenta regular
- 255 segmenta bien

En todos estos resultados y en las imágenes mostradas, tras la segmentación han sido necesarios aplicar 4 filtros de mediana para suavizar el contorno y eliminar puntos sueltos que no corresponden a la región segmentada. Mostraremos a continuación el efecto de aplicar filtro de mediana

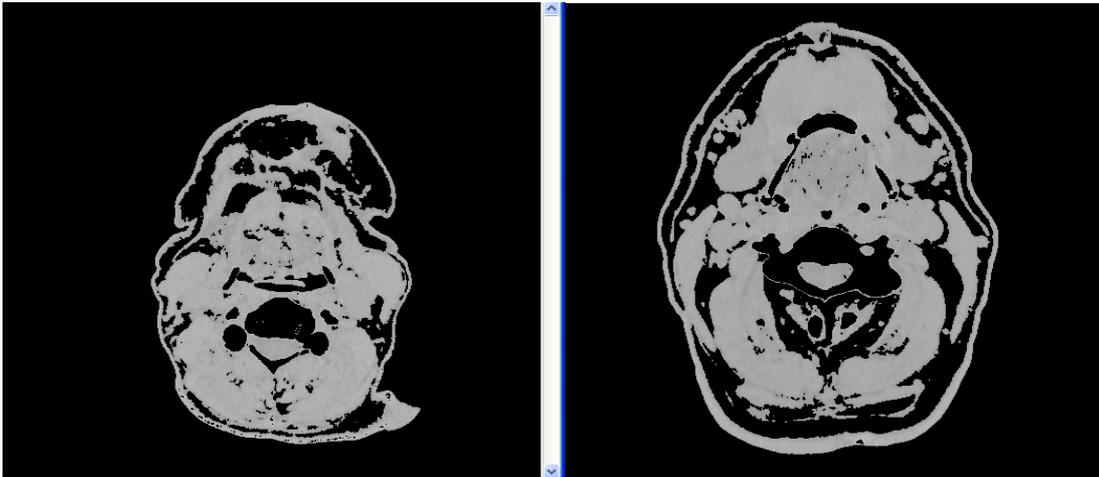


Figura 54: Segmentación de tejido por umbralización sin filtrado

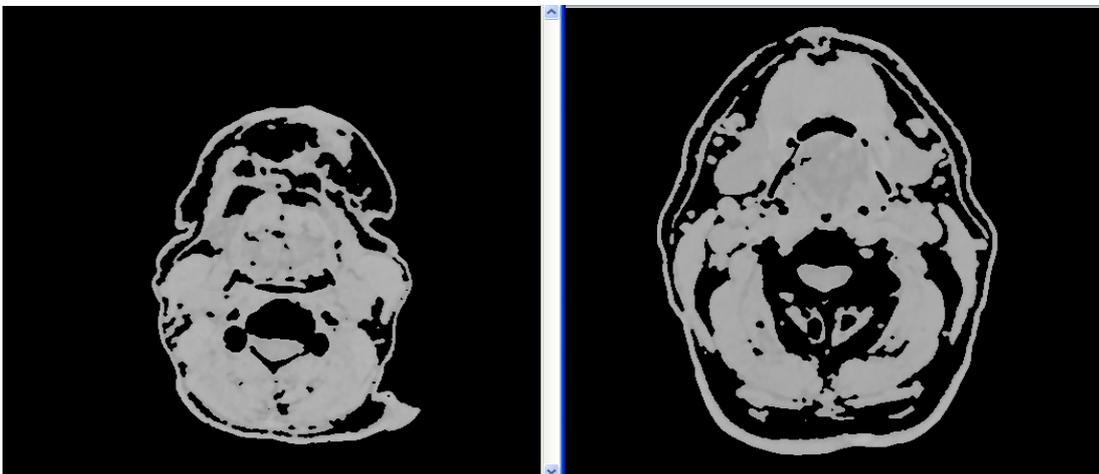


Figura 55: Segmentación de tejido por umbralización tras aplicación filtros de mediana.

### 5.3.3 Segmentación de crecimiento de regiones aplicada a tejidos exceptuando el tejido óseo

A continuación implementaremos el **algoritmo de segmentación por crecimiento de regiones**. El umbral correspondiente se fijará tras resultados experimentales como podremos ver a continuación. Para ello el primer umbral que proponemos es:

$$|\text{Sem}-I(x,y)| \leq 0.005$$

Este umbral lo aplicaremos en los mismos casos que en la segmentación por umbralización de tejidos para poder contemplar las posibles diferencias. Veamos qué ocurre en el caso primero:

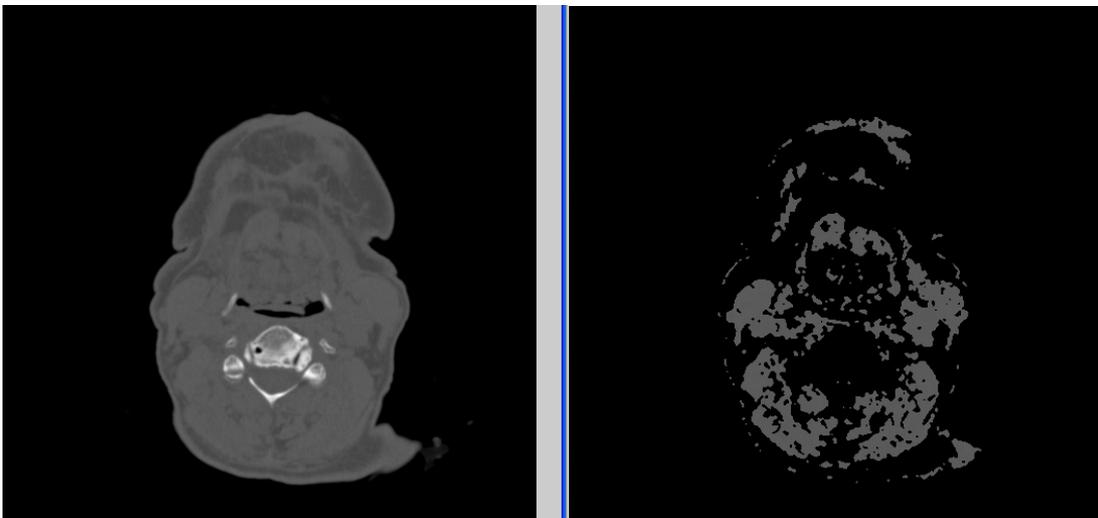


Figura 56: Caso 1. Segmentación por crecimiento de semillas con umbral 0.005

Con el umbral fijado en 0.005 el tejido seleccionado por el usuario no se reconstruye del todo. Tendremos que subir el umbral, veremos que ocurre cuando le damos el valor 0.01

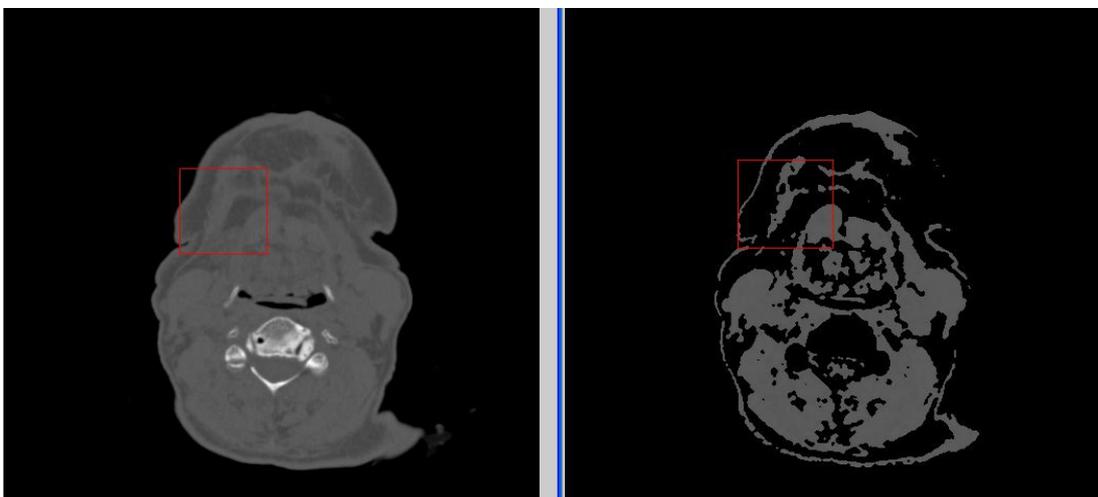
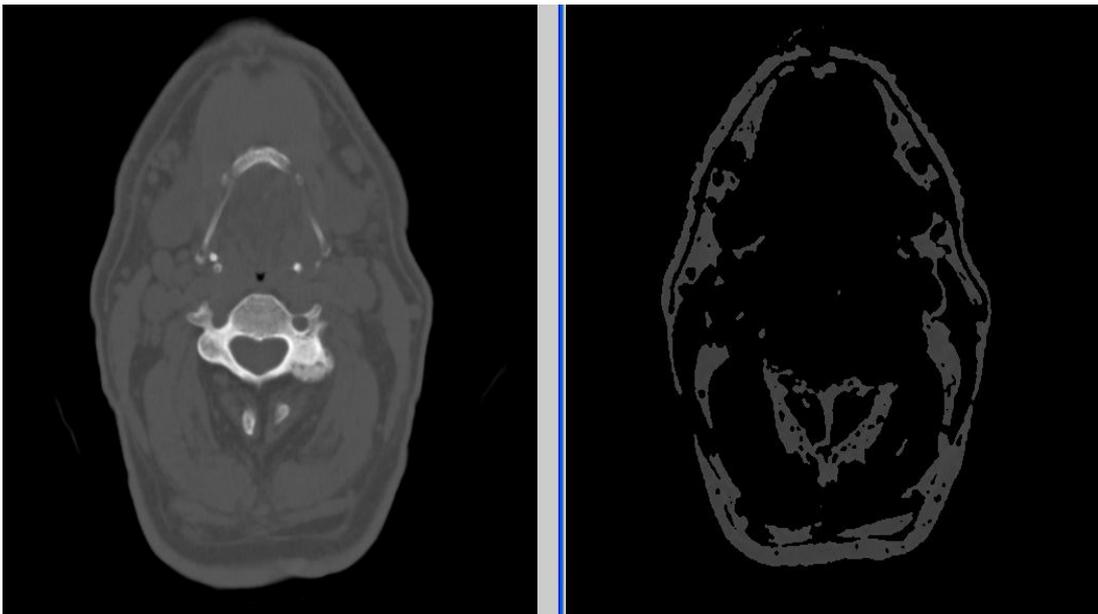


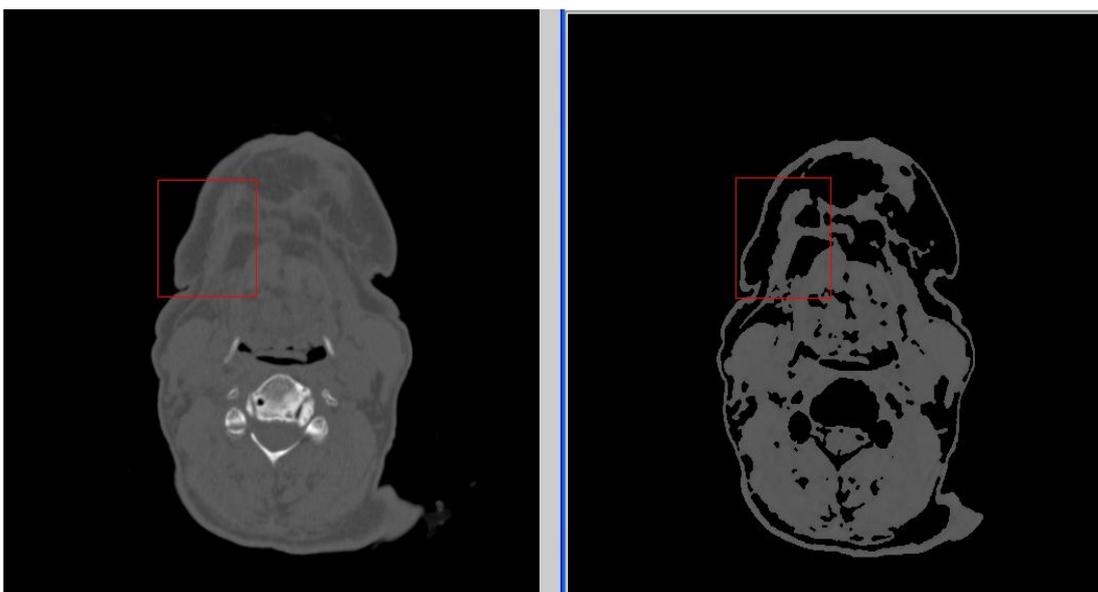
Figura 57: Caso 1. Segmentación por crecimiento de semillas con umbral 0.01

Observamos en la figura anterior, que el resultado de la segmentación con umbral en 0.01 se ve mejorado respecto al caso que utiliza de umbral 0.005. Veamos los resultados en el siguiente caso:



**Figura 58: Caso 2. Segmentación por crecimiento de región con umbral 0.01**

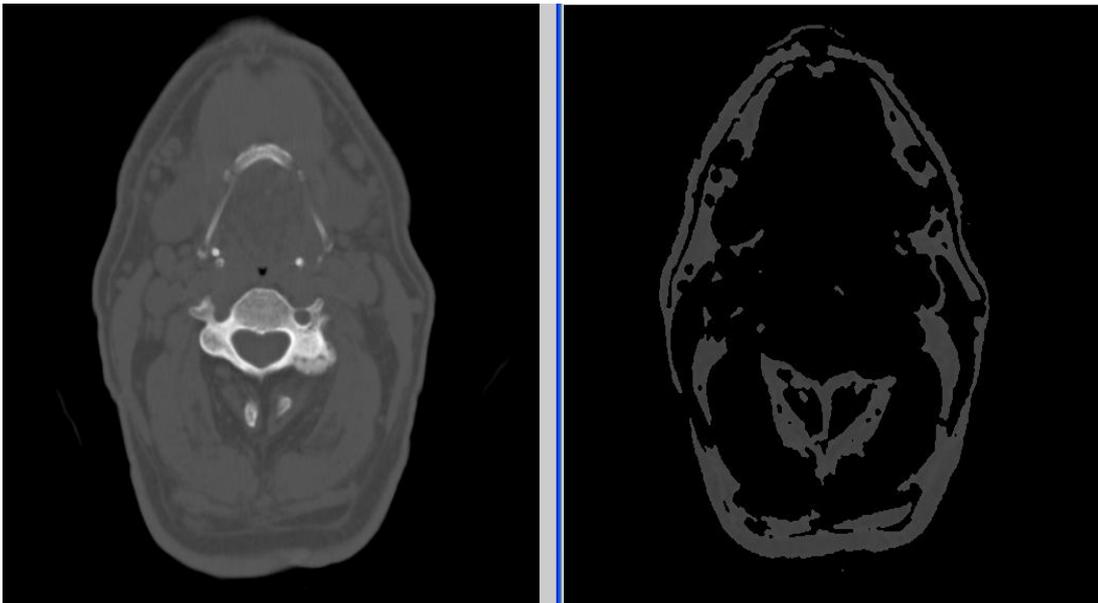
Sin embargo, los contornos de la imagen aún no están claramente definidos y las zonas segmentadas aparecen incompletas, por lo que utilizaremos un umbral mayor, tomemos el umbral 0.02.



**Figura 59: Caso 1. Segmentación por crecimiento de región con umbral 0.02**

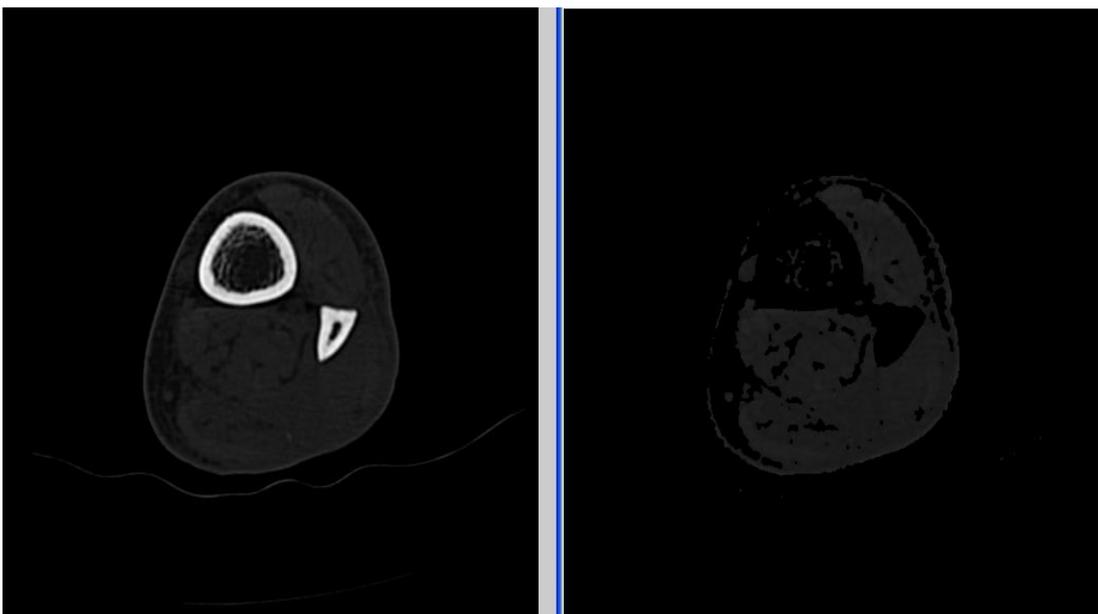
El resultado de la segmentación con umbral en 0.02 es muy bueno, como podemos ver la zona recuadrada de rojo se mejora la

reconstrucción del tejido y aparecen bordes del tejido que no existían cuando utilizábamos el umbral de 0.01. Veamos en el segundo obtenemos los mismos resultados



**Figura 60: Caso 2. Segmentación por crecimiento de región con umbral 0.02**

Los resultados son evidentes, la región segmentada y posteriormente reconstruida automáticamente se corresponde con el tejido que el usuario selecciona interactivamente. Veamos los resultados de un tercer caso:



**Figura 61: Caso 3. Segmentación por crecimiento de región con umbral 0.02**

El resultado obtenido en los tres casos tras la aplicación del algoritmo de segmentación de crecimiento de regiones por semilla con umbral 0.02 es bastante satisfactorio como podemos observar en las imágenes. Comprobemos lo que ocurre cuando el rango de

intensidades válidas impuestas en la condición de inclusión aumenta. Para ello tomamos como umbral 0.04 y lo aplicaremos a los tres casos anteriores.

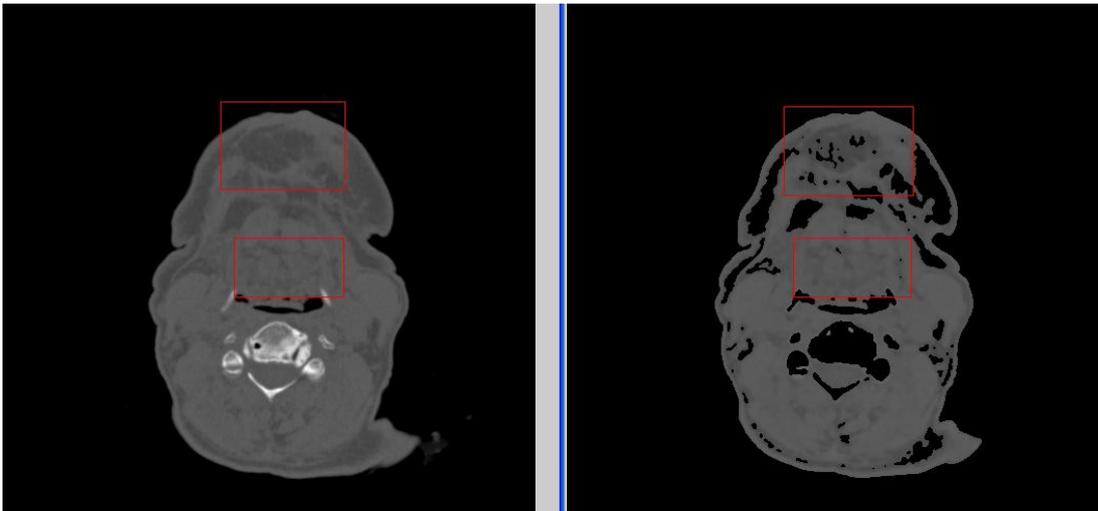


Figura 62: Caso 1. Segmentación por crecimiento de región con umbral 0.04

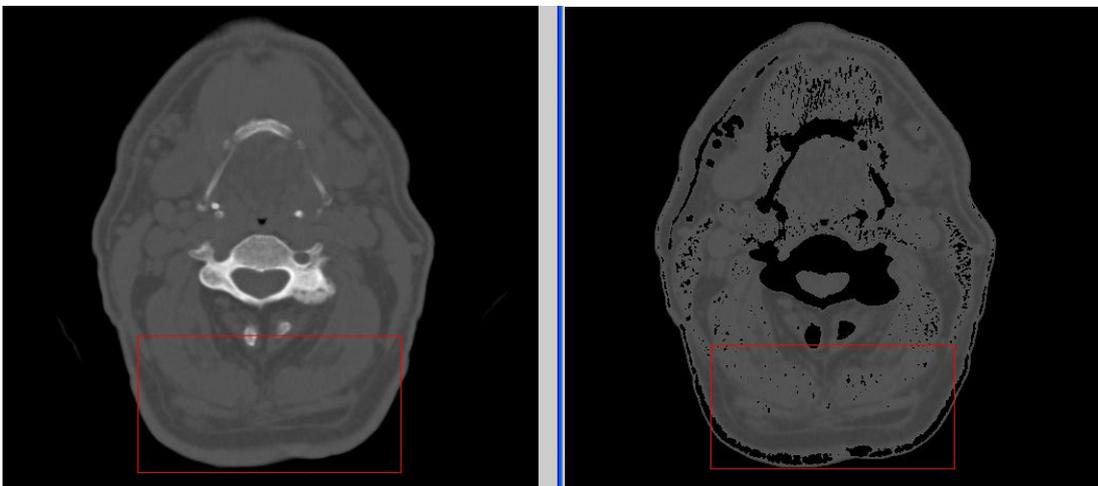


Figura 63: Caso 2. Segmentación por crecimiento de región con umbral 0.04

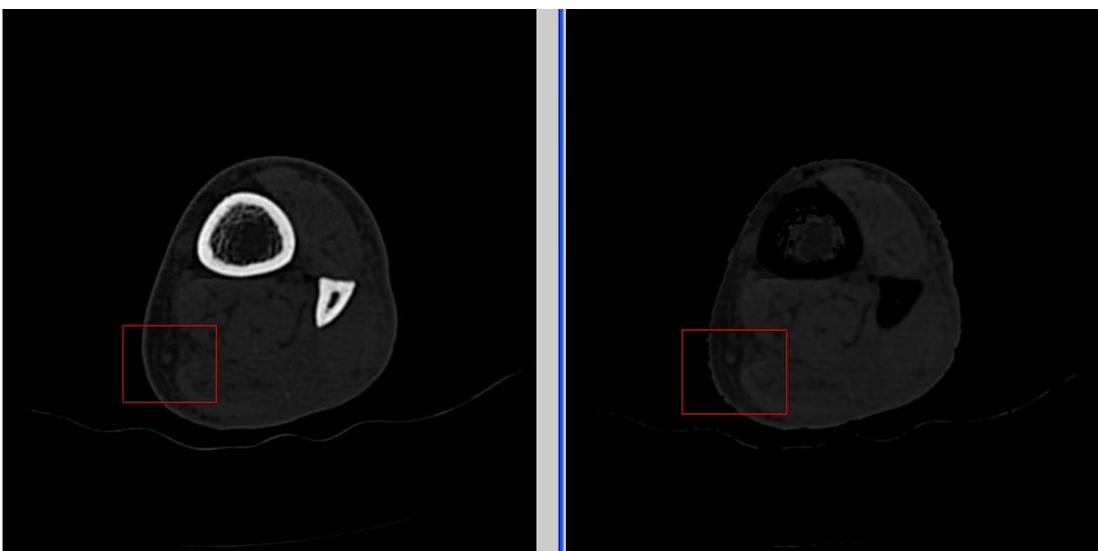


Figura 64: Caso 3. Segmentación por crecimiento de región con umbral 0.04

Como podemos visualizar en las imágenes segmentadas el tejido seleccionado por el usuario es invadido por otros tejidos próximos en relación con el nivel de intensidad de los píxeles, o incluso en el peor de los casos como ocurre en el caso dos y tres, al realizarse la reconstrucción superficial del modelo no se distinguen los diferentes tipos de tejido.

La solución a este problema es aumentar las restricciones, para ello disminuimos el rango de intensidades permitidos en 0.02, donde obtuvimos los resultados satisfactorios.

$$| \text{Sem} - I(x,y) | \leq 0.02 \quad | I(x_i,y_i) - I(x,y) | \leq 0.02$$

Siendo Sem el valor de intensidad del píxel introducido por el usuario,  $I(x,y)$  el valor de intensidad del píxel situado en coordenadas  $(x,y)$  y  $I(x_i,y_i)$  píxeles vecinos incluidos en la región.

Imponiendo de umbral 0.02 y tras realizar un estudio con 260 imágenes los resultados obtenidos son los siguientes:

- 1 Segmenta mal
- 4 Segmenta regular
- 255 Segmenta bien

Tanto a los resultados como las imágenes expuestas les hemos realizado un tratamiento de pos-procesado consistente en la aplicación de dos filtros de mediana. Estos filtros sirven para suavizar los contrastes y las regiones segmentadas, eliminando posibles ruidos.

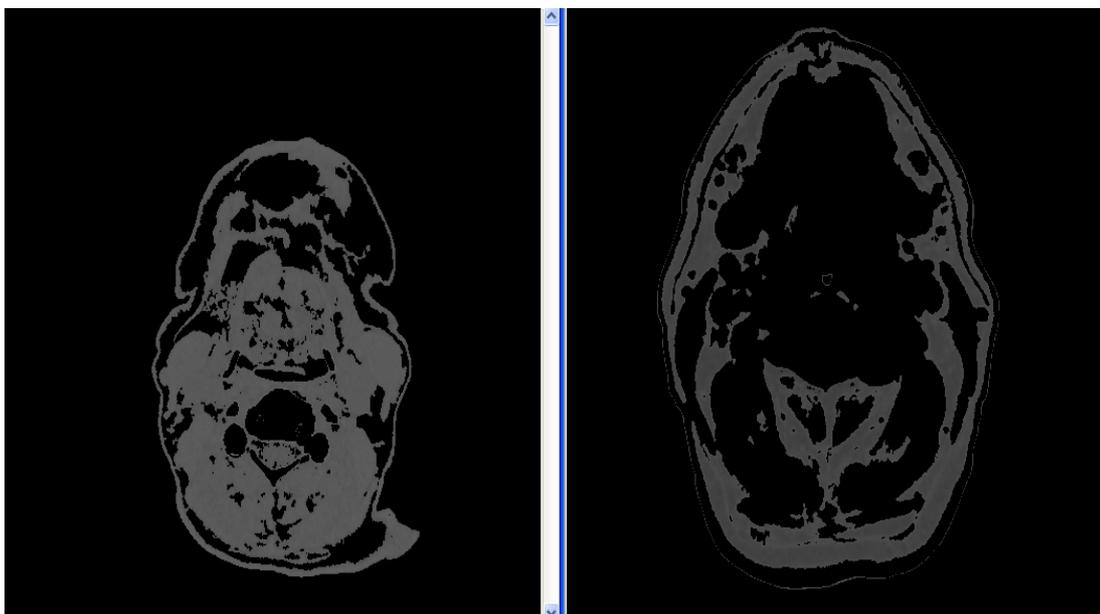


Figura 65: Segmentación por crecimiento de regiones sin filtrado

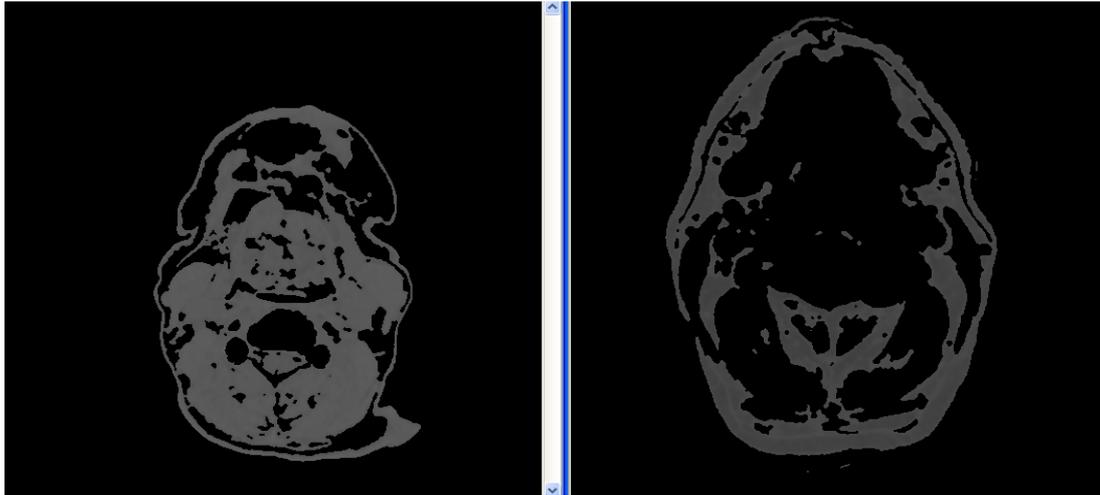


Figura 66: Segmentación por crecimiento de regiones con filtrado de mediana

Finalmente expondremos tabla comparativa con los dos algoritmos de segmentación totalmente automático, el de umbralización y el de crecimiento de regiones para los diferentes tejidos que componen el cuerpo exceptuando hueso:

<b>ALGORITMO DE SEGMENTACIÓN POR UMBRALES</b>	<b>ALGORITMO DE SEGMENTACIÓN POR SEMILLA</b>
Técnica de umbralización	Técnica de crecimiento de regiones por semilla
Umbral de 0.065 en condición inclusión	Umbral de 0.02 en condición inclusión
Tiempo invertido en procesado menor	Tiempo invertido en procesado mayor
Utiliza 4 filtros de mediana en post-procesado	Utiliza 2 filtros de mediana en post-procesado
Segmenta y reconstruye bien la imagen	Segmenta y reconstruye muy bien la imagen, conectividad

Si realizamos una comparación entre ambas técnicas de segmentación, las restricciones en la técnica de segmentación por semilla son mayores (umbral permitido es menor 0.02). Esto es debido a que en este último algoritmo diseñado las regiones crecen a través de la semilla introducida y a través de la incorporación de los vecinos de los píxeles ya incluidos. Así, aunque el rango de intensidad permitido sea menor, se aumentan las capacidades de crecimiento, asegurando además la conectividad entre los elementos de la región segmentada.

### 5.3.4 Reconstrucción tridimensional

Para la **reconstrucción tridimensional** escogeremos el algoritmo de segmentación por crecimiento de regiones, pues aunque el tiempo de CPU invertido en el procesamiento de las imágenes sea un poco mayor que el que consume el algoritmo de umbralización, la segmentación será más efectiva. Esto es debido a la característica de conectividad innata que presentan los algoritmos de crecimiento de regiones por semilla.

La imagen tridimensional digital estará conformada a partir de sucesivos cortes transversales originales a los que le hemos sometido al método de segmentación de crecimiento de regiones por semilla.

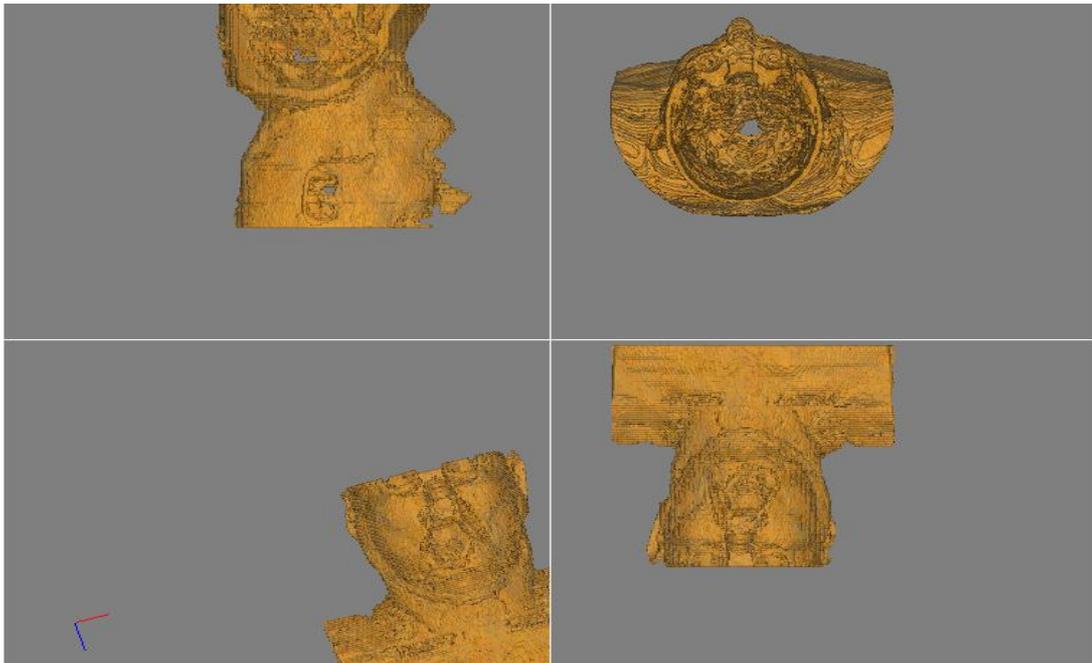
A continuación mostraremos la reconstrucción en 3D y la diferencia con los métodos de segmentación implementados en VirSSPA:

Aplicamos el algoritmo de segmentación diseñado para la reconstrucción de un tejido correspondiente a la grasa humana:



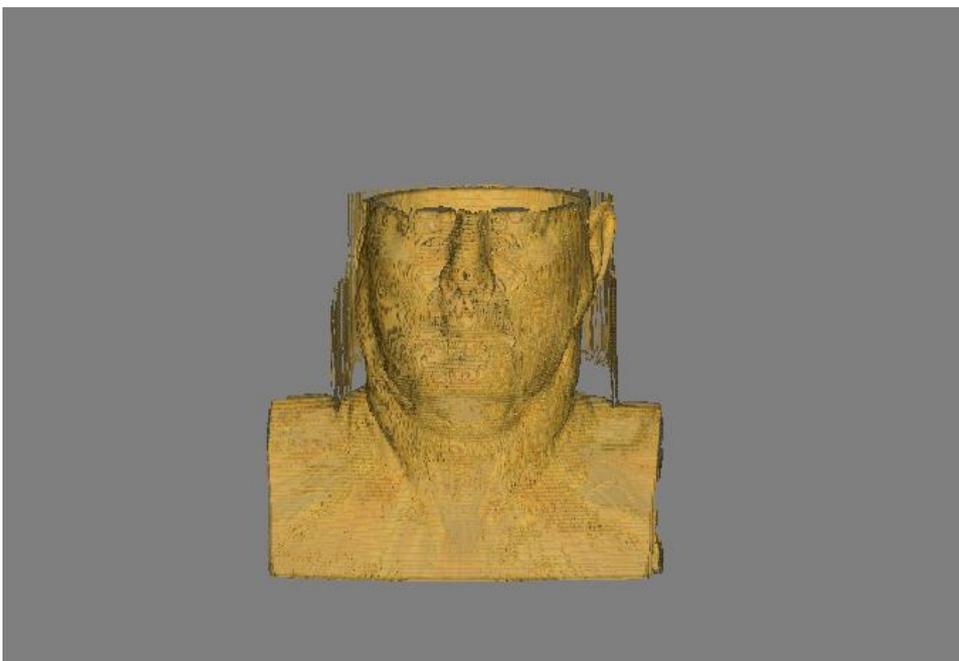
Figura 67: Segmentación correspondiente a tejido graso

Veamos el resultado obtenido en los tres planos, transversal, longitudinal y sagital de la figura



**Figura 68: Resultado segmentación automática en tres planos, transversal, longitudinal y sagital**

El resultado obtenido es bastante bueno y el tiempo empleado es relativamente pequeño, aproximadamente de un minuto. Veamos el resultado y el tiempo invertido en este proceso cuando realizamos la segmentación semiautomática mediante VirSSPA.



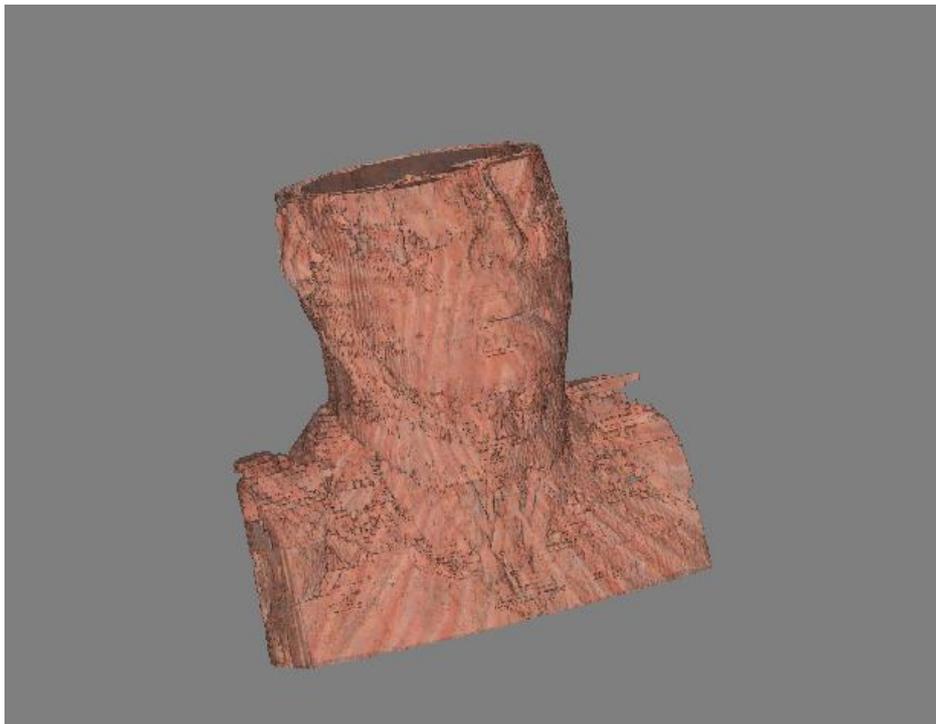
**Figura 69: Segmentación semiautomática utilizando VirSSPA**

Para realizar esta segmentación en VirSSPA hemos tenido que utilizar el algoritmo de crecimiento de regiones por semilla implementado en la herramienta de visualización, incluyendo tres semillas. Hemos tenido problemas para seleccionar los tejidos pues cuando poníamos una semilla en un sitio equivocado, teníamos que empezar de nuevo a realizar la segmentación sin guardar ninguna de las semillas que estaban bien situadas.

Como podemos observar en la figura anterior se segmenta tejido que no corresponde a la zona seleccionada, además aparece en la reconstrucción parte del objeto sobre el que se apoya el paciente al que se le realiza la tomografía, que con el nuevo algoritmo en la medida de lo posible se elimina como podemos observar en la figura 68

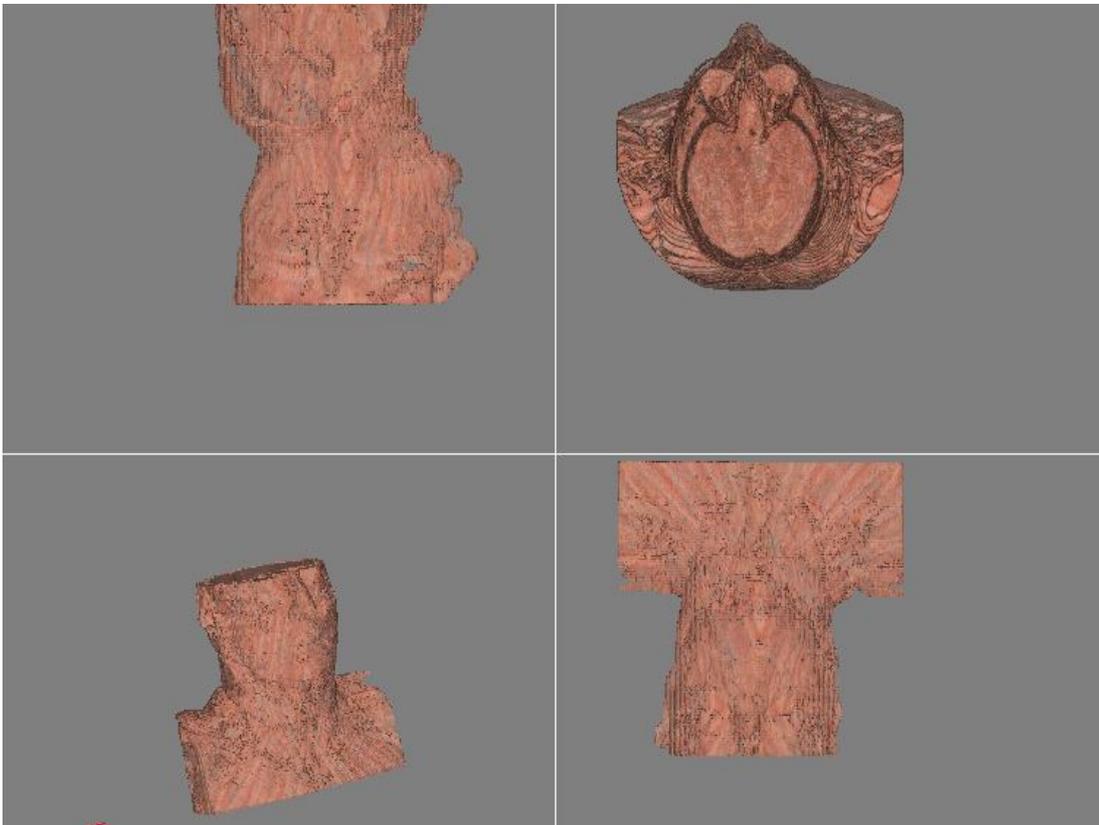
El tiempo estimado en hacer este modelo es de 20 minutos, el cual es bastante superior en comparación con el algoritmo de segmentación automático diseñado que era de un minuto.

Veamos otro caso cuando reconstruimos tejido correspondiente a la zona muscular del paciente con el nuevo algoritmo de segmentación diseñado:



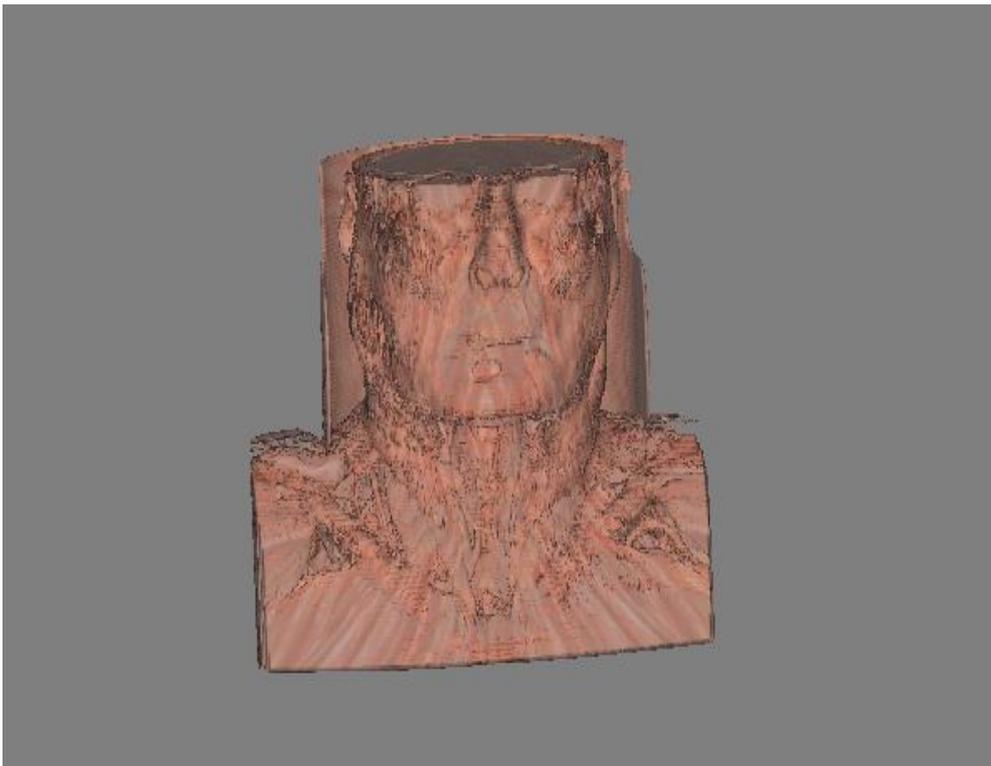
**Figura 70: Segmentación de la zona muscular con el nuevo algoritmo diseñado**

El resultado como podemos visualizar en la figura anterior es bastante bueno. El tiempo estimado en realizar la operación es de un minuto. Veamos el resultado en los tres planos: transversal, sagital y longitudinal



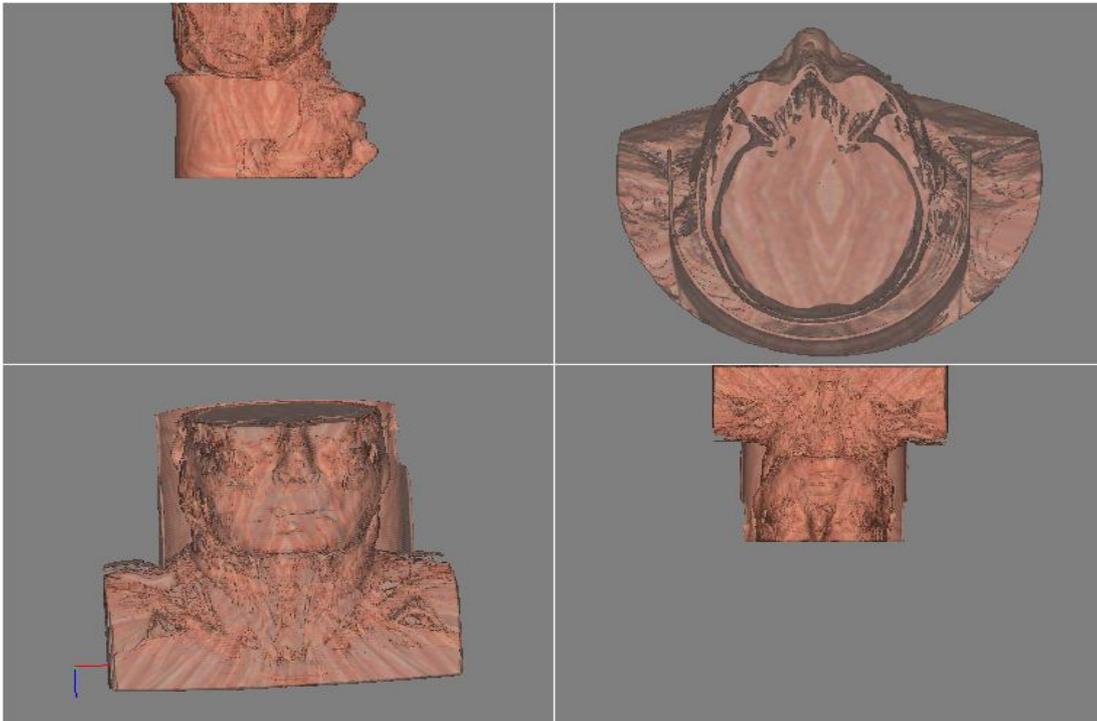
**Figura 71: Segmentación tejido muscular visualizado en tres planos, longitudinal, transversal y sagital con el nuevo algoritmo diseñado**

Veamos la segmentación semiautomática y posterior reconstrucción del tejido muscular realizado con VirSSPA



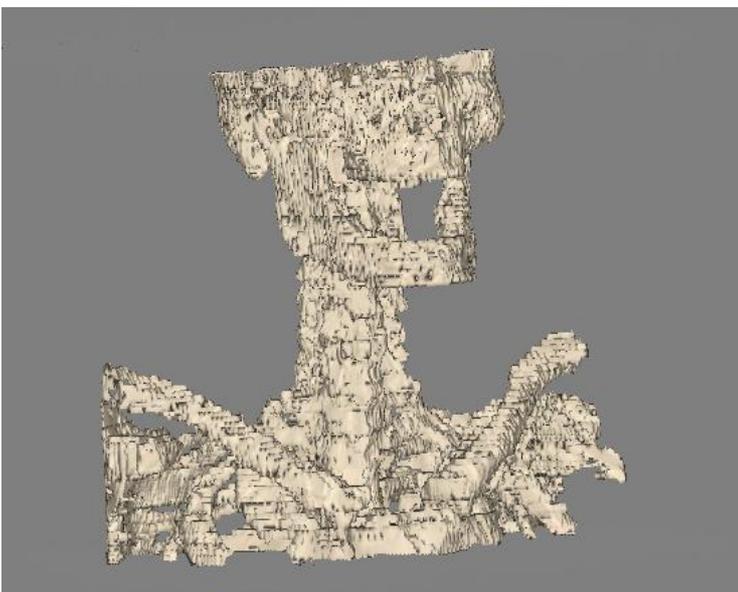
**Figura 72: Segmentación semiautomática de la zona muscular realizada con VirSSPA**

El resultado en los planos longitudinal, sagital y transversal de la aplicación de segmentación semiautomática con VirSSPA se muestra a continuación:



**Figura 73: Reconstrucción del tejido muscular en tres planos, resultado de la reconstrucción semiautomática de VirSSPA**

Veamos la reconstrucción de hueso mediante el algoritmo de segmentación por umbrales con aumento de contraste fijo y la comparación con el algoritmo de segmentación por umbrales con aumento de contraste automático:

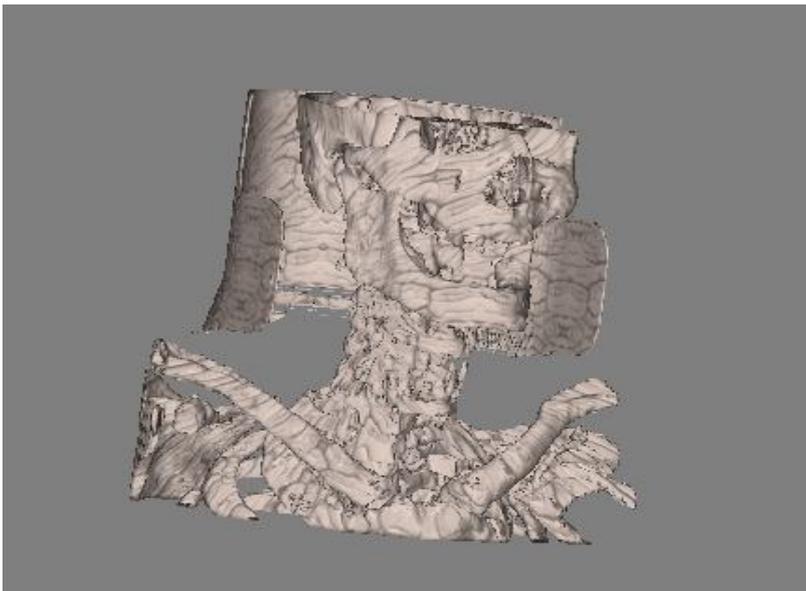


**Figura 74: Segmentación de tejido óseo por umbralización con aumento de contraste fijo**



**Figura 75: Segmentación de tejido óseo por umbralización con aumento de contraste automático**

Como podemos observar en las figuras 74 y 75 se ha mejorado el resultado aplicando segmentación por umbrales con aumento automático. El tiempo estimado en realizar este proceso es de un minuto. Veamos el resultado del algoritmo semiautomático utilizado en VirSSPA:

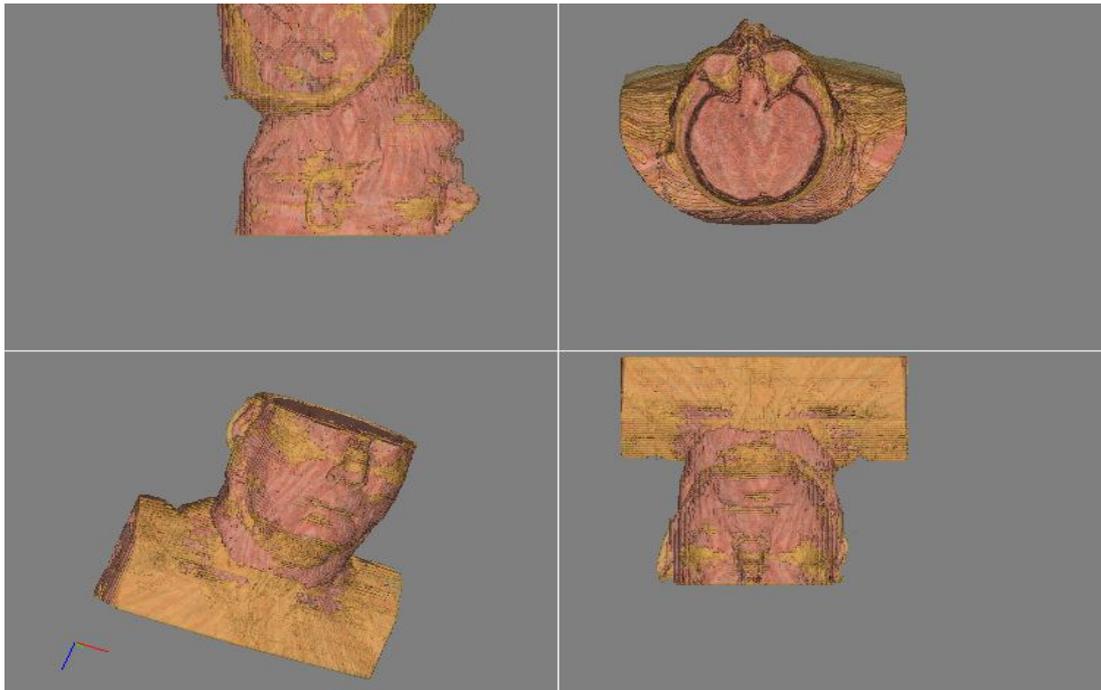


**Figura 76: Segmentación semiautomática realizada con VirSSPA**

El tiempo estimado en realizar el proceso de segmentación y reconstrucción, utilizando segmentación por umbrales semiautomática de VirSSPA es de 10 minutos. También podemos comprobar que tras realizar la segmentación semiautomática parte de la camilla donde se apoya le paciente, es considerada como

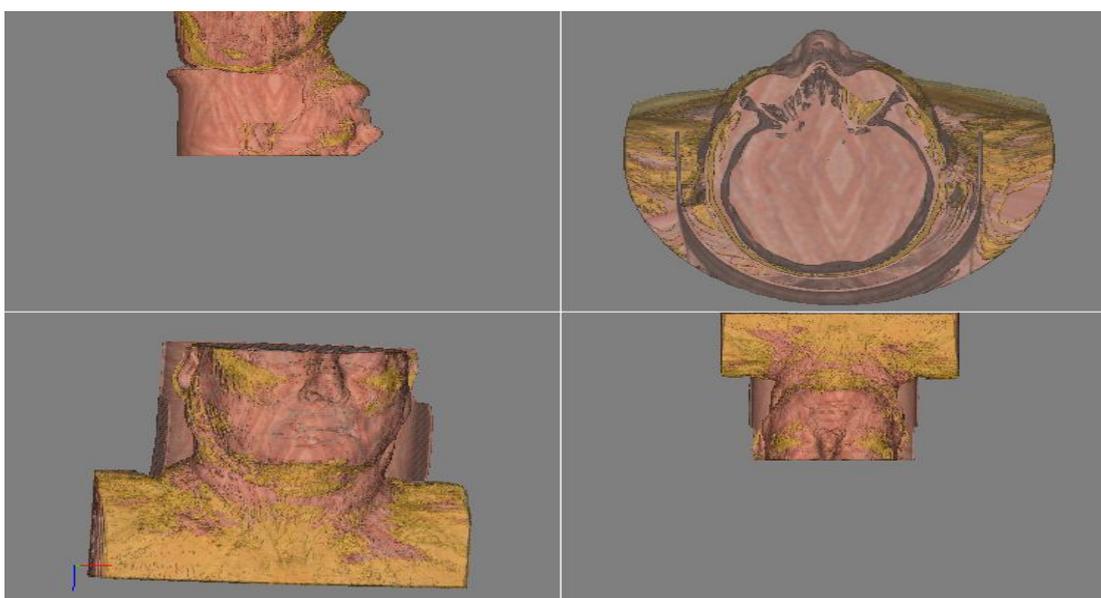
hueso. Este fallo se corrige en el nuevo algoritmo diseñado, como podemos observar en la figura 75.

El siguiente paso será la reconstrucción de los dos tejidos segmentados con el algoritmo diseñado:



**Figura 77: Reconstrucción del tejido muscular y graso en los tres planos correspondiente al nuevo algoritmo diseñado**

El resultado de la aplicación del nuevo algoritmo de segmentación automática es excelente. El tiempo invertido en realizar este proceso y su posterior reconstrucción del modelo es aproximadamente 2 minutos, veamos el resultado con VirSSPA:



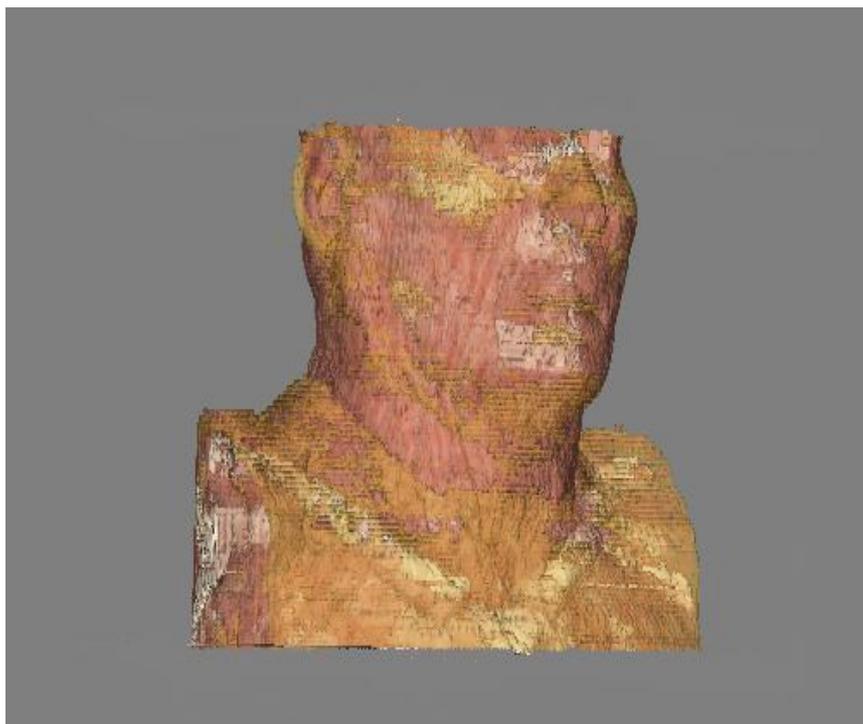
**Figura 78: Reconstrucción del tejido muscular y graso en los tres planos correspondiente al algoritmo semiautomático de VirSSPA**

Podemos apreciar en la imagen anterior que el resultado obtenido con el algoritmo semiautomático que implementa VirSSPA reconstruye más regiones que las correspondiente a cada zona y el tiempo invertido en realizar este modelo los dos tejidos es de 35 minutos.

Si realizamos una comparación entre el algoritmo de segmentación semiautomático de VirSSPA y el nuevo algoritmo de segmentación automático diseñado obtenemos los siguientes resultados:

<i>Algoritmo semiautomático que implementa VirSSPA</i>	<i>Algoritmos segmentación automáticos diseñados</i>
Segmenta más regiones que las de la zonas de interés	Segmenta sólo regiones de interés
Tiempo invertido reconstrucción modelo anterior de 45 minutos	Tiempo invertido reconstrucción modelo anterior de 3 minutos

Mostremos los resultados de las imágenes correspondientes a los tres tejidos reconstruidos, tejido graso, muscular y óseo, con los algoritmos de segmentación automáticos diseñados y con los algoritmos de segmentación semiautomáticos implementados en VirSSPA.

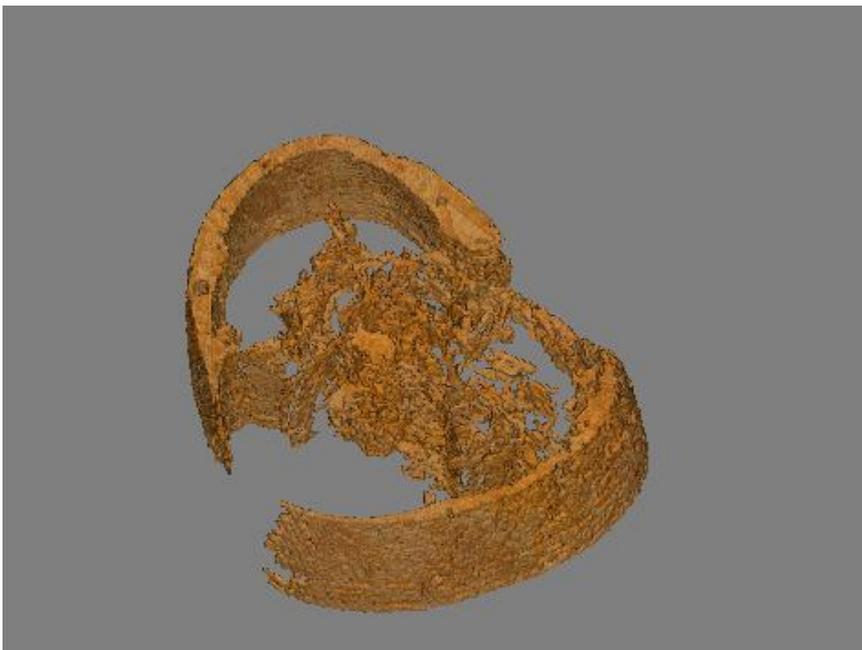


**Figura 79:** Resultado de la reconstrucción y segmentación de los algoritmos diseñados.



**Figura 80: Resultado de la reconstrucción y segmentación de los algoritmos semiautomáticos implementados en VirSSPA**

Veamos otro caso cuando el objeto a segmentar corresponde con una parte de la pierna. Para ello segmentamos primero el tejido correspondiente a la grasa muscular con las técnicas y algoritmos desarrollados en el proyecto



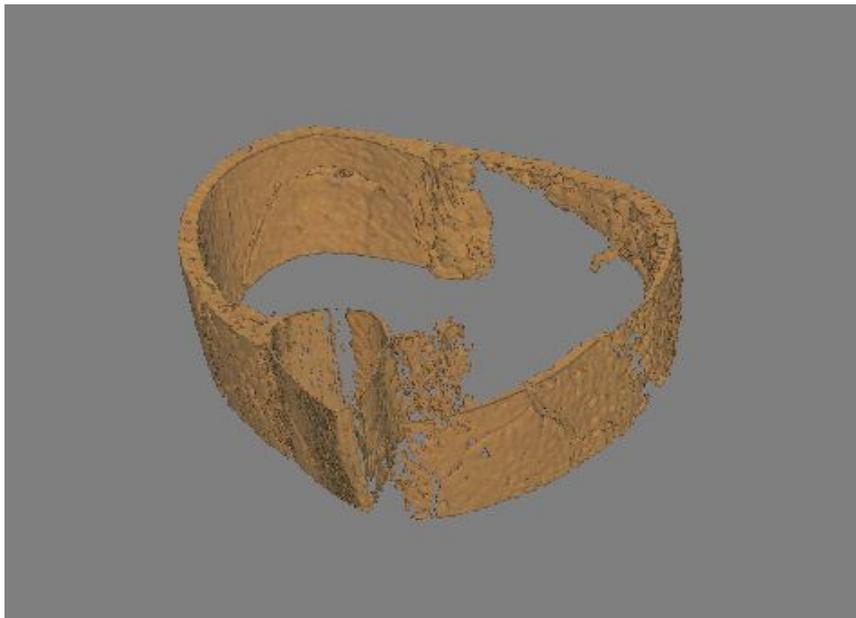
**Figura 81: Segmentación correspondiente a tejido graso con algoritmo diseñado.**

El tiempo estimado en realizar esta reconstrucción de tejido graso de la pierna con las nuevas técnicas desarrolladas fue de un minuto. Veamos este resultado en los tres planos, longitudinal, transversal y sagital en la siguiente figura:



**Figura 82: Segmentación tejido muscular visualizado en tres planos, longitudinal, transversal y sagital con el nuevo algoritmo diseñado**

El resultado que obtenemos al aplicar los algoritmos de segmentación semiautomáticos implementados en la herramienta VirSSPA podemos visualizarlo en la siguiente figura:



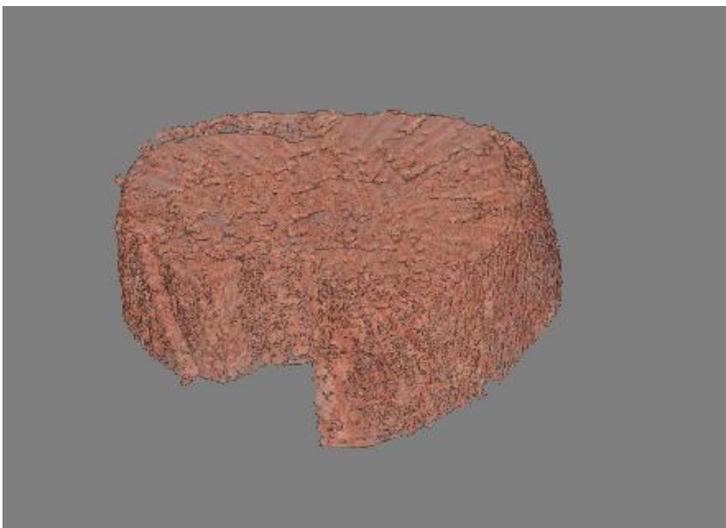
**Figura 83: Segmentación de tejido graso de la pierna mediante algoritmo semiautomático de VirSSPA**

Para realizar esta reconstrucción hemos tenido que colocar de forma manual diez semillas sobre la imagen para realizar la segmentación semiautomática de crecimiento de regiones. El tiempo estimado en realizar esta reconstrucción fue de 15 minutos. Cuando intentamos ver los tres planos de corte del modelo reconstruido con VirSSPA sólo nos permite visualizar un corte como podemos observar a continuación



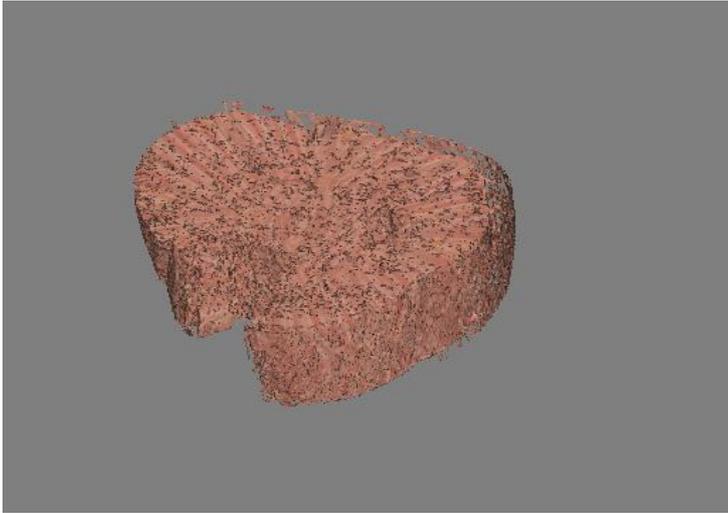
**Figura 84: Segmentación semiautomática implementada con la herramienta VirSSPA en el plano transversal.**

Podemos visualizar en las figuras 83 y 84 que no reconstruye todo el tejido graso de la pierna y el tiempo en realizar el modelo es muy superior al tiempo utilizando los algoritmos diseñados. Veamos que ocurre cuando reconstruimos tejido muscular con los nuevos algoritmos diseñados.



**Figura 85: Segmentación de tejido muscular mediante algoritmo diseñado**

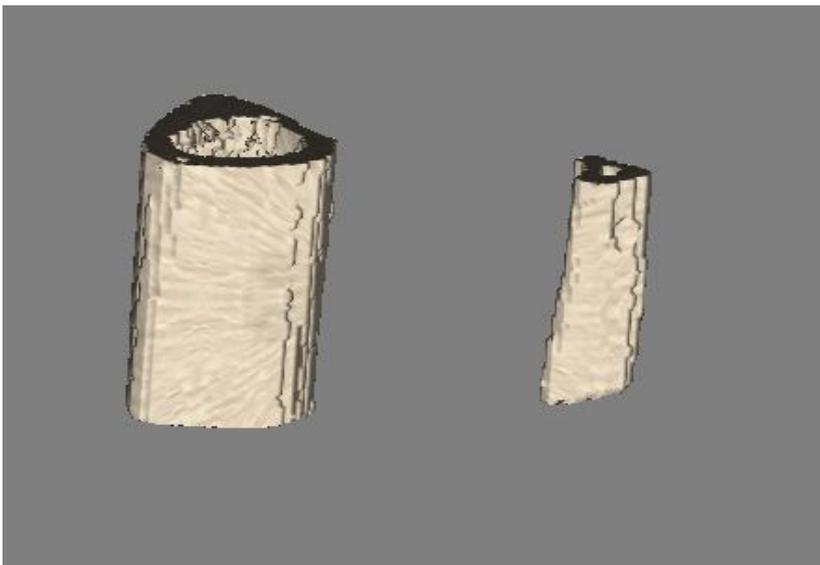
El resultado obtenido con el nuevo algoritmo diseñado es bastante bueno. El tiempo invertido en realizar el proceso es de un minuto. Veamos el modelo obtenido tras la segmentación semiautomática q implementada en VirSSPA:



**Figura 86: Segmentación de tejido muscular mediante algoritmo de crecimiento de regiones por semilla mediante VirSSPA**

Para realizar este modelo muscular hemos tenido que depositar quince semillas. El tiempo invertido en el desarrollo de este modelo muscular fue de quince minutos.

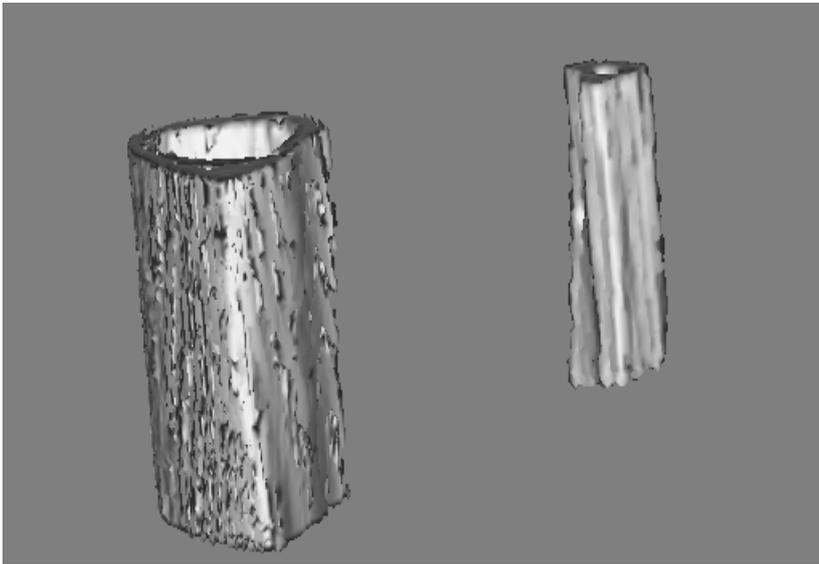
Veamos a continuación el tejido óseo mediante algoritmo por umbralización diseñado:



**Figura 87: Segmentación realizada con el algoritmo diseñado de umbralización.**

El resultado es excelente como podemos visualizar en la figura 86. El tiempo consumido en realizar el modelo es de un minuto. A continuación reconstruiremos el tejido óseo mediante la aplicación

del algoritmo semiautomático por umbrales implementado en VirSSPA:



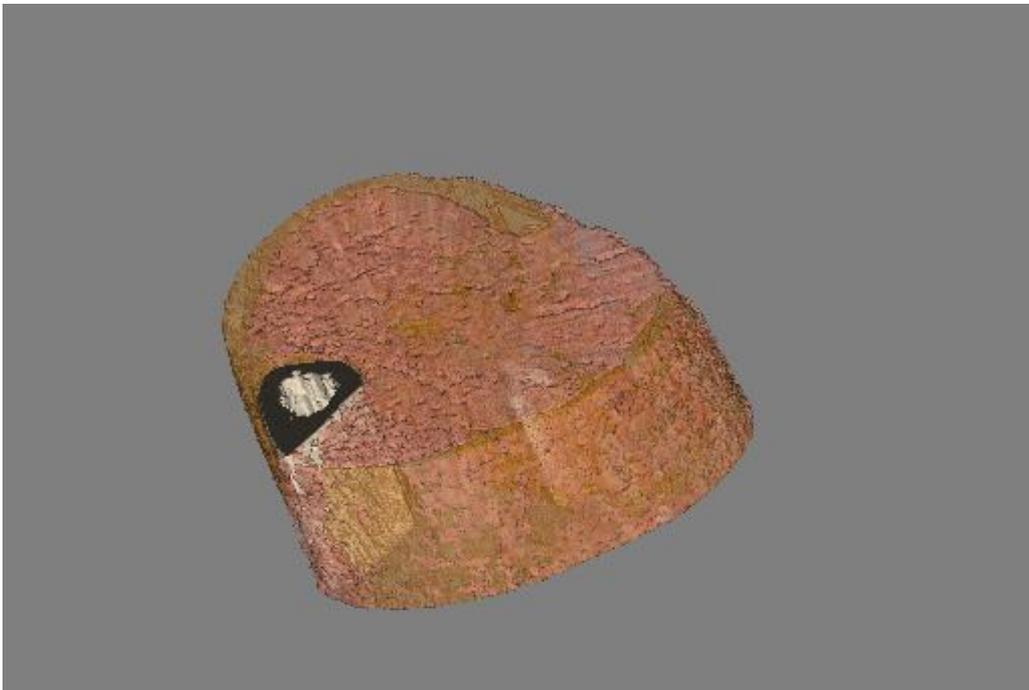
**Figura 88: Segmentación semiautomática por umbrales de VirSSPA**

El tiempo invertido en la realización de este modelo óseo es de diez minutos. Si comparamos la imagen obtenida podemos observar que la reconstrucción es menos precisa, pues aparecen huecos que en el modelo resultante del algoritmo por umbrales diseñado no aparecían (ver figuras 87 y 88).

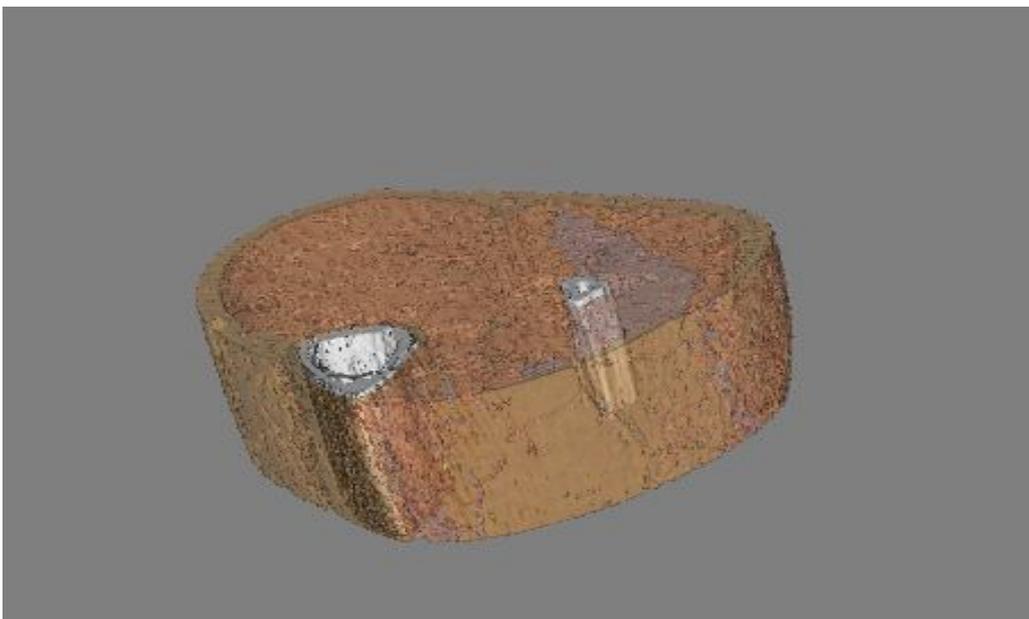
A modo de resumen realizaremos una comparación entre el algoritmo de segmentación semiautomático de VirSSPA y el nuevo algoritmo de segmentación diseñado en el caso de la pierna, obteniendo los siguientes resultados:

<i>Algoritmo semiautomático que implementa VirSSPA</i>	<i>Algoritmos segmentación diseñados</i>
Segmenta menos regiones que las de la zonas de interés	Segmenta sólo regiones de Interés
Tiempo invertido reconstrucción modelo anterior de 45 minutos	Tiempo invertido reconstrucción modelo anterior de 3 minutos

Finalmente mostremos los resultados de las imágenes correspondientes a los tres tejidos reconstruidos, tejido graso, muscular y óseo, con los algoritmos de segmentación automáticos diseñados y con los algoritmos de segmentación semiautomáticos implementados en VirSSPA.



**Figura 89:** Resultado de la reconstrucción y segmentación de los algoritmos diseñados.



**Figura 90:** Resultado de la reconstrucción y segmentación de los algoritmos semiautomáticos implementados en VirSSPA

Podemos comprobar que los resultados obtenidos con los nuevos algoritmos diseñados son más óptimos y eficientes pues realiza una reconstrucción más próxima a la realidad y en menor tiempo si lo comparamos con los algoritmos semiautomáticos existentes en VirSSPA.