

Anexo I

Interfaz gráfica de usuario

La interfaz existente sobre la que se ha trabajado para añadir nuevas etapas, desarrollada por A. Badías [12], es una GUI (Graphical User Interface) creada con Matlab. Sirve para facilitar la comunicación entre Matlab y el usuario, y comprende las fases de reconstrucción de los cuerpos vertebrales y de la instrumentación de fijación del paciente. Para mejor comprensión, haremos referencia a la interfaz anterior como Interfaz 1.0, y a la nueva versión como Interfaz 2.0.

En figura I.1 se puede observar la apariencia de la Interfaz 1.0, sobre la cual se han realizado las modificaciones.



Figura I.1 Apariencia Interfaz 1.0

A pesar de ser, en general, bastante intuitivo, en este anexo se procede a explicar el funcionamiento de la Interfaz 2.0 paso a paso, que a su vez incluye el de la versión anterior, recorriendo todas las etapas de la misma.

Tras ejecutar la interfaz desde Matlab, aparece la ventana principal (figura I.2) en la que se detalla la lista de etapas a seguir en un panel lateral situado a la izquierda. Se puede comparar con la versión anterior de la interfaz, que las etapas añadidas son las correspondientes a los pedículos vertebrales, siguiendo la misma estructura que en las otras (definición de geometría y correspondencia entre figuras). También se observa que existe un panel inferior donde van apareciendo comentarios de seguimiento de la actividad que se está llevando a cabo, como mensajes de confirmación del éxito de la ejecución de una etapa, o mensajes de error.



Figura I.2 Apariencia Interfaz 2.0

El primer paso es cargar las imágenes correspondientes a las radiografías del caso a reconstruir. Para ello hay dos botones ubicados en la parte superior derecha de la interfaz, “*Cargar imagen frontal*” y “*Cargar imagen lateral*”, donde además existe la posibilidad de elegir el formato de carga, entre un formato de imagen habitual o el formato DICOM propio de las radiografías. Una vez cargadas, se obtendrá un aspecto como el de la figura I.3.



Figura I.3 Etapa de carga de las imágenes

Para pasar de una etapa a la siguiente, existe un botón “Continuar” en la parte inferior derecha. Tras la carga con éxito de las imágenes, este botón nos llevará a la etapa de binarización de las imágenes, donde lo primero que la interfaz solicita al usuario es que determine el área donde se encuentra la instrumentación (Figura I.4).

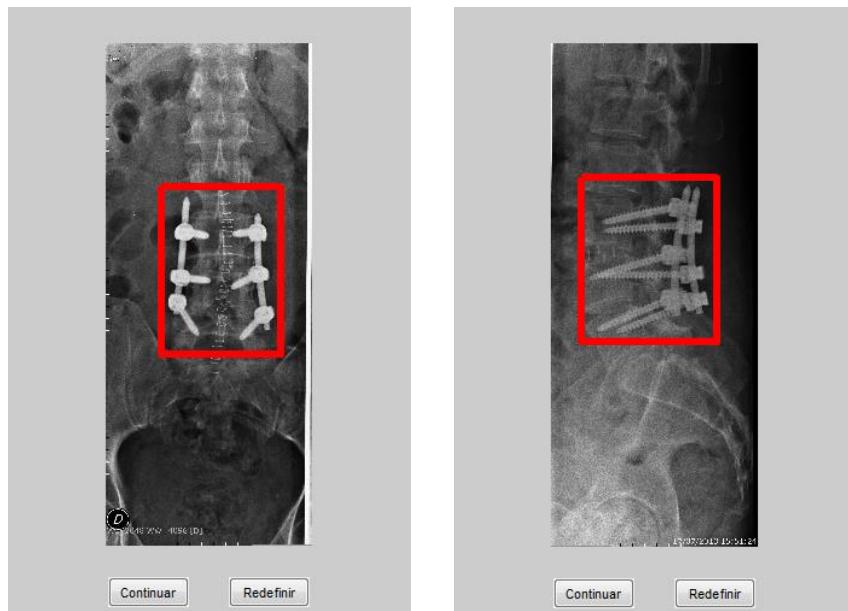


Figura I.4 Etapa de selección del área de trabajo

Una vez definidas estas áreas en las vistas frontal y lateral, se obtiene el resultado que aparece en la figura I.5, donde se encuentran las imágenes binarizadas. Se observa que, a la derecha de las imágenes, existen dos cuadros desplegables, donde puede cambiarse el método de segmentación a utilizar, siendo las posibilidades la binarización mediante umbral, detección de contornos según el algoritmo de Canny [13] y binarización mediante umbral dividiendo la imagen en segmentos horizontales.

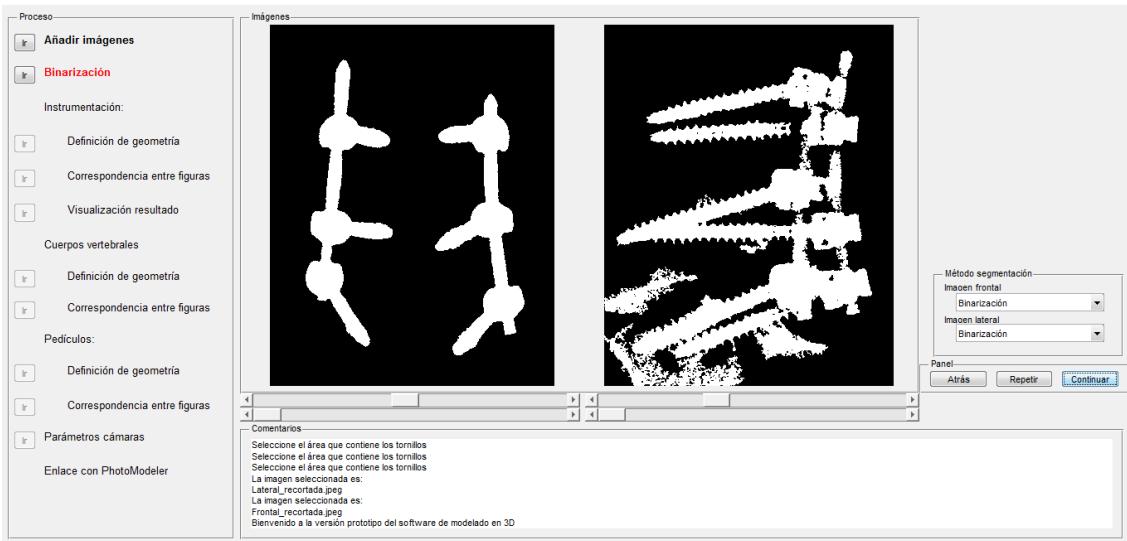


Figura I.5 Etapa de binarización de las imágenes por defecto

Por defecto, ambas imágenes utilizan el método de binarización mediante umbral, donde el umbral se calcula según la escala de grises de la imagen inicial, con el objetivo de diferenciar la instrumentación del resto de elementos. El valor de este umbral se puede modificar mediante el *slider* que se encuentra inmediatamente debajo de cada imagen. Existe otro *slider* debajo de éste, que permite llevar a cabo una operación morfológica de cierre que homogeniza la imagen binaria y elimina el posible ruido, aunque reduce también la exactitud de las aristas del contorno. Por defecto el valor del *slider* es cero, de modo que dicha operación no se lleva a cabo a no ser que el usuario lo modifique.

Cuando el resultado no es el óptimo, como en el caso de la vista lateral en la figura anterior, el usuario puede cambiar entonces el método y obtener unos resultados más adecuados, como se muestra en la figura I.6, donde se ha utilizado el algoritmo de Canny.

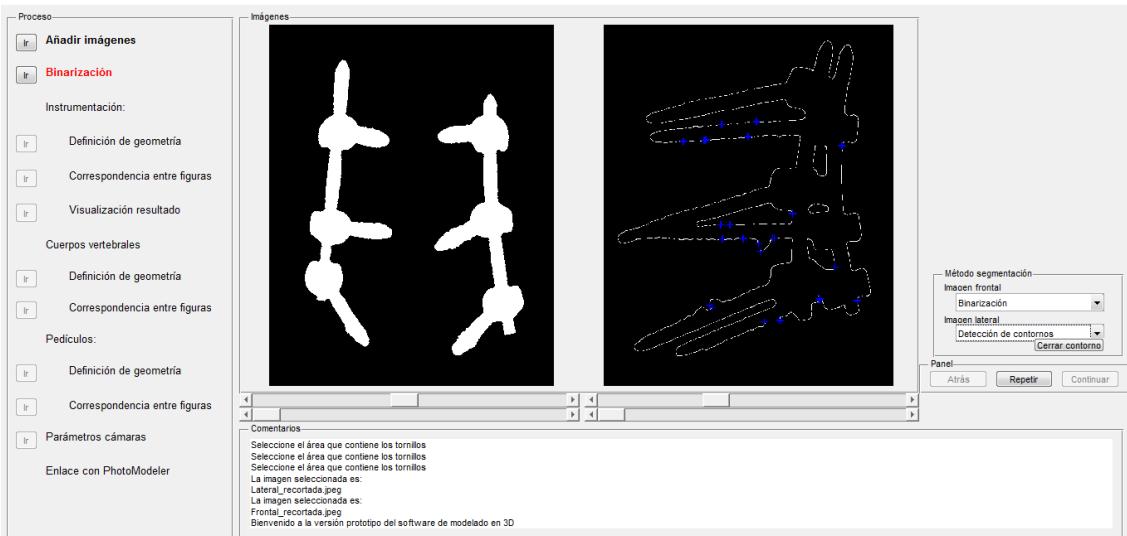


Figura I.6 Etapa de carga binarización de las imágenes, cambio de algoritmo

La utilización de detección de contornos puede dar como resultado contornos abiertos. Si el espacio entre dos contornos que deben ser cerrados es muy elevado, se requiere que el usuario los cierre de forma manual. La etapa donde deben cerrarse manualmente es la que corresponde con la figura I.7, donde estos contornos abiertos son identificados con marcas azules.

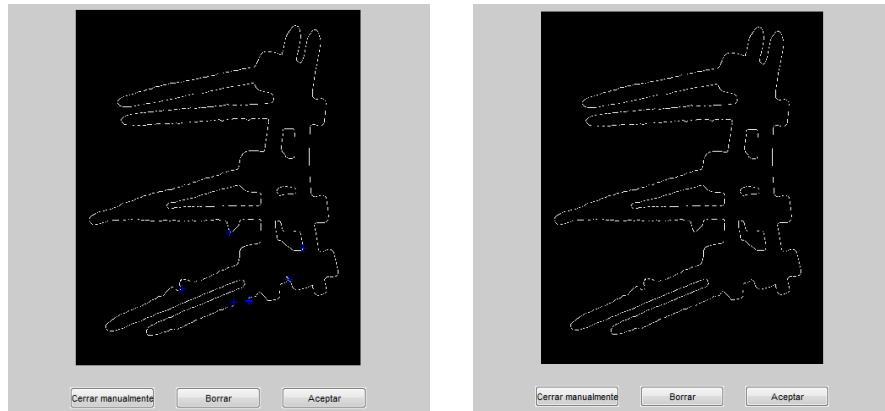


Figura I.7 Etapa de cierre manual de contornos abiertos

En el caso de utilizar el tercer método de binarización separando la imagen en segmentos horizontales, se ha de seleccionar el número de divisiones horizontales como en la figura I.8, y luego variar los *sliders* (figura I.9) hasta obtener el resultado deseado.



Figura I.8 Elección del número de divisiones para la binarización por segmentos

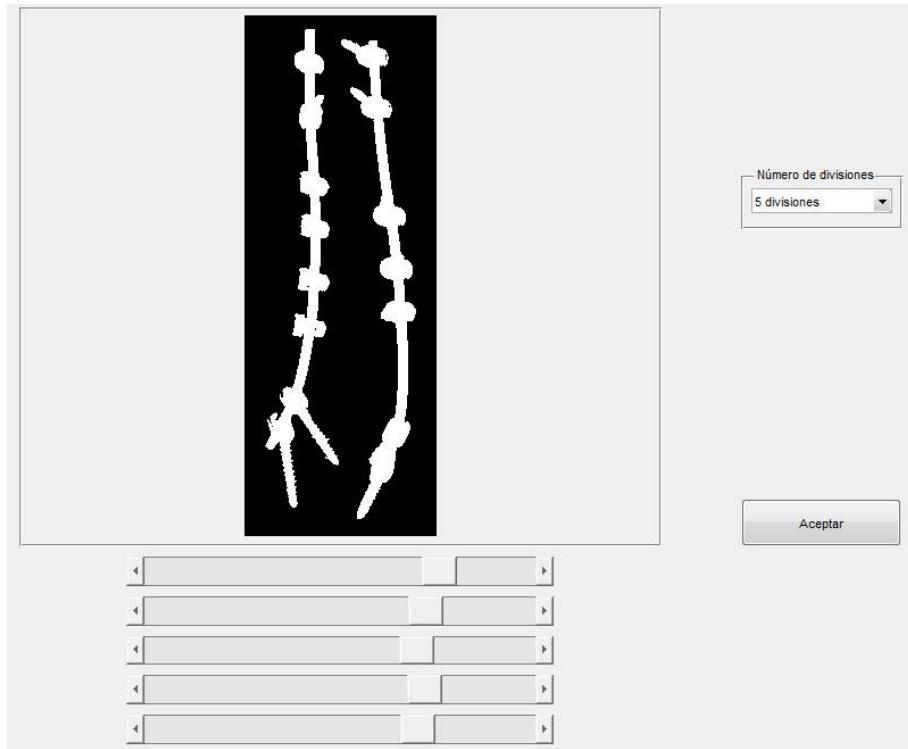


Figura I.9 Modificación del valor de los *sliders* para la binarización por segmentos

Este método es preferible frente a la binarización normal para el caso de radiografías en las cuales se quiere obtener instrumentación a lo largo de toda la columna vertebral.

Cuando se tienen los resultados deseados para ambas vistas, como en la figura I.8, se procede a la identificación de la instrumentación.

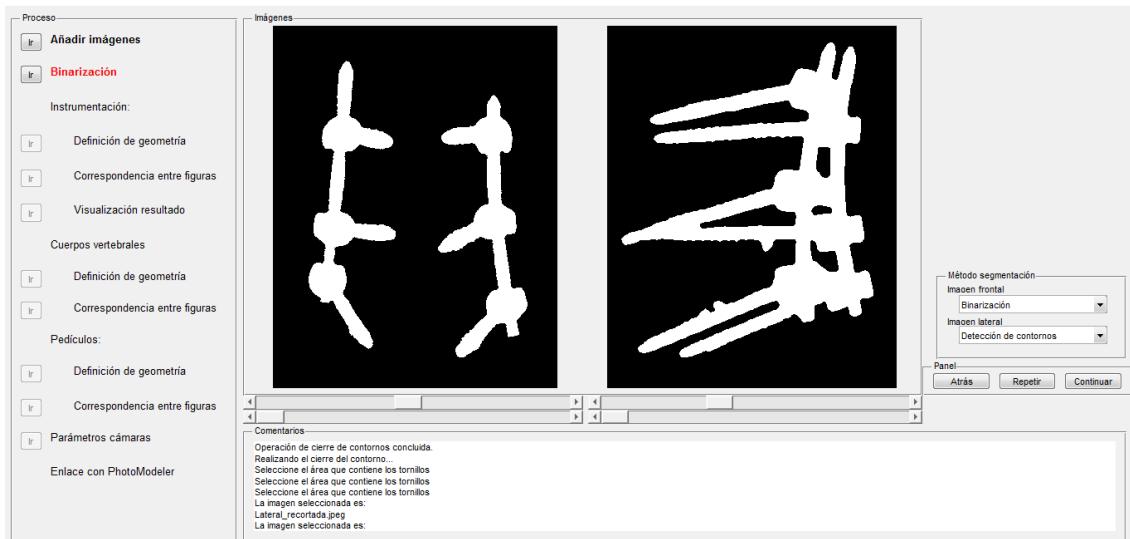


Figura I.8 Resultados de la etapa de binarización

En este proceso de identificación, el primer paso es el de detección de las cabezas de los tornillos, comenzando por la vista frontal (figura I.9). Bajo la imagen, existen dos *sliders* para

facilitar el funcionamiento de esta etapa. Las cabezas de los tornillos deben estar en color rojo mientras que el resto de elementos de la instrumentación deben estar en color verde. El *slider* superior controla el tamaño de las cabezas, que se obtienen mediante operaciones morfológicas de apertura, mientras que el inferior determina el área mínima que debe tener el resto de la instrumentación. Una vez el resultado es el correcto, se pulsa el botón “*Terminar*” para continuar con el proceso.

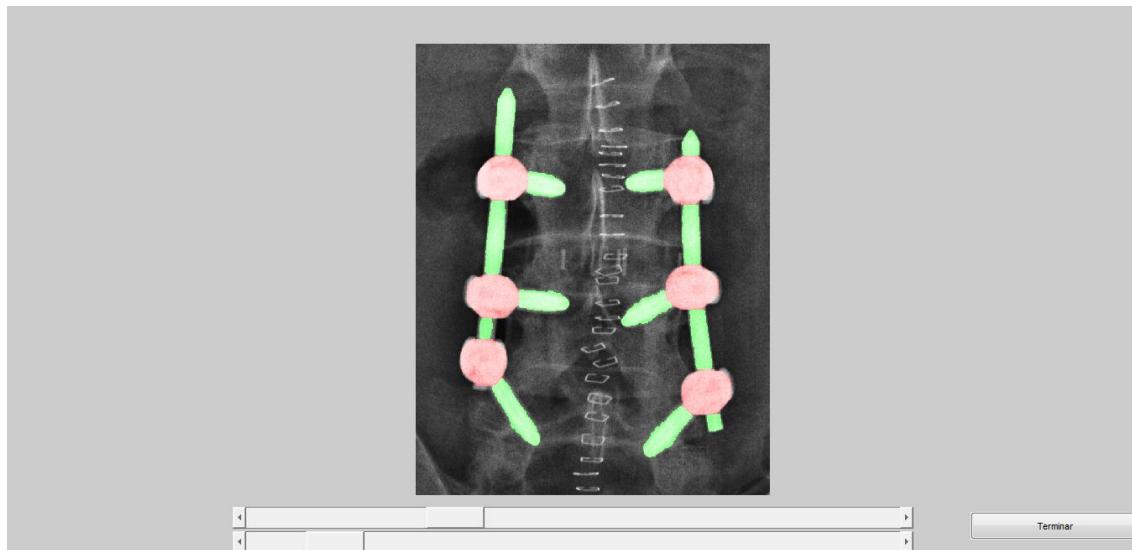


Figura I.9 Etapa de detección de la instrumentación en la vista frontal

Una vez identificadas las cabezas, el sistema calcula si alguna de las detectadas puede ser la intersección de dos de ellas. Si da positivo (figura I.12), el sistema muestra un mensaje de aviso, y aparece una figura con las cabezas donde se puede confirmar o no dichas intersecciones, que aparecen en color rojo. Si no se detecta intersección (figura I.10), aparece igualmente una figura con las cabezas por si se ha de marcar alguna no detectada.

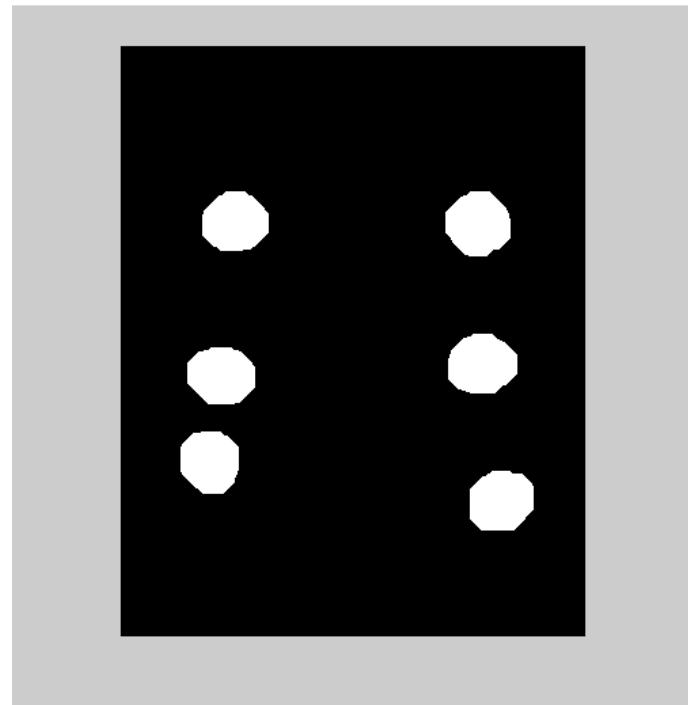


Figura I.10 Confirmación de intersecciones entre cabezas en la vista frontal

El mismo proceso se repite para el caso de la vista correspondiente a la radiografía lateral, como se muestra en las figuras I.11 y I.12.

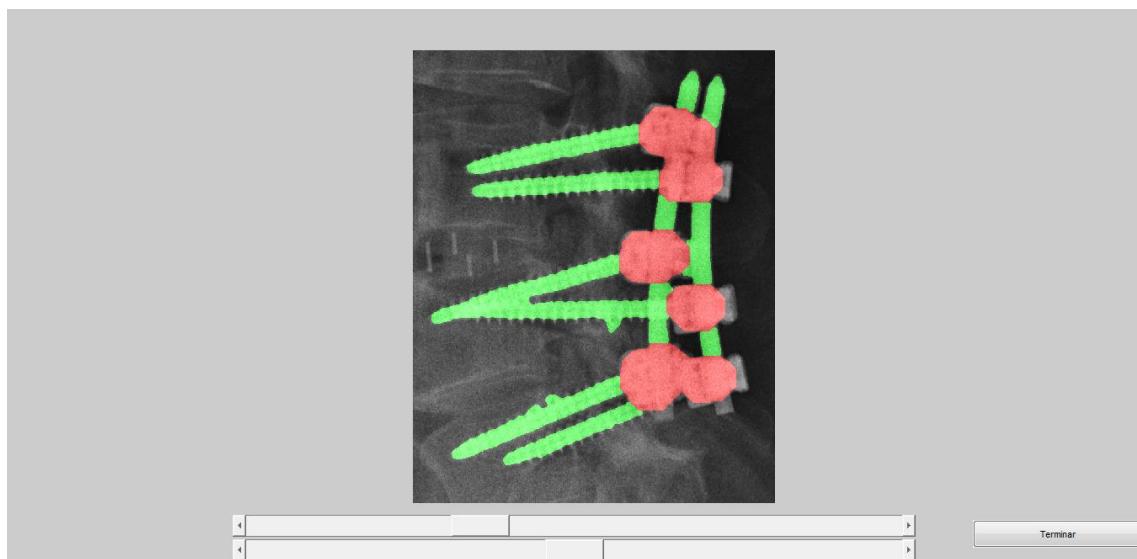


Figura I.11 Etapa de detección de la instrumentación en la vista lateral



Figura I.12 Confirmación de intersecciones entre cabezas en la vista lateral

Para el caso lateral, una vez detectada la geometría de la instrumentación, se procede a diferenciar qué elementos corresponden a una de las hileras de tornillos y cuáles pertenecen a la otra, debido a que puede existir solapamiento entre algunos de sus elementos. El sistema hace una primera diferenciación automática, sobre la cual el usuario puede hacer correcciones de forma manual si un elemento está mal categorizado. La distinción de las dos hileras se presenta como muestra la figura I.13.

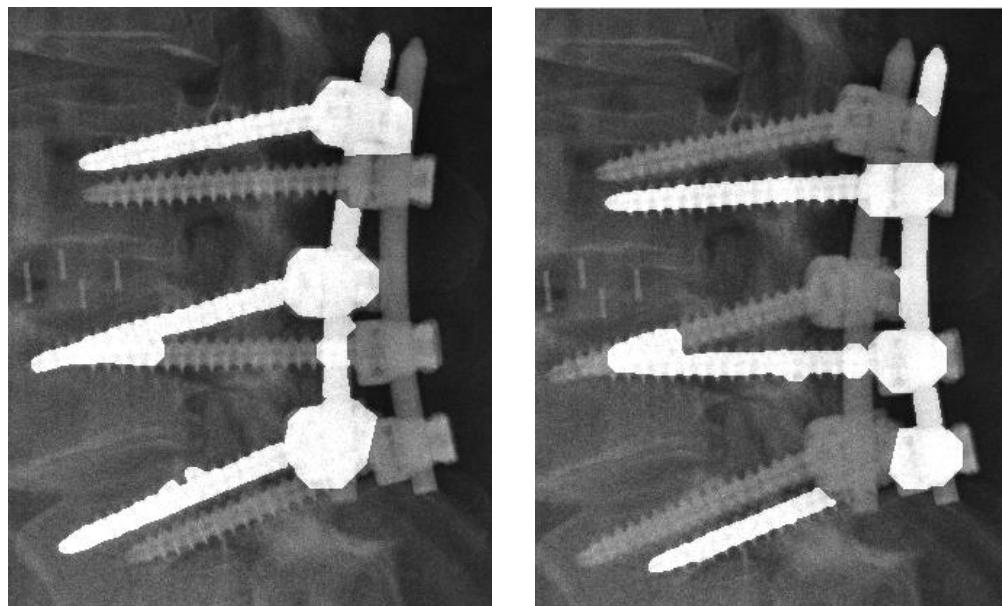


Figura I.13 Identificación de los elementos correspondientes a cada hilera

De nuevo, de forma automática pero con posibilidad de corrección por parte del usuario, el sistema trata de hacer distinción entre los elementos que son tornillos y los que son las barras que sostienen a los tornillos, mostrándose estos en colores rojo y azul, respectivamente (figura I.14).

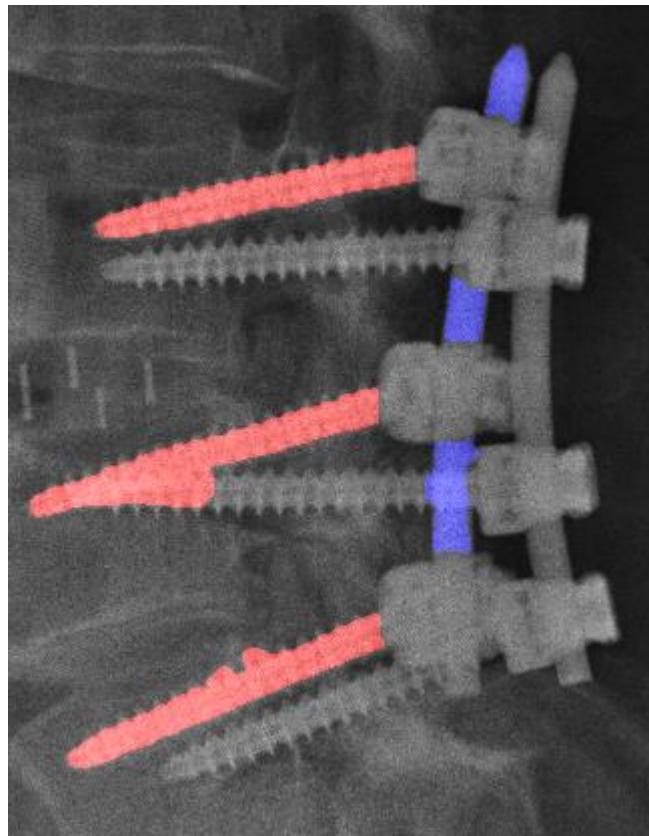


Figura I.14 Distinción entre tornillos y barras

A continuación hay una etapa de confirmación de la unión entre las cabezas y sus respectivos tornillos o barras asociadas. Aparecen sucesivas figuras como la figura I.15, donde el usuario confirmará si la asociación es correcta, o hará las correcciones oportunas si no lo es.

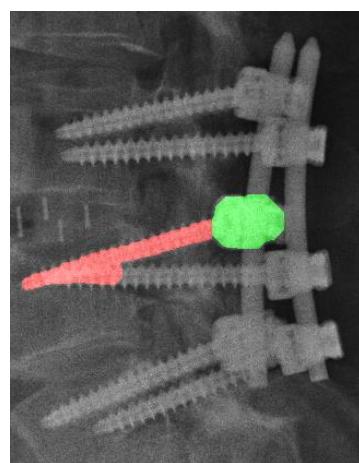


Figura I.15 Asociación entre una cabeza y su tornillo correspondiente

El mismo procedimiento de confirmación y corrección se sigue para las cabezas, tornillos y barras de ambas hileras, hasta que todos los elementos quedan definidos según su categoría, y se visualicen los resultados como en la figura I.16.

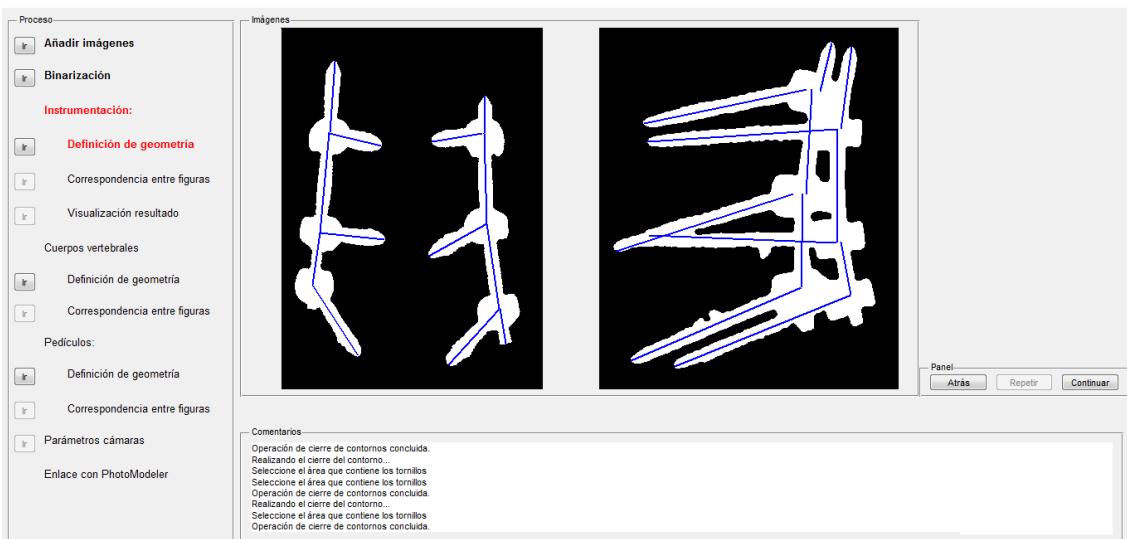


Figura I.16 Resultados de la definición de la geometría de la instrumentación

Para terminar de definir finalmente la instrumentación del paciente, el último paso es establecer una correspondencia entre los elementos de la vista frontal y los de la vista lateral. De forma sencilla, como se muestra en la figura I.17, con el botón “*Fijar elementos*” se deben seleccionar uno por uno los distintos elementos en ambas vistas. Las relaciones establecidas irán mostrándose en el panel lateral de la interfaz.

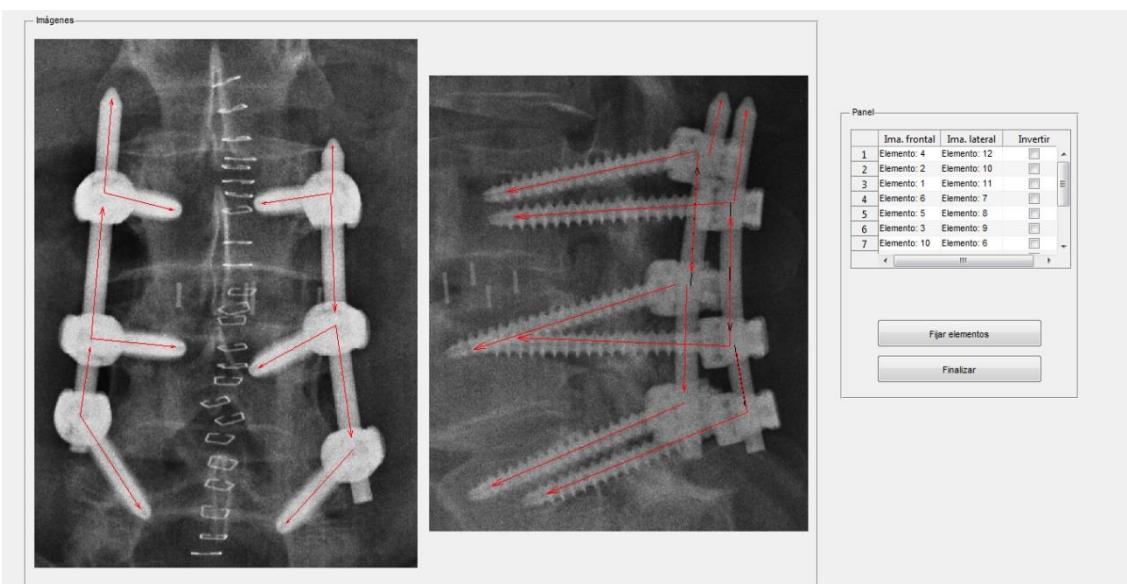


Figura I.17 Etapa de correspondencia entre vistas de la instrumentación

En la figura I.18 se visualizan los resultados sobre las propias imágenes de las radiografías del caso de estudio.



Figura I.18 Visualización del resultado de la definición de la instrumentación

Las siguientes etapas comprenden la definición de los cuerpos vertebrales, comenzando por la etapa de marcado para la extracción de sus contornos. Este marcado, para el caso frontal, debe realizarse con 5 puntos sobre cada vértebra, comenzando por la esquina superior izquierda, y continuando en sentido contrario a las agujas del reloj, como se muestra en la figura I.19.

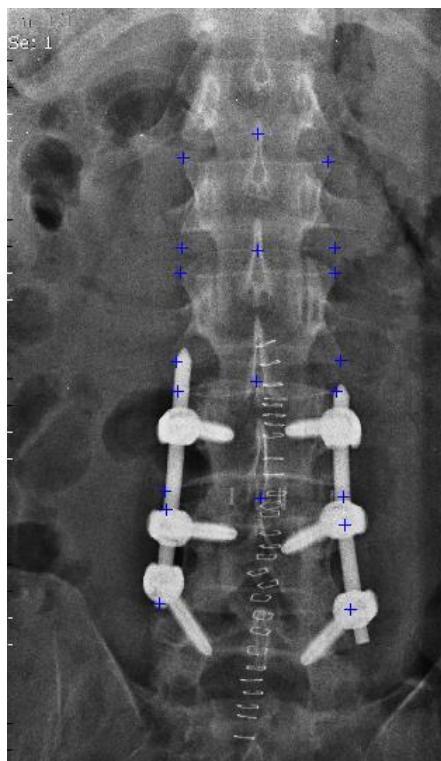


Figura I.19 Marcado de los cuerpos vertebrales en la vista frontal

Para el caso lateral será solo necesario marcar tres puntos, de nuevo comenzando por la esquina superior izquierda y continuando en sentido contrario a las agujas del reloj (figura I.20).

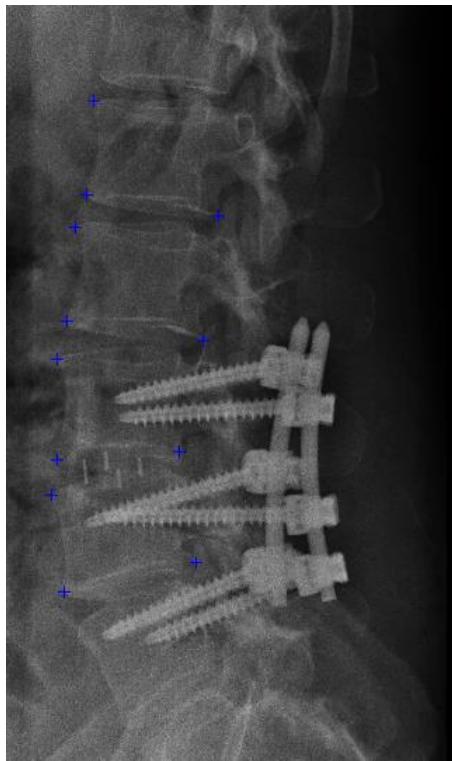


Figura I.20 Marcado de los cuerpos vertebrales en la vista lateral

En este punto comienza el proceso de extracción de contornos mediante AAM [14] de forma automática, obteniendo los resultados que se visualizan en la figura I.21.



Figura I.21 Visualización de los contornos obtenidos de los cuerpos vertebrales

Del mismo modo que con la instrumentación, se establecen las correspondencias entre los cuerpos vertebrales (figura I.22).

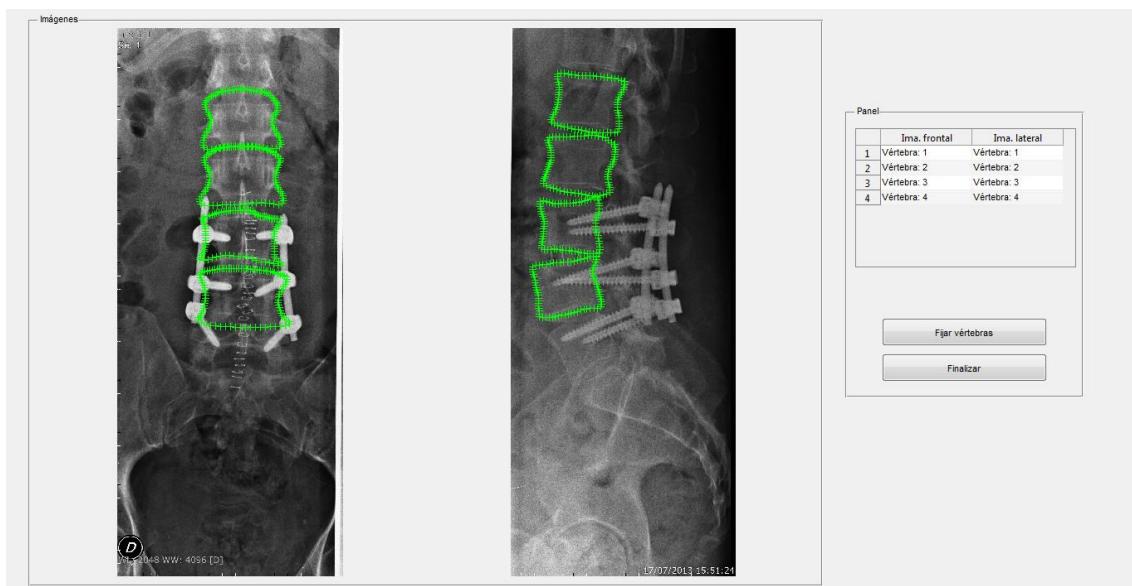


Figura I.22 Etapa de correspondencia entre vistas de los cuerpos vertebrales

Una vez definida la instrumentación y los cuerpos vertebrales, el último elemento a determinar son los pedículos. El proceso es análogo al de la definición de las vértebras, es decir, marcado, procesamiento con ASM y relación entre pedículos en las dos vistas.

En el marcado de la vista frontal se marcan dos puntos, el superior e inferior del pedículo, en ese orden, como se observa en la figura I.23.

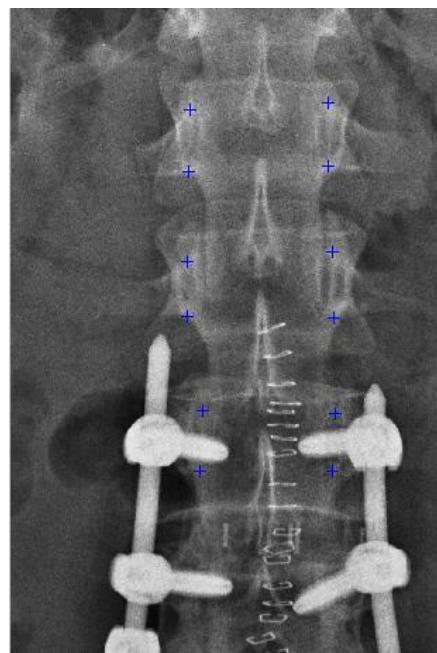


Figura I.23 Marcado de los pedículos en la vista frontal

Para la vista lateral se deben marcar cuatro puntos, comenzando por la esquina superior derecha y continuando en sentido contrario a las agujas del reloj (figura I.24).

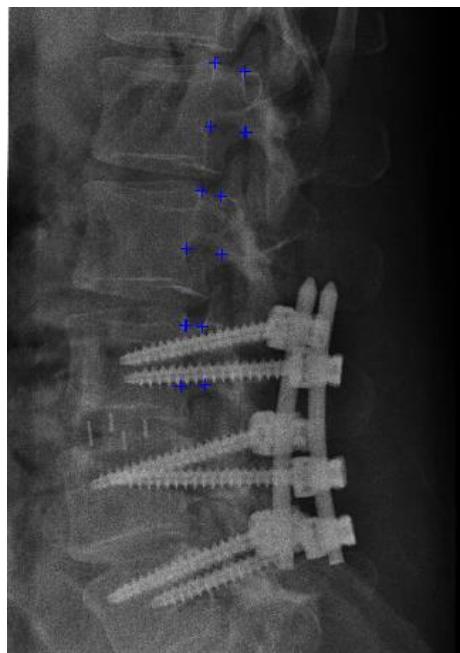


Figura I.24 Marcado de los pedículos en la vista lateral

Tras ejecutarse el algoritmo de ASM, se obtienen los resultados sobre las radiografías que se observa en la figura I.25, y se establece a continuación la correspondencia entre pedículos como en la figura I.26.



Figura I.25 Visualización de los contornos obtenidos de los pedículos



Figura I.26 Etapa de correspondencia entre vistas de los pedículos

El último paso antes de enlazar con el software de PhotoModeler es la introducción de los parámetros correspondientes al equipo con el que se han tomado las radiografías, información necesaria para la generación del modelo tridimensional.

Estos parámetros, que son la distancia focal y el tamaño de pixel en vertical y horizontal, se introducen en una tabla (figura I.27), donde de forma automática se calculan el tamaño y el centro de la imagen. Si las imágenes han sido cargadas en formato DICOM, estos parámetros se introducen de forma automática.

Una vez realizado este paso, y pulsando el botón de “Continuar”, se procede al traslado de la información a PhotoModeler y a su procesamiento, terminando así la función que desempeña la interfaz.

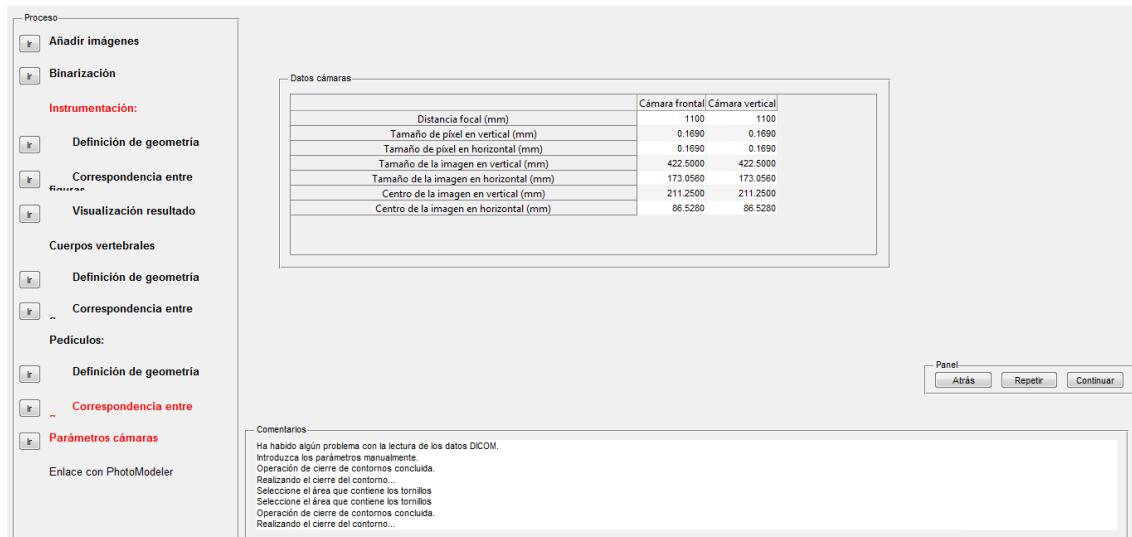


Figura I.27 Etapa de introducción de los parámetros de las cámaras

Anexo II

Redefinición de la geometría 3D

Debido a que sólo se utilizan dos vistas para generar el modelo tridimensional de los cuerpos vertebrales mediante intersección de la proyección de sus vistas, la forma de la pieza obtenida tiene diferencias con la forma real, de modo que se utiliza un criterio de mejora de su geometría [12] para que la forma obtenida sea más cercana a la realidad.

Esta redefinición de la geometría se hace a través de Matlab, y se basa en una geometría estándar de las vértebras, que varía en función del tipo de vértebra (cervical, torácica o lumbar). Como se muestra en la figura II.1, la distancia C (que representa el radio de la circunferencia que describe parte de la forma de la vértebra), equivale a dos tercios de la suma de las distancias B y C , que corresponden con la profundidad del cuerpo vertebral.

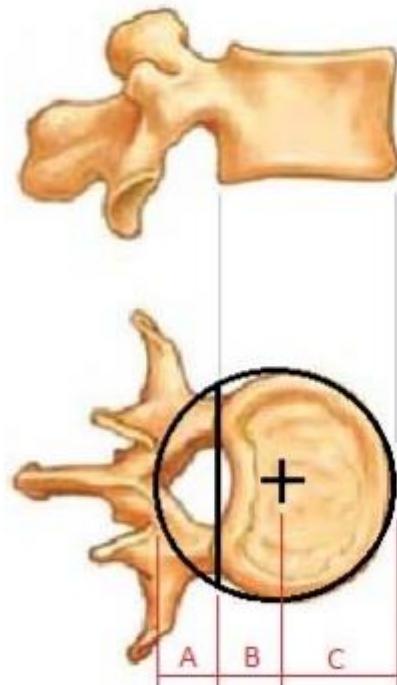


Figura II.1 Estandarización de la geometría externa del cuerpo vertebral

Así, se deduce que la distancia B es un tercio de la profundidad. Por diferencia de las distancias C y B se obtiene la distancia A . Esta última será la que se utilice para corregir la geometría de las vértebras, que se adaptará a la de un cilindro de altura de la vértebra, al que se le ha recortado la parte correspondiente a esta distancia A .

En la figura II.2 (a) se puede observar la pieza correspondiente a un cuerpo vertebral sin modificar, tal cual se obtiene mediante la interfaz, y en la figura II.2 (b) se muestra el resultado tras modificar su geometría.

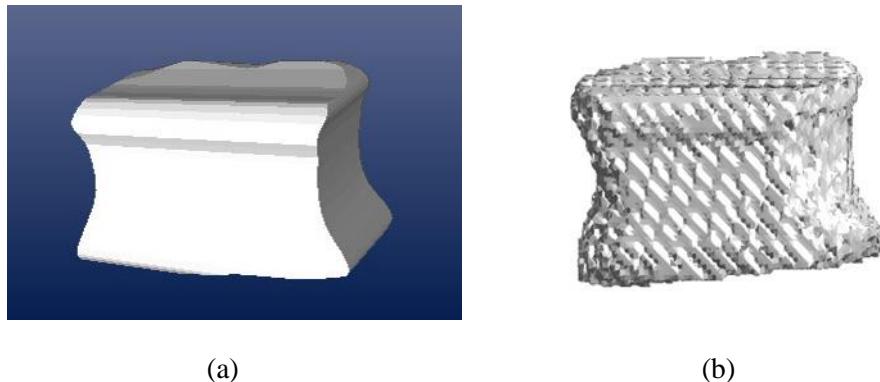


Figura II.2 Modificación de la geometría vertebral: (a) Vértebra original; (b) Vértebra modificada;

Anexo III

Descripción del software utilizado

La vía principal de comunicación con el usuario es mediante interfaces gráficas de usuario creadas desde Matlab. Estas interfaces constan de dos archivos, uno que contiene el código que describe su funcionamiento (en formato *.m*) y otro que contiene el diseño gráfico de la interfaz (formato *.fig*).

Partiendo de un software desarrollado por A. Badías [12], se han creado las funciones relativas a la parte de los pedículos, tratando de guardar la mayor similitud posible a los archivos de la interfaz anterior, de modo que quede un conjunto final bien integrado. Para una mejor comprensión, se hace una explicación del software al completo, que combina los archivos anteriores y los nuevos.

Así, para este proyecto existe una interfaz principal, llamada *Interfaz2*, que a su vez hace llamada durante su ejecución a otros cuatro archivos de interfaz: *binarizacion_segmentos*, *relaciona*, *relaciona_vertebras* y *relaciona_pediculos*. Esta subordinación, así como el objetivo de cada interfaz, se muestran en la figura III.1.

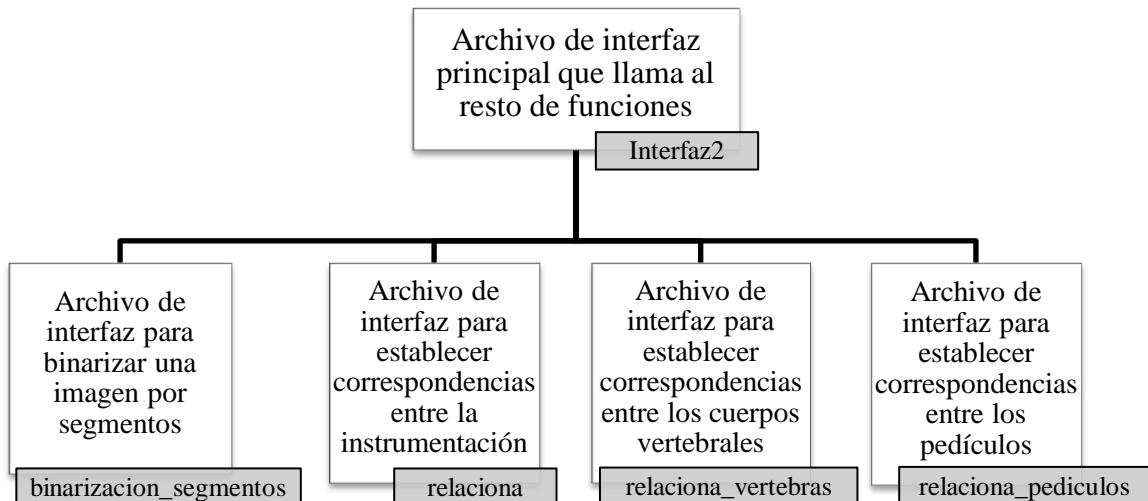


Figura III.1 Archivos de interfaz

Aunque buena parte del código está introducido en los archivos de interfaz, existen también funciones externas a las que llama desde estos archivos principales. A continuación, en la tabla III.1, se encuentra una descripción de la función que cumple cada una de ellas.

Función	Descripción
angulo2d.m	A partir de dos puntos en 2D, define el ángulo de giro de la recta que los une con respecto a la horizontal.
areacuadro.m	Permite la selección del área de la instrumentación mediante un cuadro marcado en color rojo.
ASM_pediculos_frontal.m	Lanza el proceso de marcado de los puntos de un pedículo frontal, normaliza el tamaño de la imagen obtenida y llama a la función de lanzamiento del código ASM.
ASM_pediculos_lateral.m	Lanza el proceso de marcado de los puntos de un pedículo lateral, normaliza el tamaño de la imagen obtenida y llama a la función de lanzamiento del código ASM.
AAM_vertebras_frontal.m	Lanza el proceso de marcado de los puntos de una vértebra frontal, normaliza el tamaño de la imagen obtenida y llama a la función de lanzamiento del código AAM.
AAM_vertebras_lateral.m	Lanza el proceso de marcado de los puntos de una vértebra lateral, normaliza el tamaño de la imagen obtenida y llama a la función de lanzamiento del código AAM.
borrapixeln.m	Función para pintar un píxel en rojo en una imagen.
bucleseparaobjetosmalident.m	Función que devuelve una matriz de dimensiones [pixelesy, pixelesx, nº objetos separados], donde se almacena una imagen para cada objeto separado, puesto que al haber intersecciones entre ellos, es posible que comparten píxeles en común, por lo que es necesario separar las imágenes (cada píxel de cada imagen sólo puede contener un valor).
cierracontorno.m	Función que une los huecos del contorno de una figura. Para ello, se pasa una imagen con los elementos etiquetados. Siempre y cuando la distancia entre los extremos sea menor a la mitad de “límite_recuadro” se realizará la unión. Un valor elevado de “límite_cuadrado” puede provocar un gran incremento del tiempo de cálculo. Es necesario pasar también como parámetro de entrada la variable “extremo” devuelta de la función “contornosabiertos”.

confirmaelementoshilera1.m	Función que muestra la imagen inicial y le solapa los posibles tornillos en azul y barras en rojo que se hayan reconocido de modo semiautomático. Permite la variación entre tornillos y barras entre los elementos de la hilera correspondiente.
confirmaelemtoscabezas.m	Función que muestra la imagen inicial y le solapa un elemento (tornillo o barra) en rojo y las cabezas correspondientes en verde que se hayan reconocido de modo semiautomático. Permite la variación entre cabezas para cada elemento de la hilera correspondiente.
confirmaobjetoshilera1.m	Función que muestra la imagen inicial y le solapa los posibles objetos que se hayan reconocido de modo semiautomático. Permite la selección de más objetos o la eliminación de otros.
confirmaobjetoshilera2.m	Función que muestra la imagen inicial y le solapa los posibles objetos que se hayan reconocido de modo semiautomático. Permite la selección de más objetos o la eliminación de otros.
contornosabiertos.m	Función que detecta y muestra por pantalla los contornos abiertos de una figura.
datosespureos.m	Función que devuelve los datos espúreos de un vector según un cálculo de los outliers por cuartiles. El vector debe ser de tipo columna (dimensión n x 1), y devuelve un vector del mismo tamaño con unos en la posición de los valores espúreos (si los hay).
DrawLines_2Ends.m	Dibuja una recta entre dos puntos según los parámetros [x1, x2, y1, y2].
ejecuta_AAM.m	Fija las opciones del algoritmo del modelo AAM, carga el archivo con los datos de entrenamiento de las vértebras y ejecuta el código para la imagen lateral.
ejecuta_ASM_ped.m	Fija las opciones del algoritmo del modelo ASM, carga el archivo con los datos de entrenamiento de los pedículos y ejecuta el código para la imagen lateral.
ejecuta_AAMf.m	Fija las opciones del algoritmo del modelo AAM, carga el archivo con los datos de entrenamiento de las vértebras y ejecuta el código para la imagen frontal.

ejecuta_ASMf_ped.m	Fija las opciones del algoritmo del modelo ASM, carga el archivo con los datos de entrenamiento de los pedículos y ejecuta el código para la imagen frontal.
ejecutacodigohilerai.m	Ejecuta toda la parte de identificación de elementos de la hilera específica.
elementosprimerahilera.m	Función que devuelve una variable de tipo <i>struct</i> con los objetos que pertenecen a la primera hilera de tornillos.
elementossegundahilera.m	Función que devuelve una variable de tipo <i>struct</i> con los objetos que pertenecen a la segunda hilera de tornillos.
extrarectguifrontal.m	Función que extrae un rectángulo de la imagen frontal y guarda en disco la información.
extrarectguilateral.m	Función que extrae un rectángulo de la imagen lateral y guarda en disco la información.
extremosejelong.m	Función que obtiene los puntos extremos del eje mayor de la elipse que coincide con los momentos segundos de un objeto binario en cuestión.
ginput_flecha.m	Función que devuelve los puntos y el botón clickado en una imagen sin modificar el puntero (flecha).
leeimagen.m	Lee y adapta una imagen para ser tratada sea cual sea su formato.
modifica_contornos_vertebras.m	Permite importar los cuerpos vertebrales obtenidos con PhotoModeler y modificar su geometría.
muestracontornos.m	Función previa a contornosabiertos.m que borra las etiquetas de extremo de objetos formados por un único píxel.
pintapixeln.m	Función para pintar un píxel de azul en una imagen.
pixelesenrecta.m	Función que modifica el valor de todos los puntos pertenecientes a una recta comprendida entre dos puntos en una imagen.

pixelsvox.m	Obtiene las coordenadas de los vértices de una nube de puntos binaria.
polygon2voxel.m	Convierte la superficie de un objeto formada por polígonos en un volumen (una matriz).
ptointers2rectas.m	Función que devuelve el punto de intersección de dos rectas que se cortan.
quepixel2.m	Función para determinar las coordenadas de dos píxeles en concreto de una imagen. Click con el botón izquierdo para seleccionar, click con el botón derecho para confirmar.
quepixel3.m	Función para seleccionar los 3 píxeles extremos de la vértebra vista lateralmente. Click con el botón izquierdo para seleccionar un punto. Click con el botón derecho para borrar el punto. Click con el botón central para poder seleccionar la herramienta “mano” o la herramienta “zoom”.
quepixeln.m	Función para determinar las coordenadas de un número “n” de píxeles en concreto de una imagen.
quepixelped.m	Función para seleccionar los 2 píxeles extremos de un pedículo visto frontalmente. Click con el botón izquierdo para seleccionar un punto. Click con el botón derecho para borrar el punto. Click con el botón central para poder seleccionar la herramienta “mano” o la herramienta “zoom”.
quepixelpedlat.m	Función para seleccionar los 4 píxeles extremos de un pedículo visto lateralmente. Click con el botón izquierdo para seleccionar un punto. Click con el botón derecho para borrar el punto. Click con el botón central para poder seleccionar la herramienta “mano” o la herramienta “zoom”.
quepixelvertfr.m	Función para seleccionar los 5 píxeles extremos de la vértebra vista frontalmente. Click con el botón izquierdo para seleccionar un punto. Click con el botón derecho para borrar el punto. Click con el botón central para poder seleccionar la herramienta “mano” o la herramienta “zoom”.
read_wobj.m	Lee objetos en formato Wavefront (.obj).

reconsima.m	Función que reconstruye la imagen de los tornillos con el tamaño inicial de la imagen total.
reconsptos.m	Función que lleva una serie de puntos colocados en la imagen reducida a la posición correspondiente en la imagen inicial (tamaño mayor).
recta2ptos.m	Define los parámetros “m” y “n” de una recta $y=mx+n$ en función de dos puntos de entrada.
rellenacontorno.m	Función que rellena internamente un contorno cerrado con valor true a partir de un píxel de entrada.
sdvc.m	Función que devuelve el factor de corte para una distribución introducida como un vector columna, para separarla en dos distribuciones que ocupan cada una la mitad del rango de todos los datos.

Tabla III.1 Descripción de las funciones externas a la interfaz

Además de los archivos de interfaz y de las funciones externas, existen también los archivos que contienen los datos de entrenamiento de los modelos activos para las vértebras y pedículos, tanto en vista frontal como en lateral. Estos archivos se cargan automáticamente en las funciones correspondientes, y son los que se reflejan en la tabla III.2.

Archivo	Descripción
Data_AAM_vert_frontal.mat	Archivo que contiene los datos de entrenamiento del cuerpo vertebral para la vista frontal.
Data_AAM_vert_lateral.mat	Archivo que contiene los datos de entrenamiento del cuerpo vertebral para la vista lateral.
Data_ASM_ped_frontal.mat	Archivo que contiene los datos de entrenamiento de los pedículos para la vista frontal.
Data_ASM_ped_lateral.mat	Archivo que contiene los datos de entrenamiento los pedículos para la vista lateral.

Tabla III.2 Descripción de las funciones externas a la interfaz

Referencias

- [1] Cheng L, Wang J, Zeng Z, Zhu R, Yu Y, Li C, Wu Z. *Fijación con tornillos pediculares para las fracturas traumáticas de la columna torácica y lumbar*, Cochrane Database of Systematic Reviews 2013 Issue 5. Art. No.: CD009073. DOI: 10.1002/14651858.CD009073
- [2] [\[2\] http://sve.bmap.ucla.edu/instructions/metrics/dice/](http://sve.bmap.ucla.edu/instructions/metrics/dice/)
- [3] Chan T., Vese L. *Active contours without edges*, IEEE Trans Image Process. 2001 Feb;
- [4] Lankton, S., Tannenbaum A., *Localizing Region-Based Active Contours*, IEEE Trans. On Image Processing, Vol. 17, NO. 11, Noviembre 2008.
- [5] T. F. Cootes, C. J. Taylor, D. Cooper, and J. Graham, *Active shape models – their training and application*, Computer Vision and Image Understanding 61, pp. 38-59, Jan. 1995.
- [6] Johnson R. A., Wichern D. W., *Multivariate Statistics, A Practical Approach*, Chapman & Hall, London/New York, 1988.
- [7] Dirk-Jan Kroon, *Active Shape Model (ASM) and Active Appearance Model (AAM)*, MATLAB Central File Exchange. Enero 2012.
- [8] Adams R., Bischof L., *Seeded Region Growing*, IEEE Trans. On Pattern Analysis and Machine Intelligence, Vol. 16, NO. 6, Junio 1994.
- [9] Daniel, *Region Growing (2D/3D grayscale)*, MATLAB Central File Exchange. Agosto 2011.
- [10] PhotoModeler Help File, Building a 3D Model, Creating surfaces, Import Silhouettes.
- [11] PhotoModeler Help File, Programming PhotoModeler - DDE Interface, Command Structure.
- [12] Badías A., *Modelado y medida en 3D a partir de estudios radiológicos simples de pacientes intervenidos de la columna vertebral mediante tornillos pediculares.*, Proyecto de Fin de Carrera dirigido por José J. Guerrero, Universidad de Zaragoza, 2014.
- [13] Canny. J., *A computational Approach To Edge Detection*, IEEE Trans. Pattern Analysis and Machine Intelligence, 8(6):679-698, 1986.
- [14] T. F. Cootes, G. J. Edwards and C. J. Taylor, *Active Appearance Models*, Proc. European Conference on Computer Vision, 1998.