Facultade de Informática
UNIVERSIDADE DA CORUÑA

TRABALLO FIN DE GRAO
GRAO EN ENXEÑARÍA INFORMÁTICA
MENCIÓN EN COMPUTACIÓN

Aplicación de métodos de procesado digital de imagen para la realización automatizada de medidas en imagen médica.

Estudante: Raúl Saavedra Campos
Dirección: Julián Alfonso Dorado De La Calle
Marcos Gestal Pose

A Coruña, xuño de 2021.

A mi madre y a mi padre

Agradecimientos

Agradecer, en primer lugar, a los profesores Julián y Marcos, que han hecho posible este trabajo. En segundo lugar, me gustaría agradecer a mi familia y amigos que, con su apoyo, de igual manera han sido imprescindibles para que yo pueda llegar hasta aquí y seguir adelante.

Resumen

El procesado digital de imagen es un campo que ha experimentado un gran crecimiento en los últimos años, estando ya presente en diferentes aspectos de nuestra vida cotidiana. Un ejemplo sería el ámbito médico, donde ya se empiezan a utilizar herramientas de diagnóstico automatizado a partir de imágenes.

Estos sistemas de procesado se componen de diferentes etapas, siguiendo una serie de pasos que permiten extraer la información de interés en la imagen. Con esta información de interés y con el apoyo en una base de conocimiento, el sistema sería capaz de calcular unas conclusiones y mostrarlas al usuario.

El propósito de este trabajo consiste en desarrollar una herramienta de procesado digital de imagen en diferentes etapas que trabaje sobre imagen médica. El sistema creado proporcionará asistencia a un médico, realizando mediciones sobre dos imágenes de una rodilla. El objetivo es automatizar la tarea de la toma de medidas de la Tuberosidad Tibial Anterior - Garganta de la Tróclea, habitualmente realizada de forma manual, ahorrando tiempo de trabajo al personal médico.

Abstract

Digital image processing is a field which has experienced a great growth in the last years, taking part in different aspects of our daily lives. An example would be the medical area, where automated diagnostic tools using images are already beginning to be used.

These processing systems are made up of different stages, following a series of steps that allow the extraction of the information of interest of an image. With this information of interest and with the support of a knowledge base, the system would be able to calculate some conclusions and show them to the user.

The purpose of this project is to develop an image processing tool composed of different stages that works on medical image. The created system will provide assistance to a doctor, taking meassurements on two images of a knee. The objective is to automate the task of meassuring the Tibial Tuberosity - Trochlear Groove distance, usually carried out manually, saving work time for medical staff.

Palabras clave:

- Procesado digital de imagen
- Medición automatizada
- Imagen médica
- Tuberosidad Anterior de la Tibia
- Garganta de la Tróclea
- ImageJ

Keywords:

- Digital image processing
- Automated measurement
- Medical image
- Tibial Tuberosity
- Trochlear Groove
- ImageJ

Índice general

Índice de figuras

Índice de tablas

Capítulo 1

Introdución

La visión es sin duda el sentido más empleado por la especie humana, además de ser uno de los mecanismos de procesamiento de imágenes más poderosos que existe.

La imagen digital [1] es una traducción, una representación bidimensional de una imagen empleando bits, mientras que su procesado consiste en alterar la información visual para obtener mejores resultados o extraer alguna característica en concreto.

Se puede decir que los primeros intentos de manipulación, almacenamiento y transmisión de imágenes comienzan en la década de 1920 [2], con un sistema de transmisión de fotografías a través de códigos telegráficos que comunicaba la prensa de Londres y Nueva York a través de un cable oceánico. Pero si hablamos de imagen digital asociada a la informática no sería hasta 1957 cuando, Russell Kirsch, un científico del National Institute of Standards and Technology creó la que se considera la primera imagen digital.

Fue durante la década de 1960 cuando comenzaron los avances en el ámbito del procesamiento, a medida que surgían los primeros grandes ordenadores digitales, motivados por los intereses en la exploración espacial y los avances médicos de la época. Hoy en día, los avances en este ámbito han llegado hasta el punto en que se pueden encontrar en una amplia variedad de campos, como el reconocimiento de objetos, reconocimiento facial y de expresiones, procesado de imagen médica, clasificación de áreas urbanas y agrícolas, generación de mapas…

Los usos más populares se podrían ver en aplicaciones como Facebook, Instagram o Snapchat. Todas ellas utilizan sistemas de detección de caras y facciones en tiempo real [3] para realizar manipulaciones sobre la imagen, como hacer retoques o realizar animaciones sobre ella. Estos sistemas consiguen un gran éxito entre los usuarios debido al entretenimiento que proporcionan, pero también podemos ver un incremento en el interés por estos sistemas por parte de organismos gubernamentales, como es el caso de INTERPOL [4], con el objetivo de identificar a personas de interés para realizar investigaciones.

En cuanto al ámbito médico, los recientes avances en el procesado de imagen son promete-

dores, con cada vez más científicos desarrollando herramientas de diagnóstico automatizado. En los últimos años ya podemos ver sistemas que permiten, por ejemplo, detectar casos de cáncer de mama [5], como es el caso del desarrollado por Google [6] que, con ayuda de la Inteligencia Artificial, promete tener incluso más precisión que los doctores en su diagnóstico.

En este trabajo, la problemática a tratar se sitúa en el ámbito médico. Lo que se busca es desarrollar una herramienta de diagnóstico que realice un procesado digital sobre imágenes médicas, con el objetivo de automatizar la toma de mediciones sobre ellas, ahorrando tiempo de trabajo al personal médico, que actualmente estaría realizando estas medidas de forma manual.

El sistema a desarrollar deberá operar sobre dos imágenes de una rodilla, las cuales se obtienen previamente mediante un TAC, una de ellas correspondiente a la sección del fémur y la otra a la tibia. Con estas dos imágenes se realizará un procesado que calculará la medida correspondiente a la Tuberosidad Tibial Anterior - Garganta de la Tróclea, la cual se explica en la sección 2.2.

1.1 Objetivos

En el presente proyecto se va a desarrollar la herramienta de asistencia médica comentada anteriormente. El sistema final debe cumplir los objetivos que serán descritos a continuación.

El primero de ellos será realizar un procesado digital de imagen en diferentes etapas, el cual será capaz de calcular la medición de la TA-GT de forma precisa en un porcentaje elevado de los casos. Para realizar este procesado, un médico será el encargado de seleccionar las dos imágenes de entrada que considere más adecuadas.

Al final de este procesado, el programa mostrará por pantalla los resultados de las medidas tomadas sobre las dos imágenes de entrada, permitiendo al usuario la modificación de los cálculos de forma manual ajustando los puntos de control que determinan las mediciones.

Por último, sistema resultante deberá proporcionar un medio de interacción con el usuario a través de una interfaz gráfica. Con ella, el personal médico podrá seleccionar las imágenes de entrada, visualizar y modificar los resultados y guardar una copia de estos.

Capítulo 2

Fundamentos

En este segundo capítulo explicaremos algunos conceptos que le serán útiles al lector a la hora de entender el trabajo, como por ejemplo en qué consiste el procesado de imagen, alguno de los algoritmos utilizados o las herramientas utilizadas.

2.1 Procesado Digital de Imagen

La imagen digital [7] es aquella imagen capturada en un medio electrónico y representada en un archivo de información como una serie de impulsos eléctricos. Para este trabajo se emplearán imágenes en escala de grises. Esto implica que, para cada coordenada en el plano, existe un valor de brillo asociado a ese píxel, representado por una función *f(x,y)*.

El procesamiento digital de una imagen [8, 9] consiste en aplicar operaciones que alteren el valor de intensidad en ese mapa de píxeles. Se toma una imagen y se obtiene una versión modificada de la misma, la cual se podrá emplear para la extracción de conocimiento.

Un sistema de procesado contará con una serie de etapas, que podemos ver listadas a continuación:

1. **Captura:** dado que una computadora solo podrá operar con imágenes en formato digital, es necesario un sistema que convierta la imagen capturada por un sensor, expresada como una función continua de dos coordenadas en el plano. Un digitalizador debe ser capaz de dividir la captura en píxeles, asignar un nivel de gris entero para cada uno de ellos y escribir esos datos en un dispositivo de almacenamiento.

2. **Preprocesado:** consiste en el conjunto de técnicas que buscan mejorar la apariencia de la imagen para hacerla más adecuada para un observador o para su posterior análisis digital. Esta etapa nunca aumenta la cantidad de información en la captura.

   No hay un método común de preprocesado, ya que, dependiendo de la imagen que se quiera tratar, las operaciones a realizar variarán. Los objetivos de esta fase son reducir el

ruido y distorsiones no deseadas de la imagen, así como realzar propiedades interesantes de cara a un futuro procesado.

Los métodos de preprocesado se basan en la redundancia existente en las imágenes, ya que los píxeles correspondientes a un objeto suelen tener un valor de gris muy similar entre sí. Esto permite corregir distorsiones sustituyendo un píxel por un valor medio de los que están a su alrededor. Será importante conocer el tipo de degradaciones presentes en las imágenes para poder aplicar un buen método de corrección.

Dependiendo del tipo de operaciones que se apliquen, podemos diferenciar los siguientes tipos de preprocesado:

- **Transformaciones en el nivel de gris:** modifican el nivel de gris original de los píxeles de forma individual. Pueden utilizarse para corregir el nivel de gris de una imagen (errores producidos en la etapa de captura) o modificar la escala de grises, con el objetivo de destacar algunas características de las imágenes.
- **Transformaciones geométricas:** modifican la relación espacial entre los píxeles. Consisten en una transformación que reubica un píxel en función de su posición original en el plano de la imagen, seguida de una interpolación, que asigna un nuevo valor de intensidad a cada píxel en la imagen transformada.
- **Métodos de vecindad local:** combinan el nivel de gris de un píxel con el de sus vecinos, obteniendo así un nuevo valor de intensidad. Este proceso se conoce como filtrado.
- **Operaciones morfológicas:** consisten en operaciones no lineales que actúan en la forma de los objetos. Su funcionamiento se basa en examinar la forma geométrica de un objeto, comparándolo con un patrón predefinido, denominado elemento estructurante. Pueden ser útiles en tareas de eliminación de ruido o simplificación de formas.

3. **Segmentación:** consiste en el reconocimiento de cada uno de los objetos en la captura. El objetivo es dividir la imagen en regiones significativas, que serán utilizadas en las siguientes etapas.

   La segmentación se utiliza, por ejemplo, para extraer regiones de la imagen que corresponden a una misma zona o objeto. Dentro de esta región, la imagen debe ser homogénea, siendo distinta de las regiones adyacentes. Es un problema complejo, ya que no existe una solución única, sino que esta dependerá de la interpretación.

   Habitualmente se basa en características de bajo nivel, como pueden ser el color o la textura, para encontrar objetos. En el caso más simple, cuando un objeto y el fondo

tienen niveles de gris diferentes, se recurre al uso de un umbral para determinar a que región pertenece cada píxel. En caso de tener varios objetos sería necesario determinar diferentes umbrales para detectar cada uno de ellos, siempre y cuando un objeto mantenga el mismo nivel de gris en toda su superficie.

En casos más complejos es necesario recurrir a gradientes de luminosidad, ya que los bordes de cada región suelen estar asociados a zonas con un elevado contraste. Existen diferentes algoritmos basados en esta propiedad utilizados para resaltar estas zonas con un alto gradiente de luminosidad, conocidos como detectores de contornos.

4. **Descripción:** en esta etapa se toman los datos obtenidos durante la segmentación y se trata de convertirlos a una forma adecuada para su procesamiento por computadora.

   La salida de la segmentación se compone de píxeles en bruto que componen toda una región o bien su borde. A partir de estos datos se deben extraer los rasgos que serán útiles de cara al procesado. En algunos casos esos rasgos estarán en la propia forma del borde del objeto y en otros serán de interés propiedades como el color, la textura, la estructura o otras propiedades internas.

5. **Reconocimiento e Interpretación:** el reconocimiento consistirá en asignar una etiqueta a un objeto basándose en la información obtenida en las etapas anteriores, mientras que la interpretación tratará de dar un significado al conjunto de objetos reconocidos.

   En este paso es necesario disponer de algún tipo de conocimiento de la imagen, que estará codificado en el sistema de procesamiento. Este conocimiento puede ser tan simple como las coordenadas de la zona de la imagen en la que se sabe que se dispone de información de interés, siendo posible aumentar su complejidad en función del problema que se quiera resolver.

En la figura 2.1 se puede ver un esquema de los diferentes pasos que componen un sistema de procesado de imagen.

Como podemos apreciar, la base de conocimiento puede interactuar en cualquiera de las etapas para realizar alguna tarea de reconocimiento e interpretación.

No obstante, aunque este sea el flujo típico de este tipo de aplicaciones, en algunos casos se podría prescindir de algunos de estos módulos. Esto variará en función de los requisitos que se deban cumplir, ya que, por ejemplo, un sistema que se limita al realce de detalles no necesitaría ir más allá del preprocesado.

2.1.1 Propiedades Métricas y Topológicas

Una imagen digital se representa mediante una matriz bidimensional, cuyos elementos se corresponden a un nivel de brillo, de la cual se pueden extraer ciertas propiedades que no se corresponden a ninguna función continua.

2.1.1.1 Vecindad

Un píxel *(x,y)* tiene cuatro vecinos horizontales y verticales, ubicados en las coordenadas: *(x+1,y), (x-1,y), (x,y+1), (x,y-1)*. Este mismo píxel *(x,y)* tiene también cuatro vecinos diagonales con coordenadas: *(x+1,y+1), (x+1,y-1), (x-1,y+1), (x-1,y-1)*.

Se conocen como 4-vecinos de un píxel el conjunto de sus vecinos horizontales y verticales, y como 8-vecinos el conjunto formado por los vecinos horizontales, verticales y diagonales.

En la imagen 2.2 se pueden ver el conjunto de 4-vecinos y 8-vecinos de un píxel.

2.1.1.2 Conectividad

A partir del concepto de píxeles vecinos se desarrolla el de conectividad, el cual facilitará más adelante la tarea de establecer los límites de los objetos presentes en la imagen.

Se dice que dos píxeles están 4-conectados cuando uno pertenece al conjunto de 4-vecinos del otro. De forma análoga, dos píxeles estarán 8-conectados si uno se encuentra entre los 8-vecinos del otro.

2.1.1.3 Camino Digital

Dada una imagen con una relación de vecindad definida, un camino digital entre dos píxeles sería una sucesión de píxeles del mismo color en la que todos los componentes del camino tienen exactamente dos vecinos. A continuación, en la figura 2.3 vemos un ejemplo con píxeles 4-conectados y otro con píxeles 8-conectados. Los píxeles en amarillo se corresponden al camino entre los dos píxeles de color azul.

2.1.1.4 Componente Conexa

Una componente conexa digital es aquel conjunto de píxeles tal que, para cualquier perteneciente al conjunto, existe un camino digital que los une. En este caso, diremos que ambos píxeles están conectados.

Además, una componente conexa acotada será aquella en la que ninguno de los píxeles está en el borde de la imagen.

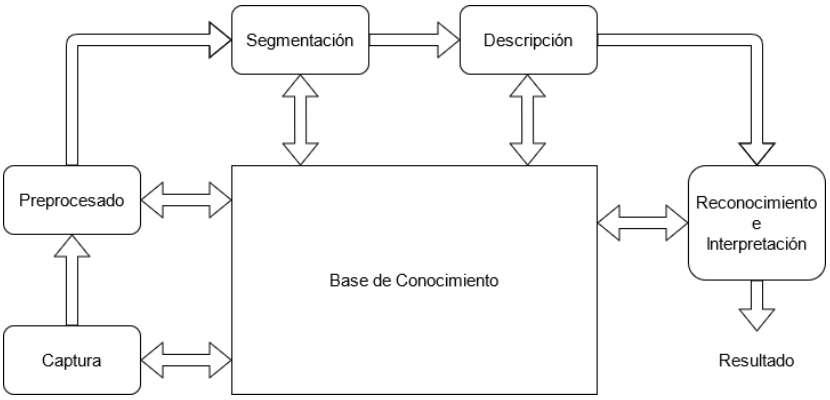


Figura 2.1: Etapas del procesado digital de imagen

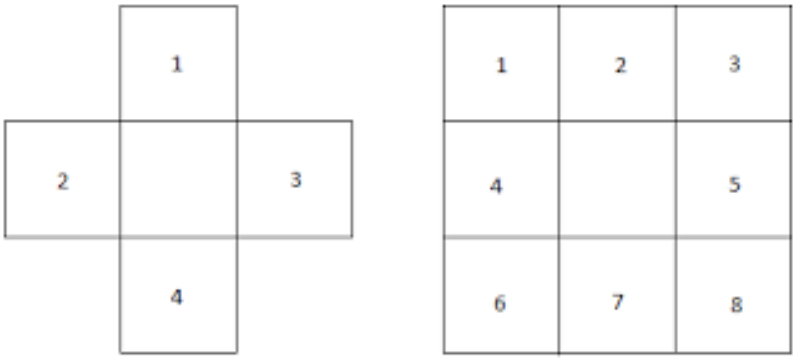


Figura 2.2: 4-vecinos y 8-vecinos de un píxel

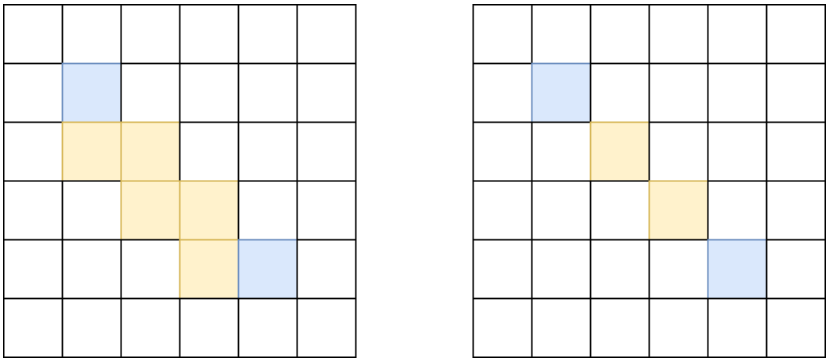


Figura 2.3: Caminos digitales 4-conectados y 8-conectados

2.1.2 Filtrado

Los métodos de filtrado digital son uno de los principales modos de operar en el procesamiento de imágenes. Su funcionamiento se basa en combinar el nivel de gris de cada píxel con los valores del vecindario en la imagen original. De esta forma obtienen un nuevo valor para la imagen transformada. Entre los usos habituales de los filtros estarán el suavizado, la eliminación de ruido y el realce de bordes.

Una imagen se puede filtrar en el dominio del espacio (trabajando directamente sobre los píxeles de la imagen) o en el dominio de la frecuencia (operando sobre la Transformada de Fourier de la imagen). En este trabajo nos centraremos solamente en el primer tipo, ya que es el más habitual y será el que utilicemos en el sistema a implementar.

2.1.2.1 Ruido

El ruido consiste en algún error (información no deseada) que degrada la calidad de una imagen. Su origen puede estar en los procesos de adquisición, transmisión o procesado.

Existen diferentes modelos matemáticos usados para describir un ruido. A continuación tenemos algunos de los más habituales:

- **Ruido Gaussiano:** es el ruido producido por los circuitos electrónicos o el generado por los sensores debido a la falta de iluminación o altas temperaturas. Además, este es el modelo utilizado de forma genérica en ausencia de información. En la figura 2.4 tenemos un ejemplo de este tipo de distorsión.

Figura 2.4: Ruido Gaussiano

- **Ruido Impulsional:** también conocido como ”sal y pimienta” (por provocar la aparición de píxeles blancos o negros en la imagen) es un ruido producido por un mal funcionamiento de los sensores de captura o por errores durante el proceso de digitalización.

Produce niveles de intensidad muy diferentes al resto de píxeles en el vecindario. Se puede ver un ejemplo de este tipo de ruido en la figura 2.5.

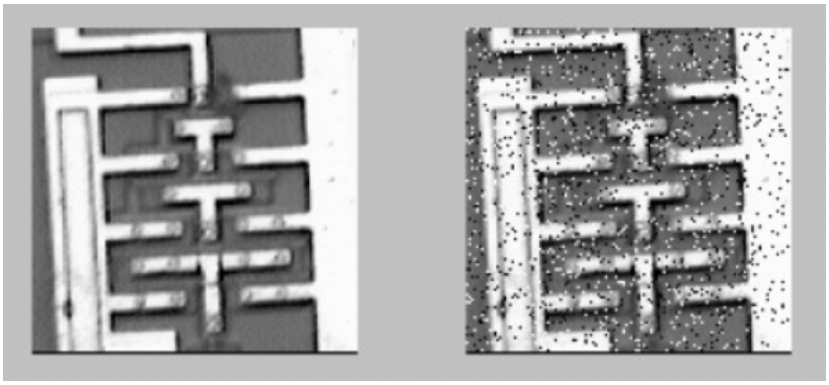


Figura 2.5: Ruido Sal y Pimienta

2.1.2.2 Filtros espaciales lineales

El elemento básico de los filtros espaciales es la máscara o kernel. Este elemento consiste en un vecindario de tamaño *mxn* centrado en un píxel *(i,j)*, con un coeficiente asignado para cada píxel del vecindario. Este elemento se irá desplazando por todos los píxeles de la imagen original, calculando un nuevo valor para la imagen transformada. Este proceso se conoce como convolución.

Dada una imagen f de tamaño *MxN* y una máscara g de tamaño *mxn*, el resultado de realizar una convolución se corresponderá con la siguiente fórmula.

$$\sum_{s=-a}^{a}\sum_{t=-b}^{b} f(i+s, j+t) * g(s,t),$$

$$i = a..M-a, j = b..N-b,$$

$$a = (m-1)/2, b = (n-1)/2$$

El valor de cada píxel en la imagen resultante será la suma de los valores de los píxeles de su vecindario multiplicados por el coeficiente que indica el kernel.

En la imagen 2.6 se puede ver una representación de como la máscara opera en cada píxel.

A continuación vemos algunos ejemplos de filtros espaciales lineales:

- **Filtro de Medias:** es el filtro más intuitivo y fácil de implementar, usado para quitar ruido y suavizar pequeños detalles en la imagen. El valor del nuevo píxel será la me-

dia del vecindario. Se puede operar mediante convolución con una máscara como la siguiente (ejemplo con tamaño *3x3*):

$$\frac{1}{9}\begin{pmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{pmatrix}$$

Este filtro tiene la desventaja de ser bastante sensible a cambios locales. Atenúa el ruido de tipo gaussiano, pero en el caso del ruido impulsional simplemente lo difumina.

En la figura 2.7 se puede ver el resultado de aplicar este filtro con una mascara de tamaño *3x3* (2.7b) y otra de tamaño *7x7* (2.7c).

- **Filtro Gaussiano:** es un filro usado para difuminar imágenes y eliminar ruido. Es similar al filtro de medias, pero en este caso se utiliza una función gaussiana para determinar los coeficientes del kernel:

$$G(x,y) = \frac{1}{2\pi\sigma^2}e^{\frac{-(x^2+y^2)}{2\sigma^2}}$$

El grado de suavizado es controlado por sigma, cuyo valor se escoge de forma arbitraria.

Algunas de sus ventajas respecto al filtro de medias serían que se puede separar en dos filtros unidimensionales, pudiendo aplicarlo dos veces, una de forma horizontal y otra de forma vertical, siendo así más eficiente. Además, produce un suavizado más uniforme, menos brusco y con mejor preservación de bordes. Esto se debe a que el peso de los píxeles decrece a medida que aumenta la distancia al centro.

A continuación vemos un ejemplo de máscara gaussiana con tamaño *5x5* y σ=1:

$$\begin{pmatrix} g(-2,2) & g(-1,2) & g(0,2) & g(1,2) & g(2,2)\\ g(-2,1) & g(-1,1) & g(0,1) & g(1,1) & g(2,1)\\ g(-2,0) & g(-1,0) & g(0,0) & g(1,0) & g(2,0)\\ g(-2,-1) & g(-1,-1) & g(0,-1) & g(1,-1) & g(2,-1)\\ g(-2,-2) & g(-1,-2) & g(0,-2) & g(1,-2) & g(2,-2)\end{pmatrix}$$

$$\frac{1}{273}\begin{pmatrix}1 & 4 & 7 & 4 & 1\\ 4 & 16 & 26 & 16 & 4\\ 7 & 26 & 41 & 26 & 7\\ 4 & 16 & 26 & 16 & 4\\ 1 & 4 & 7 & 4 & 1\end{pmatrix}$$

En la figura 2.8 vemos un par de ejemplos de los resultados de aplicar este filtro para $\sigma = 1$ (2.8b) y para $\sigma = 3$ (2.8c).

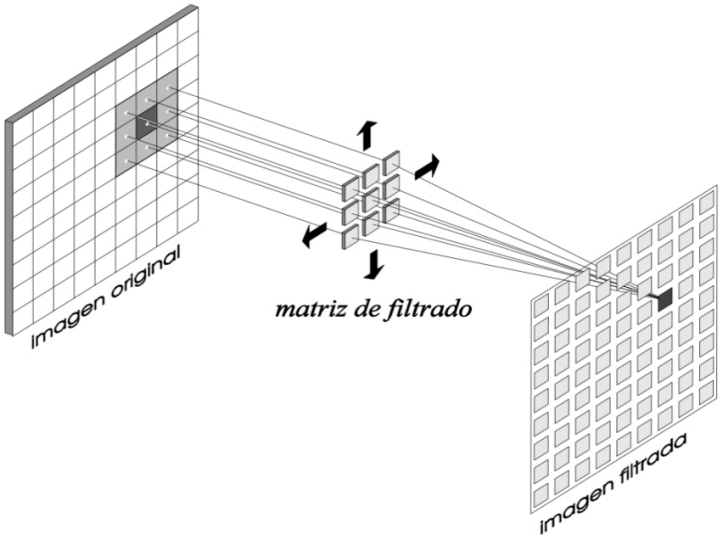


Figura 2.6: Filtrado mediante una máscara de convolución

(a) Imagen original (b) Máscara 3x3 (c) Máscara 7x7

Figura 2.7: Resultado al aplicar el filtro de medias

(a) Imagen original (b) $\sigma = 1$ (c) $\sigma = 3$

Figura 2.8: Resultado al aplicar el filtro gaussiano

2.1.2.3 Filtros de orden

Este tipo de filtros se diferencia de los lineales en que funcionan ordenando los valores en la vecindad de cada punto de menor a mayor, sacando el resultado a partir de esa lista ordenada, en lugar de realizar una operación sobre el conjunto de píxeles.

Pertenecen a este tipo el filtro de máximos (devuelve el valor máximo del vecindario), el filtro de mínimos (devuelve el valor mínimo del vecindario) y el filtro de la mediana (devuelve el valor en posición intermedia). De estas tres posibilidades, el más polivalente es el de la mediana, ya que elimina los dos tipos de ruido impulsional sin aclarar ni oscurecer la imagen y preservando los bordes. Tiene desventajas cuando se encuentra lineas muy finas, ya que puede que las elimine de la imagen.

Vemos un ejemplo de su funcionamiento en la figura 2.9.

(a) Imagen original (b) Imagen filtrada

Figura 2.9: Resultado al aplicar el filtro de la mediana

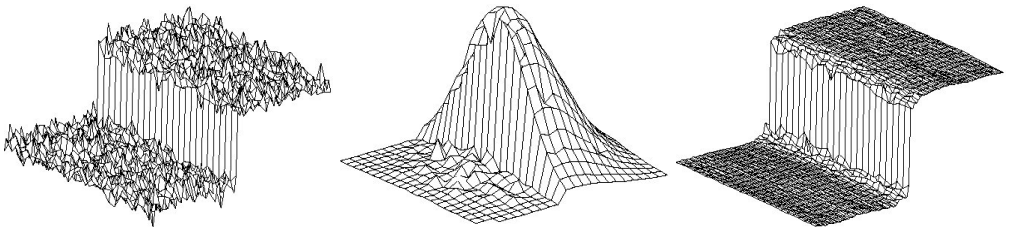
2.1.2.4 Filtro bilateral

El filtro bilateral [10] pertenece al conjunto de los filtros no lineales. Es un filtro de reducción de ruido y suavizado de imágenes con preservación de bordes. Es un filtro un poco más avanzado y, por lo tanto, más complejo que los citados anteriormente.

Su funcionamiento consiste en que el valor de intensidad de cada píxel es reemplazado por una media ponderada de los píxeles cercanos. El valor de los pesos se puede basar en una distribución de Gauss, pero también depende de diferencias radiométricas, como puede ser la intensidad de color, lo cual permite preservar los bordes. El filtro operará en base a un tamaño de vecindario y un valor sigma indicados por el usuario.

Este filtro combina el filtrado en el espacio y en el rango de intensidad (de ahí el nombre ”bilateral”), utilizando una función de similitud para los píxeles cercanos. El valor de un píxel en una posición *(x,y)* es reemplazado por un valor promedio de los píxeles en su vecindario y con una intensidad similar. En regiones suaves corregirá las pequeñas diferencias causadas por el ruido, mientras que en regiones con cambios bruscos de intensidad mantendrá la variación.

En la figura 2.10 se aprecia de forma más clara el comportamiento de esta función de similitud. En este ejemplo se ven los valores de intensidad en el borde de un objeto, con una discontinuidad pronunciada, en la imagen 2.10a.



(a) (b) (c)

Figura 2.10: Funcionamiento del filtro bilateral

La función de similitud aplicada en la imagen 2.10b está centrada dos píxeles a la derecha del borde, con una dimensión de *23x23*. Se puede apreciar que los píxeles cercanos pertenecientes al lado derecho presentan una alta similitud, mientras que los píxeles en el lado izquierdo del borde tendrían una similitud cercana a cero. El resultado del filtrado bilateral en ese píxel sería el valor medio de los píxeles con un valor de similitud elevado, ignorando los valores al otro lado del borde. De esta forma y, como se muestra en la imagen 2.10c, se consigue un buen comportamiento del filtrado mientras que se conserva el detalle en los contornos.

En la imagen 2.11 se puede apreciar el funcionamiento en un caso real del filtrado bilateral, con una dimensión del vecindario de *23x23* y un $\sigma = 120$.

(a) Imagen original (b) Imagen filtrada

Figura 2.11: Resultado al aplicar el filtro bilateral

2.1.3 Detección de Contornos

La detección de contornos es una etapa de interés y gran practicidad en el procesado de imagen, debido a que su localización puede permitir evidenciar la presencia de objetos, que posteriormente podrán ser identificados.

Un borde se corresponde con un cambio pronunciado en el nivel de intensidad en una imagen. El contorno de un objeto se compone de una serie de píxeles conectados que delimitan dos regiones disjuntas.

Disponer de una imagen de bordes supone un nivel superior de abstracción en el procesado de una imagen, ya que concentra la cantidad de información a procesar, conservando las propiedades estructurales de la imagen.

La detección de bordes es posible utilizando conceptos análogos al cálculo de derivadas por el método de diferencias finitas. Una derivada en el dominio discreto de una imagen sería el resultado de la diferencia entre la intensidad de dos píxeles contiguos.

El resultado de calcular la primera derivada de una imagen tendría en su resultado la siguiente información:

- **Derivada nula.** En regiones con el mismo valor de intensidad para píxeles vecinos, la diferencia entre ellos será nula, por lo tanto también lo será la primera derivada.
- **Derivada no nula.** Indica que se está produciendo algún cambio en el valor de intensidad de los píxeles. Si su valor es constante, el cambio en el nivel de gris sería gradual, mientras que un incremento o decremento pronunciado indicaría un cambio brusco.

La segunda derivada permite obtener información más detallada acerca de la imagen:

- **Derivada nula.** Se corresponde a zonas con niveles de gris constante o zonas con un cambio gradual en el nivel de intensidad (zonas con la primera derivada constante no nula).
- **Derivada positiva.** Indica la transición a una zona con un nivel de gris mayor.
- **Derivada negativa.** Indica el paso a un nivel de intensidad menor.

En la figura 2.12 se puede ver un ejemplo del cálculo de la primera (2.12c) y segunda (2.12d) derivada para un borde gradual (2.12b). En ella se aprecia que la función de la segunda derivada produce bordes dobles.

En la práctica, en procesado de imágenes es necesario discretizar las funciones, para lo que se realiza una convolución con máscaras direccionales (la derivación se realiza por separado en ambos ejes). El cálculo de derivadas mediante diferencias finitas se podría hacer con las siguientes máscaras:

$$M_x = \begin{pmatrix} -1 & 1 \\ 0 & 0 \end{pmatrix}, \quad M_y = \begin{pmatrix} -1 & 0 \\ 1 & 0 \end{pmatrix}$$

A partir de estas dos máscaras se obtendría una imagen de gradientes horizontal (G_x) y otra vertical (G_y). A partir de ellas se podrá calcular la magnitud y la orientación del borde con las siguientes fórmulas:

- **Magnitud.**

$$|\bigtriangledown f| = \sqrt{G_x^2 + G_y^2}$$

- **Orientación.**

$$\phi = arctan(G_x/G_y)$$

Estas máscaras serían el ejemplo más simple de cálculo de derivadas. En aplicaciones reales se utilizan métodos más complejos, conocidos como operadores de primera y segunda derivada. A continuación se explica el funcionamiento de algunos de ellos.

2.1.3.1 Operador de Prewitt

Es un operador de primera derivada basado en la idea de diferencias centrales. Se utiliza un promediado (se calcula en tres lineas de píxeles) para reducir su sensibilidad al ruido.

Se define mediante las siguientes máscaras:

$$M_x = \begin{pmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{pmatrix}, \quad M_y = \begin{pmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{pmatrix}$$

Es un operador simple y eficiente, con buenos resultados para bordes verticales y horizontales. Es uno de los mejores métodos a la hora de detectar la orientación de las imágenes. Como punto negativo estaría que los contornos diagonales no siempre se detectan con precisión.

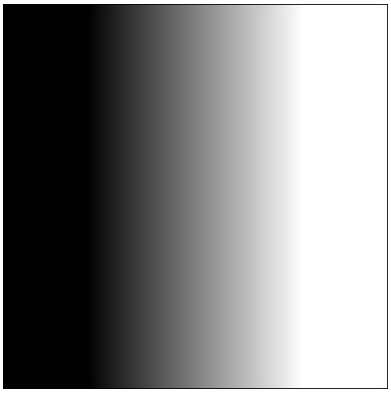
2.1.3.2 Operador de Sobel

Similar al operador de Prewitt, pero con mayor ponderación en los píxeles centrales.

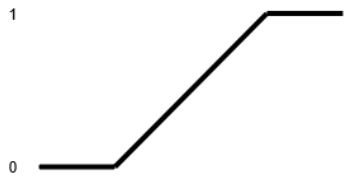
$$M_x = \begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix}, \quad M_y = \begin{pmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

Es eficiente y obtiene buenos resultados en la detección de bordes finos. Es más sensible al ruido que el operador de Prewitt, manteniendo los problemas con algunos contornos diagonales.

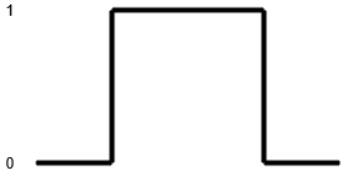
Las imágenes de la figura 2.13 muestran los resultados de aplicar este operador.



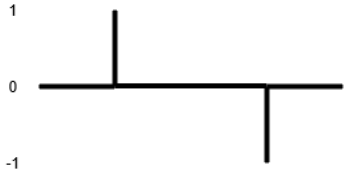
(a) Imagen de contorno



(b) Función de nivel de gris

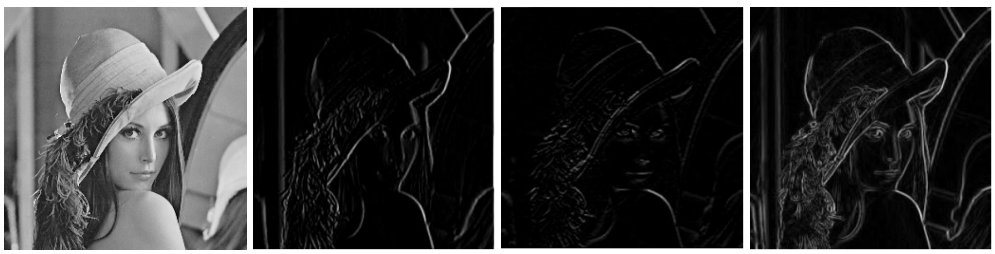


(c) Primera derivada



(d) Segunda derivada

Figura 2.12: Ejemplo del cálculo de derivadas



(a) Imagen original (b) G_x (c) G_y (d) Resultado

Figura 2.13: Ejemplo del operador de Sobel

2.1.3.3 Operador de Roberts

Este operador se implementa calculando las máscaras con las diagonales correspondientes al píxel de interés:

$$M_x = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \quad M_y = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

Este kernel está diseñado para responder con precisión ante bordes diagonales, siendo también muy eficiente. Su desventaja es que al ser un operador tan simple es altamente sensible al ruido.

2.1.3.4 Operadores de Compás

Consisten en introducir un conjunto de *n* máscaras que representan *n* rotaciones espaciales en la imagen. Se busca tener la máxima precisión en la detección de bordes.

Un ejemplo de este tipo de operadores sería el de Kirsch, que tendría el siguiente conjunto de máscaras para cada una de las 8 rotaciones espaciales principales:

$$M_0 = \begin{pmatrix} -3 & -3 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & 5 \end{pmatrix}, \quad M_{\pi/4} = \begin{pmatrix} -3 & 5 & 5 \\ -3 & 0 & 5 \\ -3 & -3 & -3 \end{pmatrix}, \quad M_{\pi/2} = \begin{pmatrix} 5 & 5 & 5 \\ -3 & 0 & -3 \\ -3 & -3 & -3 \end{pmatrix},$$

$$M_{3\pi/4} = \begin{pmatrix} 5 & 5 & -3 \\ 5 & 0 & -3 \\ -3 & -3 & -3 \end{pmatrix}, \quad M_{\pi} = \begin{pmatrix} 5 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & -3 & -3 \end{pmatrix}, \quad M_{5\pi/4} = \begin{pmatrix} -3 & -3 & -3 \\ 5 & 0 & -3 \\ 5 & 5 & -3 \end{pmatrix},$$

$$M_{3\pi/2} = \begin{pmatrix} -3 & -3 & -3 \\ -3 & 0 & -3 \\ 5 & 5 & 5 \end{pmatrix}, \quad M_{7\pi/4} = \begin{pmatrix} -3 & -3 & -3 \\ -3 & 0 & 5 \\ -3 & 5 & 5 \end{pmatrix}$$

El punto negativo de este tipo de operadores es su coste computacional, ya que aplican 8 máscaras de convolución en lugar de 2.

2.1.3.4 Laplaciana de Gaussiana (LoG)

Es un operador basado en una distribución Gaussiana que utiliza la segunda derivada de la imagen obtenida a partir del Laplaciano, definido por la siguiente fórmula:

$$\bigtriangledown^2 f(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2} + \frac{\partial^2 f(x,y)}{\partial y^2}$$

Teniendo en cuenta que la distribución Gaussiana en dos dimensiones se define de la siguiente manera:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{\frac{-(x^2+y^2)}{2\sigma^2}}$$

El operador de la Laplaciana de Gaussiana se definiría multiplicando el valor de la segunda derivada por la función Gaussiana:

$$\bigtriangledown^2 G(x,y) = \frac{\partial^2 f(x,y)}{\partial x^2} G(x,y) + \frac{\partial^2 f(x,y)}{\partial y^2} G(x,y) = \frac{x^2+y^2+2\sigma^2}{\sigma^4} e^{\frac{-(x^2+y^2)}{2\sigma^2}}$$

Esta fórmula permitiría calcular una máscara de convolución de un tamaño *mxn*, como podría ser la siguiente *5x5*:

$$\begin{pmatrix} 0 & 0 & -1 & 0 & 0 \\ 0 & -1 & -2 & -1 & 0 \\ -1 & -2 & 16 & -2 & -1 \\ 0 & -1 & -2 & -1 & 0 \\ 0 & 0 & -1 & 0 & 0 \end{pmatrix}$$

Este operador permite detectar bordes de forma eficiente en todas las direcciones, pero tiene la desventaja de ser sensible al ruido, pudiendo generar falsos contornos.

2.1.3.5 Algoritmo de Canny

El algoritmo de Canny [11, 12] es un operador de detección de bordes avanzado, basado en un algoritmo en múltiples fases con capacidad para detectar diferentes tipos de contornos. Es el operador más utilizado para la detección de bordes en sistemas de procesado de imagen, ya que su precisión es muy alta, con la desventaja de tener un elevado coste computacional.

Su aplicación se divide en los siguientes pasos:

1. **Filtrado.** En el primer paso se busca eliminar el posible ruido que pueda afectar a la detección de bordes mediante un suavizado gaussiano.

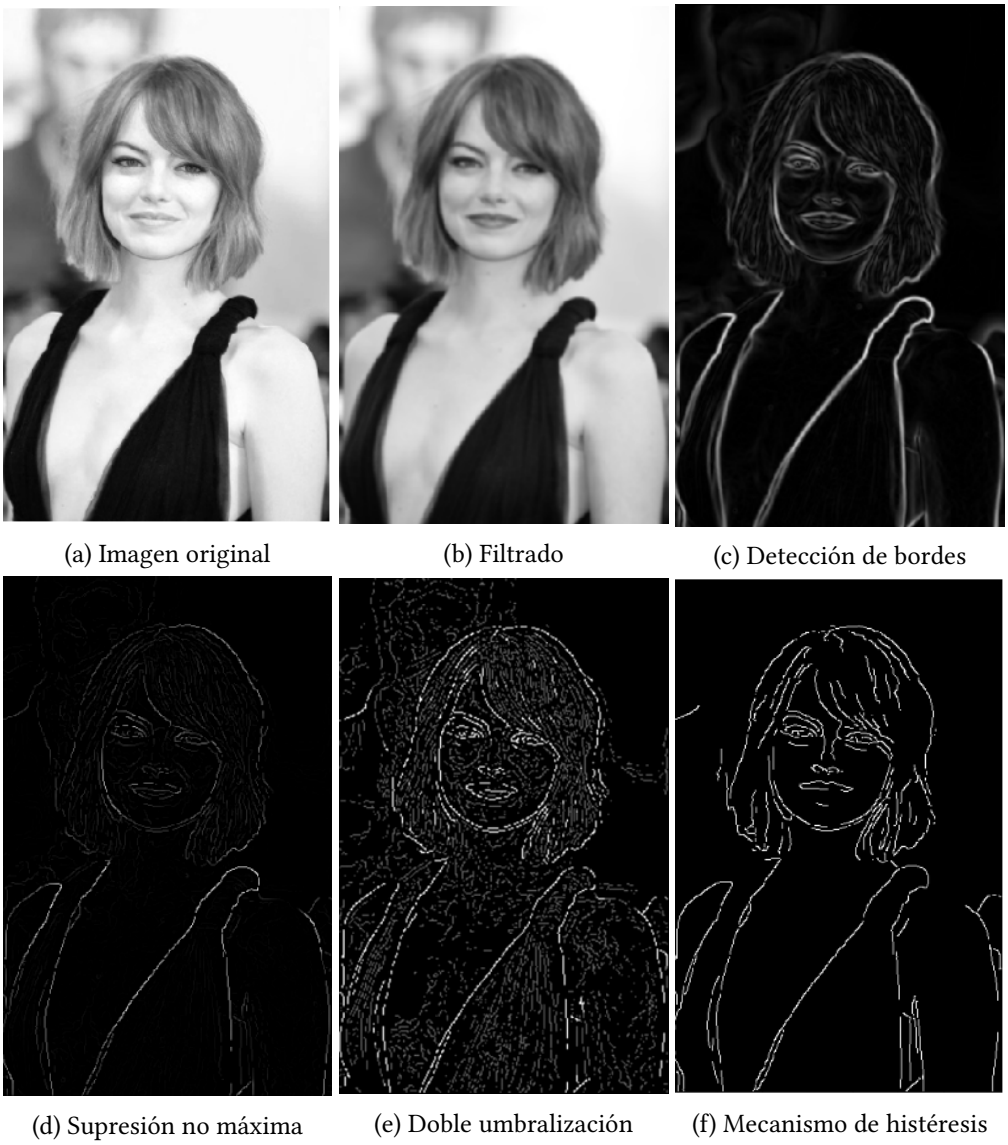
2. **Detección de bordes.** A la imagen suavizada se le aplica el operador de Sobel para realizar el cálculo de gradientes, que permitirán obtener la magnitud y la orientación del borde en cada píxel.

3. **Supresión no máxima.** Lo ideal es que la imagen final tenga bordes finos, por lo que se aplica una supresión para reducirlos a un píxel de grosor. En algoritmo busca los puntos de mayor intensidad en la dirección del borde y elimina el resto.

4. **Umbralización con histéresis.** Se aplican dos umbrales, alto y bajo, para identificar tres tipos de píxeles: fuertes, débiles y no relevantes. Los píxeles fuertes superan el umbral alto y son considerados bordes siempre, al contrario que los no relevantes, que

no alcanzan el umbral bajo y son desestimados. Los píxeles débiles están entre los dos umbrales y se considerarán bordes cuando están conectados a algún píxel fuerte, lo que se conoce como mecanismo de histéresis. Al final de este proceso se descartan todos los píxeles que no fueron considerados como bordes.

En las imágenes de la figura 2.14 se pueden ver las etapas del proceso descrito.



(a) Imagen original (b) Filtrado (c) Detección de bordes

(d) Supresión no máxima (e) Doble umbralización (f) Mecanismo de histéresis

Figura 2.14: Etapas del algoritmo de Canny

2.2 Fundamentos Médicos

En esta sección veremos el tipo de imagen que se procesará en este trabajo y las medidas que se tomarán sobre esas imágenes, ambos conceptos necesarios para entender el sistema implementado.

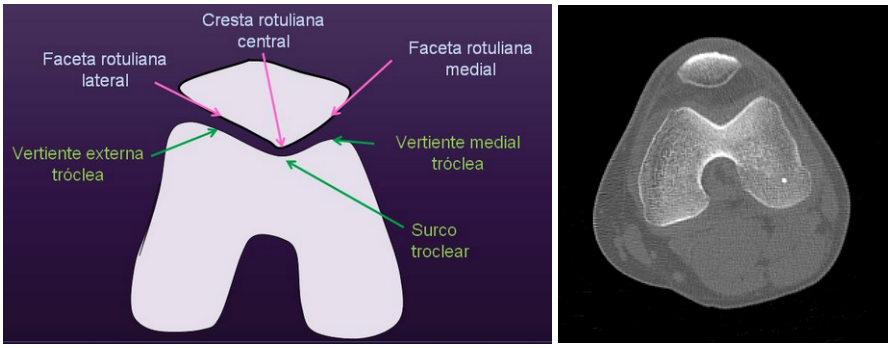
2.2.1 Imágenes de entrada

Las mediciones de rodilla se realizan sobre dos imágenes obtenidas mediante Tomografía Axial Computerizada (TAC) [13]. Para tomar las capturas, el paciente estará colocado en posición de decúbito supino, con una rotación externa de los pies de 15º y cuádriceps relajados.

Durante la realización del TAC se extraerán dos grupos de imágenes.

2.2.1.1 Garganta de la tróclea

El primero de ellos irá desde la porción media de la rótula incluyendo cortes de los cóndilos femorales. En este grupo se podrá apreciar la Garganta de la Tróclea (GT), también llamada báscula rotuliana, región situada sobre el fémur sobre la cual reposa la rótula.



(a) Representación (b) Imagen de un TAC

Figura 2.15: Báscula rotuliana

La imagen escogida entre las del grupo para tomar las medidas será la que mejor muestre el surco intercondileo en ”arco romano” (zona inferior del fémur en la imagen 2.15a).

En la imagen 2.15b se puede ver un ejemplo real de una captura de este grupo.

2.2.1.2 Tuberosidad anterior de la tibia

El segundo grupo será un corte desde la epífasis tibial hasta alcanzar la porción más elevada de la Tuberosidad Anterior de la Tibia (TAT), una prominencia ósea con forma triangular situada en la parte superior de la tibia, justo por debajo de la rodilla. En la figura 2.16 se puede ver un ejemplo obtenido en un TAC.

2.2.2 Medidas realizadas

Las medidas tomadas sobre la imagen se conocen como Tuberosidad Tibial Anterior - Garganta de la Tróclea (TA-GT). Para ello se escogerá la imagen de la báscula rotuliana que mejor muestre el surco intercondileo y la imagen de la tibia que muestre el punto más saliente de la TAT.

Las dos imágenes son superpuestas y sobre ellas se trazarán tres líneas:

1. Una línea horizontal tangencial al margen posterior de los cóndilos del fémur.
2. Una línea vertical perpendicular a la primera que pasará por la parte más profunda del surco troclear.
3. Una línea vertical perpendicular a la primera que va al margen más saliente de la TAT.

En la imagen 2.17a se pueden ver las lineas trazadas en los dos cortes por separado y en la imagen 2.17b esas mismas lineas están dibujadas sobre las dos capturas superpuestas.

La medición se corresponde a la distancia que separa las dos líneas verticales, siendo su valor normal de 11-12 mm, mientras que un valor superior a 15 mm se considera patológico.

2.3 DICOM

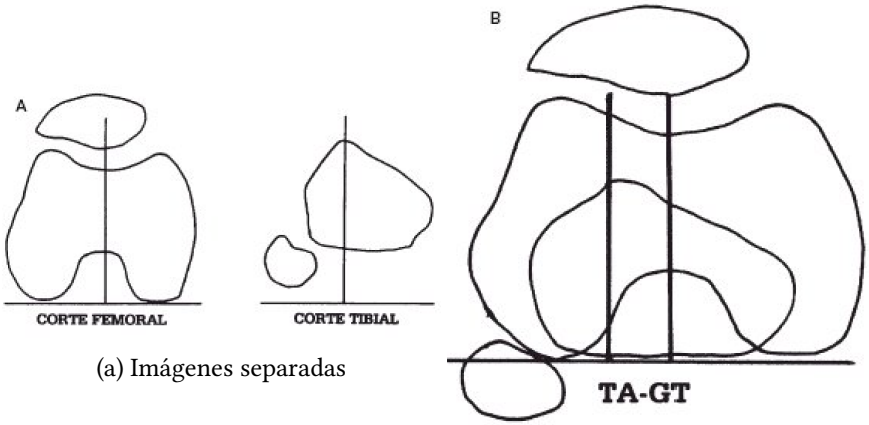
A continuación se explicará el estándar internacional utilizado para imágenes médicas e información relacionada, Digital Imaging and Communications in Medicine (DICOM) [14]. Las aplicaciones más comunes son la visualización, almacenamiento, impresión y transmisión de este tipo de imágenes.

Actualmente, DICOM está implementado en casi cualquier equipo de captación de imagen médica (Rayos-X, ultrasonidos, etc.). Desde su primera versión en 1993 ha revolucionado los sistemas existentes, permitiendo dar el paso a una metodología de trabajo completamente digital.

2.3.1 Historia

Su nacimiento se debe a la aparición de dispositivos de Tomografía Computerizada y Resonancia Magnética en las décadas de 1970 y 1980. Al principio resultaba muy complicado trabajar con las imágenes generadas por este tipo de dispositivos, por lo que la American College of Radiology (ACR) y la National Electrical Manufacturers Association (NEMA) se asociaron para intentar solventar este problema. En 1985 se liberó el primer estándar que trataba la comunicación de imagen extremo a extremo, *ACR-NEMA 300*. En 1993 el estándar se cambia para trabajar en redes de área local sobre el protocolo TCP/IP, cambiando su nombre

Figura 2.16: Tuberosidad Anterior de la Tibia



(a) Imágenes separadas

(b) Imágenes superpuestas

Figura 2.17: Medidas TA-GT

por el de DICOM. A lo largo de los años, el número de áreas médicas soportadas, así como los servicios proporcionados, ha ido en aumento.

2.3.2 Objetivos

El objetivo del estándar es el de facilitar la operabilidad entre diferente material médico especificando:

- Una serie de protocolos de red para la comunicación entre dispositivos.
- La semántica y sintaxis de comandos e información asociada que podrá ser intercambiada mediante estos protocolos.
- Un conjunto de servicios de almacenamiento que deberán ser implementados por los dispositivos que lo utilicen, así como un formato de ficheros y una estructura de directorios que facilitan el acceso a los datos almacenados y el intercambio de datos.
- Información que debe ser proporcionada por una implementación que se quiera adecuar al estándar.

Lo que el estándar no especifica son detalles de implementación o funciones esperadas por un sistema que lo cumpla. Tampoco se especifica un procedimiento concreto para la validación y el testeo de los sistemas.

2.3.3 Formato de ficheros.

La información es agrupada en conjuntos de datos. Los archivos conformes a este estándar guardan el número de identificación del paciente junto a la imagen, de forma que no puedan ser separados por error.

Un objeto DICOM consiste en un conjunto de atributos que contienen la información del paciente, acompañados de los datos de imagen. El objeto puede contener una sola imagen, pero esta puede estar compuesta de diferentes fotogramas.

2.4 Java

Java [15] es uno de los lenguajes de programación más utilizados por los desarrolladores de software en todo el mundo. Fue desarrollado en la década de 1990 por *Sun Microsystems* como un lenguaje de programación orientado a objetos, funcionando en diferentes plataformas. De acuerdo con Oracle, Java es usado actualmente en más de tres mil millones de teléfonos móviles.

2.4.1 Historia

James Gosling comenzó el proyecto de Java en el año 1991 [16] con el propósito de desarrollar una máquina virtual y un lenguaje con la ya conocida notación del lenguaje C, pero con mayor precisión y simplicidad.

Los objetivos de este nuevo lenguaje eran: soportar orientación a objetos, soportar la ejecución en múltiples sistemas operativos, disponer de soporte de redes, poder ejecutar código de fuentes remotas de forma segura y ser sencillo de utilizar.

La primera versión pública fue liberada en el año 1995, con el compromiso de ser capaz de ejecutar un mismo código en diferentes plataformas.

2.5 ImageJ

ImageJ [17] es un programa open source escrito en Java desarrollado por el National Institutes of Health (NIH). Funciona en cualquier máquina virtual de Java con versión *1.8* o superior. Tiene una amplia base de usuarios que aportan plugins y macros.

Su principal desarrollador es Wayne Rasband, quien creó una primera versión de un programa de edición de imagen escrita en Pascal en la década de 1970. No fue hasta 1997 cuando comenzó a trabajar en ImageJ, en un momento en el que Java empezaba a ganar popularidad. Rasband escogió este lenguaje influenciado por la idea de poder ejecutar el programa en cualquier máquina.

ImageJ permite mostrar, editar, analizar, procesar, guardar e imprimir imágenes en diferentes formatos (entre ellos DICOM). Soporta *pilas* de imágenes y proporciona diferentes operaciones de procesado de imagen, como puede ser la aplicación de diferentes filtros, máscaras de convolución o algoritmos de detección de bordes.

Capítulo 3

Metodología

En este tercer capítulo explicaremos la metodología utilizada durante el desarrollo del sistema, haciendo un análisis del plan de trabajo escogido.

A lo largo de este proyecto se ha trabajado siguiendo una metodología de desarrollo iterativo e incremental, la cual divide el trabajo en bloques temporales que conocemos como iteraciones.

Estas iteraciones se pueden entender como miniproyectos, ya que proporcionan un programa funcional que representa una parte del producto final, de manera que las diversas funcionalidades se van incluyendo de una manera incremental hasta lograr un sistema completo.

En cada iteración se va evolucionando el sistema partiendo de los resultados de las anteriores iteraciones, añadiendo nuevos requisitos. Uno de los aspectos a tener en cuenta para decidir la planificación de requisitos será priorizar aquellos que sean de mayor relevancia para el funcionamiento final del sistema.

El proyecto contará con una fase previa de análisis, a partir de la cual se podrán definir los requisitos que deberá tener el sistema final, la prioridad de estos y las iteraciones necesarias para implementarlos.

En las primeras iteraciones se buscará realizar las tareas básicas que permitan efectuar el procesado de imagen requerido, mientras que en las tareas finales se refinarán aspectos como la interacción con el sistema o la persistencia de los cálculos.

Para el proyecto que se va a tratar, se procurará que las iteraciones definidas no superen las dos semanas de trabajo, teniendo en cuenta que este será realizado por una sola persona. De esta manera se podrá realizar una revisión periódica de las funcionalidades implementadas, permitiendo gestionar posibles modificaciones de manera regular con cambios a corto plazo.

3.1 Requisitos

A continuación se hará un pequeño análisis de los requisitos identificados para este sistema, que serían los siguientes:

1. **Interacción gráfica.** Se debe proporcionar una interfaz gráfica que permite que el usuario interaccione con el sistema para realizar la selección de imágenes de entrada y para guardar los resultados una vez realizada la medición.

2. **Carga de imágenes.** El sistema debe ser capaz de cargar imágenes en formato DICOM y trabajar sobre ellas.

3. **Procesado de imagen.** El sistema realizará un procesado digital sobre las dos imágenes recibidas, siendo capaz de localizar unos puntos de interés concretos sobre las imágenes (los necesarios para calcular la medición requerida).

4. **Toma de medidas.** El sistema será capaz de calcular las mediciones de la TA-GT sobre las imágenes recibidas.

5. **Muestra de resultados.** El programa debe mostrar las mediciones calculadas por pantalla. Se mostrarán sobre las dos imágenes las líneas trazadas a partir de los puntos de interés, las cuales permiten calcular la medición de la TA-GT, y el valor de las medidas tomadas.

6. **Modificación de resultados.** Se proporcionará algún medio mediante el cual el usuario podrá modificar la posición de las líneas trazadas sobre las imágenes, cambiando así el valor de la medición.

7. **Almacenamiento de resultados.** El sistema generará un archivo con la información sobre el paciente y los resultados de la medición.

A partir de este listado habría que establecer un orden de relevancia, ya que la prioridad en el sistema es ofrecer un procesado de imagen capaz de tomar medidas de forma automatizada, por lo que se resolverán primero las tareas que permitan cumplir esos requisitos.

Teniendo esto en cuenta, el proyecto debería comenzar implementando los requisitos 2, 3 y 4, en ese orden, ya que serán los que permitan efectuar las mediciones. A continuación se resolverían los requisitos 5 y 6, que permitirán al usuario visualizar y modificar el resultado. El siguiente paso sería guardar los cálculos efectuados, lo que se corresponde con el requisito número 7. Por último se implementaría el requisito 1, una interfaz de usuario que permita interactuar con el sistema de forma gráfica.

3.2 Iteraciones

Teniendo en cuenta los requisitos identificados y la prioridad de estos, el siguiente paso consistiría en definir las iteraciones que permitan su cumplimiento.

El proyecto se dividirá en nueve iteraciones, buscando obtener un programa cuya funcionalidad se irá incrementando al final de cada una de ellas y que estas no tengan una duración de más de dos semanas.

Serían las siguientes:

1. **Carga de ficheros DICOM.** El objetivo de la primera iteración es buscar un método que permita visualizar y trabajar con imágenes en formato DICOM (requisito 2). El sistema resultante permitirá abrir dos imágenes a partir de su ruta de archivo y las mostrará por pantalla.

2. **Procesado: Filtrado.** En la segunda iteración se busca comenzar a trabajar sobre las imágenes realizando un procesado. El programa resultante incrementará la funcionalidad de la primera iteración, mostrando dos imágenes filtradas.

3. **Procesado: Detección de bordes.** Se busca realizar una detección de contornos sobre las imágenes de salida de la anterior iteración. El sistema permitirá visualizar esas imágenes de contornos al finalizar su ejecución.

4. **Procesado: Etiquetado.** Se incrementa la funcionalidad de la tercera iteración identificando cada uno de los bordes en la imagen y cambiando la intensidad de gris de aquellos que son de interés. La salida del sistema serían las imágenes con los contornos etiquetados.

5. **Procesado: Supresión de bordes no deseados.** En la quinta iteración se eliminan los contornos que no fueron identificados durante la etapa de etiquetado. La salida permite visualizar de forma más limpia los contornos de la pierna y los huesos.

6. **Procesado: Detección de puntos característicos.** La sexta iteración añade la funcionalidad de detectar los puntos de interés sobre la imagen de bordes (requisito 3) generada en la iteración anterior. En esta etapa se calcula la posición de las líneas que determinan el valor de las mediciones. El sistema permitiría, al final de esta iteración, visualizar las líneas dibujadas sobre las imágenes de contornos.

7. **Procesado: Visualización y modificación de resultados.** La última iteración dentro del grupo de procesado añadiría diferentes opciones de visualización de los resultados (en diferentes imágenes, con superposición de bordes, etc.), mostrando el valor de la medición (requisito 4) y mostrando las líneas (requisito 5) sobre la imagen. Además, se

permitiría el ajuste manual de esas líneas mediante el uso de teclado y ratón (requisito 6).

8. **Almacenamiento de resultados.** Después de terminar el procesado, la octava iteración añade la funcionalidad de guardar los resultados calculados (requisito 7) en las anteriores al finalizar la ejecución.

9. **Interfaz de usuario.** La última etapa completa las funcionalidades del sistema, dotándolo de una interfaz gráfica (requisito 1) que permite al usuario escoger las imágenes con un buscador, ejecutar el procesado, alternar entre las diferentes opciones de visualización de resultados y guardar estos en un archivo.

Capítulo 4

Análisis y Viabilidad del Proyecto

En este cuarto capítulo se hará un análisis previo de las horas de trabajo y los costes estimados del proyecto.

Siguiendo la metodología comentada en el capitulo 3, se podrán planificar diferentes iteraciones, desglosando el tiempo y el coste invertido en cada fase del proyecto.

Para empezar, dividiremos el trabajo a realizar en tres papeles: un analista, un diseñador y un programador. En este caso, los tres roles estarán desempeñados por la misma persona, pero cada tarea tendrá una dedicación y un coste diferentes.

La primera etapa en el proyecto será el trabajo del analista. Este se encarga de recoger los requisitos que deberá cumplir el sistema final y definir una serie de iteraciones que los vayan cumpliendo de forma incremental. Esta tarea durará un tiempo estimado de 10 horas.

La segunda y la tercera etapa consisten en el diseño y la programación del sistema. Ambas etapas se repetirán para cada una de las iteraciones definidas por el analista, y se podrán solapar cuando sea necesario modificar el diseño propuesto tras la realización de pruebas. En la siguiente tabla se muestra el tiempo en horas de diseño y desarrollo empleados en cada iteración:

Iteración	Diseño	Programación
It 1. Carga de ficheros DICOM	5	15
It 2. Procesado: Filtrado	10	20
It 3. Procesado: Detección de bordes	10	15
It 4. Procesado: Etiquetado	15	25
It 5. Procesado: Supresión de bordes no deseados	5	5
It 6. Procesado: Detección de puntos característicos	20	35
It 7. Procesado: Visualización y modificación de resultados	20	30
It 8. Almacenamiento de resultados	5	10
It 9. Interfaz de usuario	10	15

Tabla 4.1: División por iteraciones del esfuerzo invertido

Por lo tanto, el esfuerzo total invertido por cada trabajador sería de 10 horas por parte del analista, 100 horas del diseñador y 170 horas del programador. Conocidos estos datos y basándonos en el promedio de los salarios ofrecidos para cada puesto en el mercado laboral en España , el coste del proyecto sería el siguiente:

Trabajador	Coste por hora	Horas	Coste total
Analista	35€	10	350€
Diseñador	30€	100	3000€
Programador	20€	170	3400€

Tabla 4.2: Coste total por trabajador

A estos costes se le sumará el del jefe de proyecto, que en este caso serán los tutores, encargados de revisar la evolución del trabajo. Se calcula que se necesitará por su parte una dedicación de 2 horas por cada 30 horas de trabajo del estudiante, por lo que su coste sería el siguiente:

Trabajador	Coste por hora	Horas	Coste total
Jefe de proyecto	60€	9,5	570€

Tabla 4.3: Coste del jefe de proyecto

Por último, se añadirían los costes de amortización del equipo utilizado, en este caso, el ordenador del estudiante. Siendo un ordenador valorado en 1200€ y teniendo una vida útil estimada de tres años años, para realizar un trabajo que ocupará 35 días laborables se estima un coste de amortización de 80€.

Por lo tanto, sumando todos los gastos (estudiante, jefe de proyecto y equipo), tendríamos un coste total del proyecto estimado en 7400€.

Capítulo 5

Desarrollo del sistema

DURANTE el quinto capítulo se explicarán las decisiones de diseño e implementación tomadas durante el desarrollo del sistema a partir de los requisitos iniciales. Se comentarán las opciones contempladas y el resultado final.

5.1 Diseño

Una vez definidos los requisitos en el apartado 3.1 será posible diseñar los diferentes componentes que tendrá el sistema. El flujo del programa se distribuirá en diferentes etapas, representadas en el diagrama de bloques de la figura 5.1.

Los diferentes componentes del sistema serían:

- **Interfaz de usuario.** La interfaz permite al usuario interactuar con el programa (requisito 1), lo cual hace posible seleccionar las imágenes de entrada para realizar las mediciones y, al final de la ejecución, proporcionará la opción de guardar los resultados.
- **Carga de ficheros DICOM.** El sistema trabaja sobre imagen médica en formato DICOM, por lo que se debe utilizar alguna librería que permita cargar y operar sobre estas imágenes (requisito 2).
- **Procesado digital de imagen.** El tercer paso consiste en realizar un procesado de imagen en diferentes etapas (filtrado, detección de bordes, etiquetado, supresión de bordes, detección de puntos característicos y visualización y modificación de resultados) que tendrá como resultado las mediciones sobre las imágenes de entrada (requisitos 3 y 4), que se mostrarán por pantalla pudiendo ser modificadas (requisitos 5 y 6). Se debe emplear una librería que permita realizar operaciones de procesado sobre imagen.
- **Almacenamiento de resultados.** Como último bloque, el programa permitirá guardar el resultado de las mediciones en un fichero (requisito 7).

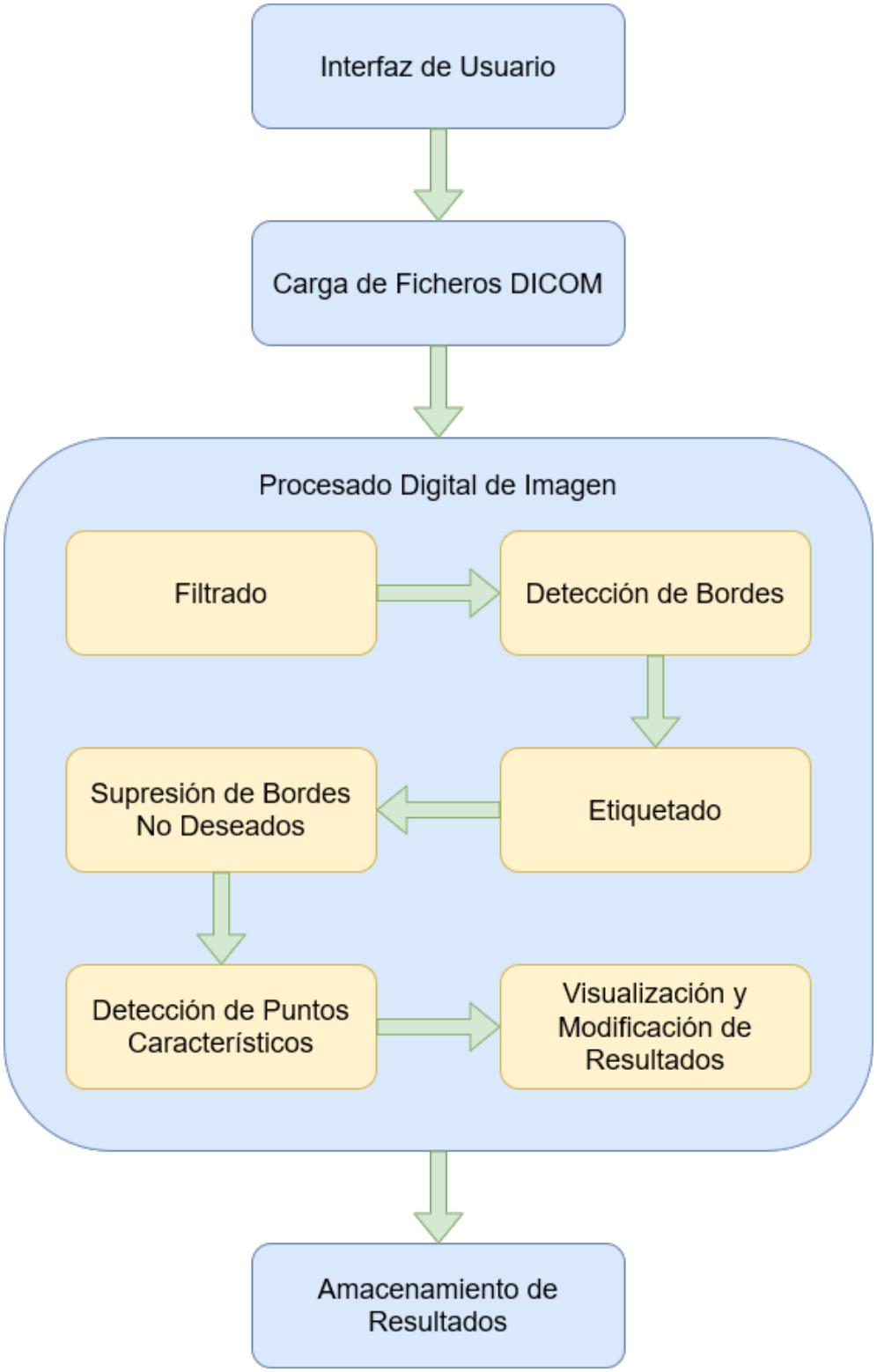


Figura 5.1: Diseño por bloques del sistema

5.1.1 Diseño Iterativo

El diseño por bloques del sistema permite realizar un desarrollo siguiendo un proceso iterativo, descrito a continuación. En cada una de las iteraciones se implementa alguno de los módulos o submódulos, valorando posibles soluciones y escogiendo la que ofrezca resultados más convenientes.

5.1.1.1 Carga de Ficheros DICOM

El objetivo de este trabajo es realizar un procesado digital de imagen médica. Como ya se vio anteriormente, el estándar más utilizado en la actualidad por los dispositivos que trabajan con este tipo de imagen es DICOM, por lo que también será el formato utilizado en este trabajo.

DICOM no es un formato con el que se pueda trabajar utilizando cualquier lenguaje de programación o librería gráfica. Además, el propósito del sistema a implementar es que se pueda utilizar indistintamente en diferentes plataformas y que admita la posibilidad de continuar su desarrollo en un futuro. Para cumplir con estos requisitos se ha escogido el lenguaje de programación Java, combinado con la librería ImageJ.

ImageJ es un programa de edición y procesado de imagen de código abierto escrito en Java, por lo que se puede utilizar como base para desarrollar otras herramientas de procesado. Además proporciona soporte para imagen médica, pudiendo cargar y trabajar con imágenes en formato DICOM de forma nativa. Dispone también de una amplia comunidad de usuarios que, a lo largo de los años, han ido desarrollando una amplia variedad de plugins, aportando nuevas operaciones de procesado y edición de imagen. Todo esto hace que sea una de las mejores librerías sobre la que desarrollar nuestro sistema.

El resultado de esta primera iteración sería un programa que permite cargar y visualizar imágenes médicas usando la librería ImageJ. En la captura 5.2 se puede visualizar el resultado de la ejecución.

5.1.1.2 Procesado Digital de Imagen: Filtrado

Una vez cargados los archivos ya se puede comenzar con el procesado de imagen. Para ello se seguirá utilizando la librería ImageJ, que proporciona diferentes herramientas para aplicar filtros y máscaras de convolución. Además, con el uso de plugins se puede expandir el conjunto de filtros disponibles.

Antes de decidir el filtro que se va a aplicar debemos analizar el tipo de ruido presente en las imágenes de entrada. En el conjunto de imágenes se puede apreciar tanto granulado como ruido impulsional en algunas zonas. Además, el nivel de gris de la zona correspondiente al hueso y a la pierna no es totalmente homogéneo, por lo que habrá que evitar que el sistema detecte estas variaciones como un contorno en los siguientes pasos.

La primera prueba se realiza con un filtrado gaussiano, ya que es un método que presenta buenos resultados en suavizado de imagen en presencia de granulado.

Para comprobar la efectividad del filtro se comprueba el resultado de aplicar detección de bordes con el algoritmo de Canny a las imágenes de salida. En la figura 5.3 se puede ver el resultado del filtro con valores de $\sigma = 1$ y $\sigma = 2$. En el primer caso el suavizado no sería suficiente, ya que todavía se conserva granulado en ciertas zonas de las imágenes, lo cual produce una posterior sobrestimación en la detección de bordes, detectando contornos que no se corresponderían con ninguno de los huesos. Por el contrario, en el segundo caso el filtrado sería demasiado agresivo, difuminando en exceso algunos de los contornos clave a la hora de detectar el hueso, provocando una detección errática en la que aparecen bordes incompletos.

Por estos motivos, las pruebas realizadas con el filtro gaussiano concluyen sin encontrar un punto óptimo con un buen balance entre sobrestimación y subestimación en la posterior etapa de detección de bordes.

Las siguientes pruebas se realizan con el filtro de medianas. De nuevo, para comprobar los resultados ofrecidos por el filtro se aplica detección de bordes mediante el algoritmo de Canny a los resultados. Los test se realizan con un vecindario de tamaño *3x3* y otro de tamaño *5x5*, cuyos resultados se pueden ver en la figura 5.4.

Para un vecindario *3x3*, el grado de filtrado no es suficiente, conservando ruido en la imagen que es detectado como un contorno por el algoritmo de Canny. En el caso del vecindario *5x5*, el filtro comienza a eliminar los bordes, especialmente en la situación del hueso del fémur, como se ve en la imagen 5.4c. Esto provoca que esos bordes no se puedan extraer durante la detección de contornos, que sería incompleta, con una importante subestimación.

Las pruebas realizadas muestran que el filtro de la mediana sigue sin ser una buena opción para este tipo de imágenes al no ofrecer un valor óptimo que pueda equilibrar el grado de filtrado, ya que un valor bajo no puede eliminar el ruido y un valor más alto borraría los contornos finos.

Por último se prueba el filtro bilateral. Este es más complejo que los anteriores, utilizando un algoritmo que combina el filtrado espacial con el filtrado en el rango de grises para obtener un buen suavizado preservando los bordes. Para su implementación es necesario emplear un plugin [18] de ImageJ, ya que no es una operación que soporte de forma nativa.

En esta etapa se va probando los resultados que ofrece el filtro con diferentes valores de vecindario y sigma. Se comprueban valores de vecindario en un rango entre *5x5* y *21x21*. Para el valor de sigma, los valores probados oscilan entre 50 y 500. Aumentar el valor de vecindario provoca que el promediado se calcule entre un número mayor de píxeles, lo que resulta en un mejor suavizado en zonas con intensidad de gris similar. Aumentar el valor de sigma produce una mayor agresividad en el filtrado, suavizando puntos con intensidades de gris diferente.

En las pruebas realizadas posteriormente con el algoritmo de Canny, se pudo ver que con valores bajos de vecindario el sistema no era capaz de suavizar los falsos contornos que no se corresponden con el borde de la pierna y el hueso. Se puede ver un ejemplo del resultado del filtro con un valor de vecindario *9x9* y un valor $\sigma = 150$ en la figura 5.5a. Esta configuración produce sobrestimación en la detección de bordes, por lo que no sería válida.

Cuando se aumenta el tamaño del vecindario, el resultado de el filtrado en niveles de gris similares es más homogéneo, por lo que se puede optar por un valor alto. El valor de sigma, sin embargo, produce resultados demasiado agresivos en los bordes cuando es muy elevado, por lo que deberá ser más moderado. Con valores de sigma superiores a 350, el difuminado de contornos es demasiado elevado con cualquier tamaño de filtro, mientras que con valores inferiores a 150 se mantiene cierto ruido que provoca la detección de falsos contornos.

Un tamaño de vecindario de *15x15* produce un suavizado suficiente en zonas con similar valor de gris, mientras que con un valor $\sigma = 300$ se encontró un buen balance que elimine el ruido sin degradar en exceso los bordes. En la figura 5.6 se puede ver un ejemplo de la salida producida a partir de las imágenes obtenidas en la anterior iteración. Con estos valores, la posterior detección de contornos lograría un resultado bastante preciso sin encontrar una cantidad excesiva de bordes no deseados. Estos motivos son los que determinan que esta configuración del filtro bilateral sea la que se utilizará para el resto del sistema.

5.1.1.3 Procesado Digital de Imagen: Detección de Bordes

El objetivo de esta iteración será extraer los contornos de las imágenes filtradas en el paso anterior por lo que, en primer lugar, será necesario analizar las opciones disponibles para llevar a cabo esta tarea.

Para la detección de contornos se podrían utilizar diferentes operadores tanto de primera como de segunda derivada. De forma nativa, ImageJ proporciona un algoritmo de búsqueda de bordes basado en estos operadores. El problema de estos métodos es que devuelven una imagen de bordes no umbralizada, resultando en bastantes bordes débiles que no serían de utilidad para el análisis de la imagen. Por este motivo, el único algoritmo probado para esta iteración es el detector de Canny.

El algoritmo de Canny proporciona una herramienta capaz de detectar bordes con precisión, pudiendo escoger un doble umbral para ignorar contornos muy débiles. Además, el resultado será una imagen con bordes de un único píxel de grosor, lo que facilitará el posterior análisis de la imagen.

La integración de este algoritmo se realiza por medio de un plugin [19] de ImageJ, ya que la operación no está soportada de forma nativa.

Para decidir los valores de filtrado y umbral utilizados se realizan pruebas con valores de sigma comprendidos entre 1 y 4 y valores para los umbrales entre 0 y 1.

En primer lugar se buscaría determinar el grado de filtrado. Con un valor de $\sigma = 1$, como el empleado en las imágenes de la figura 5.7, el resultado contiene una elevada cantidad de ruido en la detección, por lo que se deberán utilizar valores más elevados. Un valor de $\sigma = 4$ produce un filtrado excesivo, resultando en una subestimación en la detección de bordes, produciendo contornos incompletos y cortados.

Con un valor de $\sigma = 3$ se encuentra un buen equilibrio en la detección de contornos, sin generar ruido en exceso ni eliminar bordes de interés. En la figura 5.8 se puede ver un ejemplo con este valor de sigma y valores de umbral alto de 0,50 y bajo de 0,30.

Sin embargo, estos valores de umbral no son los óptimos para la detección del contorno del hueso ya que, como se puede ver, produce discontinuidades y contornos incompletos. Esta subestimación en la detección del borde se aprecia de forma más pronunciada en el caso de la imagen de la Garganta de la Tróclea (5.7a). En este caso se eliminan algunos de los puntos de interés necesarios para trazar las mediciones, por lo que será necesario probar valores de umbral más bajos.

Tras probar diferentes valores, se encuentra que el algoritmo genera un resultado balanceado con un valor de umbral alto de 0,35 y un umbral bajo de 0,15, sin detectar demasiados falsos contornos ni eliminar los que son necesarios para el cálculo de las mediciones, por lo que este será el valor final utilizado en el sistema.

5.1.1.4 Procesado Digital de Imagen: Etiquetado

En esta etapa del desarrollo se busca etiquetar los resultados proporcionados por el detector de Canny, ya que este no proporciona un método automático para identificar su salida.

La implementación de esta fase se realiza de forma manual, encapsulando en una clase las funciones necesarias para identificar los contornos de interés sobre la imagen de bordes. El resultado del etiquetado será una imagen de contornos en la que los correspondientes a la pierna y el hueso tendrán un valor de intensidad en escala de grises diferente.

Para trabajar sobre la imagen se utilizan las funciones proporcionadas por ImageJ, que permiten operar directamente sobre el array de píxeles. Se proporcionan dos funciones públicas, una para etiquetar los contornos de la rótula y otra para los de la tibia.

El funcionamiento en ambos casos es similar. El algoritmo parte del punto central entre las dos piernas en la imagen y busca los contornos a izquierda y derecha. El primero en ser encontrado se corresponde con el borde de la pierna. Una vez localizado, se recorre todo el conjunto de píxeles conectados formando un camino, de forma que todo el contorno quede etiquetado con el mismo valor.

Una vez localizadas las piernas, los huesos se buscarán a ambos lados en el interior de los contornos. El punto de partida dependerá de si se trata de la imagen de la rótula o de la tibia, buscando el hueso en la zona central de la pierna en el primer caso y en la zona superior

en el segundo. De nuevo, cuando se localiza un píxel perteneciente al contorno buscado, se recorrerá el camino de píxeles conectados hasta que todo el hueso haya sido etiquetado.

Para el caso de la imagen de bordes obtenida en la iteración anterior, las funciones de etiquetado producirían la salida de la figura 5.10. Como se puede apreciar, los contornos de interés correspondientes a la pierna y a los huesos de la tibia y el fémur tienen un valor de gris diferente, mientras que el resto de píxeles mantiene su intensidad original.

5.1.1.5 Procesado Digital de Imagen: Supresión de Bordes No Deseados

En esta iteración se eliminarán los bordes no deseados de la imagen etiquetada en el paso anterior. El objetivo es facilitar la visualización de los resultados para etapas futuras.

Como en el caso anterior, no se dispone de ninguna función que elimine los píxeles sin etiquetar de una imagen, por lo que se deberá hacer de forma manual.

La supresión de bordes se integra en la clase de etiquetado. Su funcionamiento consiste en recorrer toda la imagen de bordes, eliminando todos los píxeles sin ninguna etiqueta.

Para las imágenes obtenidas en la anterior iteración se produciría el resultado de la figura 5.11.

5.1.1.6 Procesado Digital de Imagen: Detección de Puntos Característicos

Una vez se tiene la imagen con los contornos ya etiquetados, el siguiente objetivo consistirá en buscar los puntos característicos sobre ellos. Estos puntos serán los que permitirán trazar las líneas utilizadas para la toma de medidas.

No se dispone de ninguna librería que tenga conocimiento sobre este tipo concreto de imágenes, por lo que la única opción es realizar una implementación manual que busque los puntos deseados. Las funciones correspondientes a la detección se encapsularán en una misma clase, que proporcionará un objeto detector con métodos para la búsqueda y definición de líneas en ambas imágenes.

De nuevo, las operaciones necesarias para esta iteración se realizaran trabajando sobre el array de píxeles de la imagen con los métodos proporcionados por la librería ImageJ.

Para guardar las coordenadas de las líneas calculadas se creará un nuevo objeto que almacene las imágenes (procesadas e iniciales) y las líneas que marcan los puntos característicos, en lugar de dibujarlas directamente sobre la imagen. Esto permitirá modificar su posición más adelante.

La localización será llevada a cabo de la misma forma para la pierna izquierda y derecha y tendrá el siguiente orden:

1. **Línea inferior.** El proceso comenzará trabajando con la imagen de la báscula rotuliana. Sobre ella se buscará la línea inferior tangencial al margen posterior de los cóndilos.

En la imagen 5.12 se puede ver un ejemplo del resultado de este algoritmo cuando recibe la imagen de la báscula rotuliana calculada en la iteración anterior.

Se comenzará situando una línea horizontal a una altura de varios píxeles por debajo de la zona etiquetada como hueso (linea azul de puntos 1 en la imagen 5.12), correspondiente al fémur. Luego, en un proceso iterativo, se irán subiendo los vértices de la línea hasta que se produzca contacto con el contorno de alguno de los dos cóndilos (linea azul de puntos 2 en la imagen 5.12). Cuando existe contacto, se deja de subir el vértice del lado en el cual se produce y se continúa en el opuesto hasta que la línea toque el fémur en ambos lados (linea continua blanca en la imagen 5.12).

Una vez terminado este proceso, se realiza un ajuste bajando toda la línea y repitiendo el algoritmo, de forma que el ángulo resultante sea el adecuado, para que la línea toque cada cóndilo en un solo punto.

El algoritmo utilizado tendría el siguiente pseudocódigo:

```
1
2     y1 = pixelMasBajo(etiquetaHueso) + margen
3     y2 = y1
4
5     dibujarLinea(x1, y1, x2, y2)
6
7     repetir hasta (encontrarLinea) {
8
9         si (no tocaCondiloIzquierdo) y1 = y1 - 1
10        si (no tocaCondiloDerecho) y2 = y2 - 1
11
12        si (tocaCondiloIzquierdo y tocaCondiloDerecho) {
13
14            si (realizarAjuste) {
15                y1 = y1 + 5
16                y2 = y2 + 5
17                realizarAjuste = falso
18            }
19            si (no realizarAjuste) encontrarLinea = verdadero
20
21        }
22
23        dibujarLinea(x1, y1, x2, y2)
24
25    }
26
```

2. **Primera línea perpendicular.** El cálculo de la segunda línea se realizará de nuevo

sobre la imagen de la báscula rotuliana. Esta línea será vertical y perpendicular a la primera, cortando el contorno del fémur en la parte más profunda del surco troclear.

El algoritmo utilizado tendrá la estructura representada a continuación con pseudocódigo:

```
1
2     x1 = puntoMedioHorizontal(etiquetaHueso) + desplazamiento
3
4     dibujarLineaPerpendicular(x1, lineaInferior)
5
6     repetir hasta (margen) {
7
8         si (no esMinimo) x1 = x1 - 1
9         si (esMinimo) salir
10
11        dibujarLineaPerpendicular(x1, lineaInferior)
12
13    }
14
```

El resultado de este cálculo se puede ver en la imagen 5.13, cuando el algoritmo recibe la salida del paso anterior.

Se comienza calculando una línea vertical perpendicular a la línea inferior y con un valor horizontal concreto para uno de sus vértices. Este valor situará la línea a uno de los lados del centro del hueso (linea azul de puntos 1 en la imagen 5.13). Luego, la línea es movida en el eje horizontal sobre la inferior, parando cuando encuentre un punto mínimo en el eje vertical (linea continua blanca en la imagen 5.13), que se corresponderá con el punto más profundo en el contorno de la tróclea. El recorrido en el cual buscará ese mínimo llegará hasta el otro lado del surco troclear (linea azul de puntos 2 en la imagen 5.13), limitando el margen de operación del algoritmo.

3. **Segunda línea perpendicular.** Para el trazado de la última línea se utilizará la imagen de la Tuberosidad Anterior de la Tibia. La línea ha de ser vertical y perpendicular a la primera, cortando el contorno de la tibia en el punto más saliente.

   En este caso, el algoritmo tendría un pseudocódigo similar al anterior, buscando el punto de máxima altura en lugar del menor.

   El resultado de este paso se puede ver en la imagen 5.14, recibiendo como entrada la imagen resultante del etiquetado.

   Se comenzaría calculando la línea vertical perpendicular a la línea inferior, situándola a uno de los lados del hueso (linea azul de puntos 1 en la imagen 5.14). Luego se movería

sobre la línea inferior, pasando por encima del contorno del hueso, parando cuando corte el borde en el punto máximo en el eje vertical, el cual corresponde al punto más saliente de la TAT (linea continua blanca en la imagen 5.14). El margen de operación llegaría hasta el otro lado del hueso (linea azul de puntos 2 en la imagen 5.14).

5.1.1.7 Procesado Digital de Imagen: Visualización y Modificación de Resultados

La última etapa del procesado de imagen consistirá en la visualización por pantalla y modificación de resultados.

Por pantalla se mostrará una imagen suma, compuesta por las dos imágenes iniciales superpuestas. Sobre esta imagen se dibujarán tres líneas que determinan las medidas, así como el valor de la distancia entre las dos verticales, correspondiente a la medida TA-GT.

El objeto encargado de almacenar tanto las imágenes como las líneas proporciona diferentes métodos para el visualizado, siendo posible mostrar una imagen conjunta (5.16) o dos imágenes separadas, mostrando sus respectivas líneas en cada una de ellas (5.15). En todos los casos también estaría permitido destacar los contornos de interés en un color diferente al de la imagen (5.17 y 5.18).

Como último paso, el usuario podría ajustar las mediciones de forma manual (sobre cualquiera de las imágenes) con el objetivo de corregir posibles errores. Se dispondrá de dos métodos para realizar los ajustes:

- **Ratón:** permitirá al usuario desplazar las líneas pulsando y arrastrando sobre alguno de los vértices. Cuando se desplaza la línea inferior, el ángulo de las perpendiculares debe cambiar, por lo que se recalcula su posición.

- **Teclado:** permite que el usuario desplace las líneas seleccionando previamente alguno de los vértices con el ratón. El vértice se desplazará píxel por píxel utilizando las flechas. Este método permite realizar un ajuste más preciso que en el caso anterior. De nuevo, cuando la línea desplazada es la inferior, la posición de las perpendiculares se vuelve a calcular.

Con estos métodos se podría corregir el error en la medición de la rodilla derecha, que debido a una discontinuidad en la imagen de la base rotuliana no sitúa la línea perpendicular en el punto correcto. Las mediciones modificadas se pueden ver en la figura 5.19.

5.1.1.8 Almacenamiento de Resultados

En esta iteración se busca dar persistencia a los resultados de las operaciones realizadas, que serán guardados cuando se finalice la etapa de procesado.

El método utilizado será guardar la información sobre las imágenes, el paciente y las medidas en un fichero de texto, y guardar un archivo con la imagen compuesta de la medición de la TA-GT.

El fichero de texto guardará los siguientes campos:

- ID del paciente correspondiente a las imágenes.
- Ruta donde se encuentran las dos imágenes de entrada.
- Resultado de la medición TA-GT para ambas rodillas.

5.1.1.9 Interfaz de Usuario

En la última iteración se implementará una interfaz gráfica para que el usuario sea capaz de interactuar con el sistema. Esta interfaz debe permitir seleccionar las dos imágenes de entrada, ejecutar diferentes opciones de procesado y guardar los resultados.

La solución escogida para el desarrollo de esta interfaz será la librería Java Swing, la cual proporciona objetos para la creación de ventanas, campos de texto, botones y más elementos para permitir la interacción gráfica.

Al ejecutar el programa, se abrirá una ventana con el aspecto de la imagen 5.20. Se le permitirá al usuario seleccionar dos imágenes, para lo cual se desplegará el buscador de archivos mostrado en la imagen 5.21. El botón *Procesar* estará bloqueado hasta que las dos imágenes hayan sido seleccionadas.

Cuando se presiona el botón *Procesar*, el aspecto de la ventana pasará a ser el de la imagen 5.22. Además, se desplegará la imagen compuesta por las dos iniciales, mostrando sobre ella la medida de la Tuberosidad Tibial Anterior - Garganta de la Tróclea, y en el campo con el ID del paciente se escribirá el correspondiente a las dos imágenes.

En esta ventana se podrán utilizar los siguientes botones:

- **Mostrar GT.** Abre una nueva ventana con la imagen del fémur, mostrando sobre ella la línea inferior y la primera perpendicular.
- **Mostrar TAT.** Abre una nueva ventana con la imagen de la tibia, dibujando sobre ella la línea inferior y la segunda perpendicular.
- **Mostrar TA-GT.** Abre una nueva ventana (en caso de que hubiera sido cerrada) con la imagen combinada, trazando sobre ella todas las líneas y escribiendo el valor de las mediciones.
- **Contornos.** Permite dibujar o borrar la superposición de contornos en todas las ventanas de imagen abiertas.

- **Guardar.** Permite seleccionar un nombre para el archivo de texto y de imagen y guardar los dos ficheros con los resultados.

5.2 Análisis de Resultados

En esta sección se probará el funcionamiento del sistema con diferentes imágenes de entrada, seleccionadas a partir de los estudios de TAC de pacientes en formato DICOM disponibles.

Las imágenes pertenecientes al estudio con las que se llevan a cabo las pruebas fueron escogidas por el estudiante que desarrolla este proyecto y no por personal médico, por lo que pueden no ser las entradas óptimas para realizar la medición.

Se etiquetarán las líneas que delimitan las medidas en todas las imágenes de resultados. La etiqueta *LB* se corresponde a la línea horizontal, tangencial al margen posterior de los cóndilos del fémur. La etiqueta *LP1* se corresponde con la primera línea perpendicular que pasa por el punto más bajo del surco troclear. Por último, la etiqueta *LP2* indica la segunda línea perpendicular que corta el hueso de la tibia en su punto más saliente.

Además, los contornos etiquetados también serán superpuestos en todas las imágenes de resultados, para poder observar si la detección ha sido la correcta. Se superpondrá en azul el contorno de la imagen de la Garganta de la Tróclea y en verde el contorno de la Tuberosidad Anterior de la Tibia.

A continuación se comentará el resultado para cada una de las pruebas realizadas:

1. El primer test recibe como imágenes de entrada las dos que se pueden ver en la figura 5.23. En la imagen 5.24 se ven los resultados generados por el sistema.

   Para la pierna izquierda, la detección de contornos realizada en la imagen del fémur presenta algún fallo en forma de discontinuidades en los bordes laterales del hueso. A pesar de esto, el contorno de los cóndilos y el surco trocleano está bien identificado, por lo que no afecta al resultado. En la imagen de la tibia no se presenta ninguna discontinuidad y la detección es bastante precisa.

   En cuanto a la detección de puntos característicos, vemos que la línea horizontal se ajusta correctamente el margen posterior de los cóndilos. La primera línea vertical que corta el surco trocleano, en cambio, tendría una pequeña desviación hacia la derecha de alrededor 1mm del punto más bajo (sería un médico quien debería valorar y ajustar la medición si fuese necesario). La segunda línea vertical sí cortaría correctamente la zona más saliente de la tibia. Con estos puntos característicos, la medición calculada sería de 7,72mm.

   El la pierna derecha, la detección de contornos es bastante precisa en los dos casos, con una discontinuidad irrelevante en el borde inferior de la pierna en la imagen de la tibia.

La detección de puntos característicos en este caso es la correcta para la línea horizontal, tocando el margen posterior del fémur. En la primera línea vertical, sin embargo, se volvería a apreciar una pequeña desviación de entre 1 y 2mm hacia la izquierda (de nuevo, un médico debería valorar la medición y ajustar en caso de que fuese necesario). La localización de la segunda línea perpendicular sí sería la correcta. La medida generada por el sistema sería de 9,20mm.

2. La segunda prueba produce la salida 5.26 a partir de las imágenes de la figura 5.25.

   Para la rodilla izquierda, la detección de contornos es bastante precisa para las dos imágenes de entrada, con la única excepción de que en la imagen del fémur detecta un borde doble en el lateral derecho.

   La localización de puntos característicos también es precisa, ajustando bien la línea inferior y trazando las dos perpendiculares por el surco trocleano y la parte más saliente de la tibia correctamente. Con esto en cuenta, el resultado de la medida sería de 9,81mm.

   En el caso de la rodilla derecha, la detección de bordes es precisa en el caso de la imagen de la TAT, pero falla en la imagen de la GT. Esto puede ser debido a que en la imagen de entrada no se aprecia correctamente el surco troclear, por lo que no sería la adecuada para realizar la medición.

   La medición en este caso sería errónea, ya que el sistema no es capaz de calcular correctamente la localización de la primera línea perpendicular. El resto de líneas sí estarían bien situadas, ajustando bien el margen de los cóndilos en el caso de la línea baja y localizando el margen más saliente en la tibia.

3. La tercera prueba se haría con las imágenes de entrada de la figura 5.27 y la salida 5.28.

   En la pierna izquierda, la detección de contornos realizada es buena en los dos casos, fallando únicamente en la zona lateral derecha del hueso del fémur, lo cual no afecta al resultado.

   Las líneas que delimitan los puntos característicos se trazan de forma correcta, ajustando bien la zona posterior del fémur en el caso de la línea vertical, cortando el surco trocleano en el caso de la primera linea perpendicular y situando bien la zona más saliente de la tibia con la segunda línea perpendicular. El resultado de esta medida sería de 5,80mm.

   Para la pierna derecha, nos encontramos con que la imagen de entrada 5.27a no es la adecuada para apreciar el surco trocleano. Por este motivo, el resultado de la medición no será preciso, a pesar de que la detección de contornos sea bastante acertada, a excepción de alguna sobrestimación en los bordes del fémur.

Teniendo en cuenta el problema con la imagen de entrada, la medida no podrá ser la correcta. A pesar de esto, la localización de la línea inferior y la segunda perpendicular sí sería la correcta.

4. La cuarta prueba presenta una situación similar a la anterior, partiendo de las imágenes de entrada de la figura 5.29, y obteniendo como resultados los vistos en la imagen 5.30.

   En este caso, sería la pierna izquierda la que no ofrece una imagen del fémur a partir de la cual se puedan realizar las mediciones, por lo que el único resultado a valorar sería la detección de contornos en la imagen de la tibia, que se realiza de forma correcta.

   En la pierna derecha, la detección de contornos encontraría algún problema de continuidad en la imagen del fémur, pero esta no afectaría a puntos clave. Los contornos correspondientes a la tibia sí se detectan de forma precisa.

   Las mediciones tomadas en este caso serían acertadas, con una línea inferior bien ajustada al margen de los cóndilos, una primera perpendicular que localiza correctamente el surco troclear y una segunda vertical que localiza correctamente el margen de la tibia.

5. El quinto test parte de las imágenes de entrada de la figura 5.31. Los resultados obtenidos se pueden ver en la imagen 5.32.

   En este caso, en la rodilla izquierda, la detección de contornos sería correcta en las dos imágenes, a excepción del lateral derecho en el hueso del fémur. De nuevo, esta discontinuidad no afecta a puntos críticos para el cálculo de las medidas y no supone un problema.

   La línea inferior estaría localizada correctamente, al igual que la segunda línea perpendicular que corta el extremo de la tibia. En la primera perpendicular se apreciaría una pequeña desviación de alrededor de 1mm hacia la derecha del surco troclear (tendría que ser valorado y ajustado por el personal médico si se considerase necesario). El resultado de la medida sería de 6,33mm.

   En la rodilla derecha la detección de bordes realizada sería correcta en los dos casos, sin ninguna incorrección que destacar.

   El cálculo de medidas también sería correcto. La línea inferior se ajusta al límite del hueso, se corta el surco trocleano en el punto adecuado y lo mismo sucede para el extremo de la tibia. La medida resultante sería de 10,58mm.

6. En el último caso apreciamos dos imágenes de entrada (5.33) con una cantidad muy elevada de ruido. El resultado obtenido de forma automática se puede ver en la imagen 5.34.

Tanto para la rodilla derecha como en la izquierda, el ruido en las imágenes de entrada provoca que el sistema falle completamente en la detección de contornos, resultando en una importante sobrestimación. Esto provoca que los contornos de interés no sean detectados y etiquetados correctamente, tanto en la imagen del fémur como en el caso de la tibia.

En esta situación, la medición aportada automáticamente carece de valor, ya que ninguno de los puntos de interés pudo ser identificado con precisión.

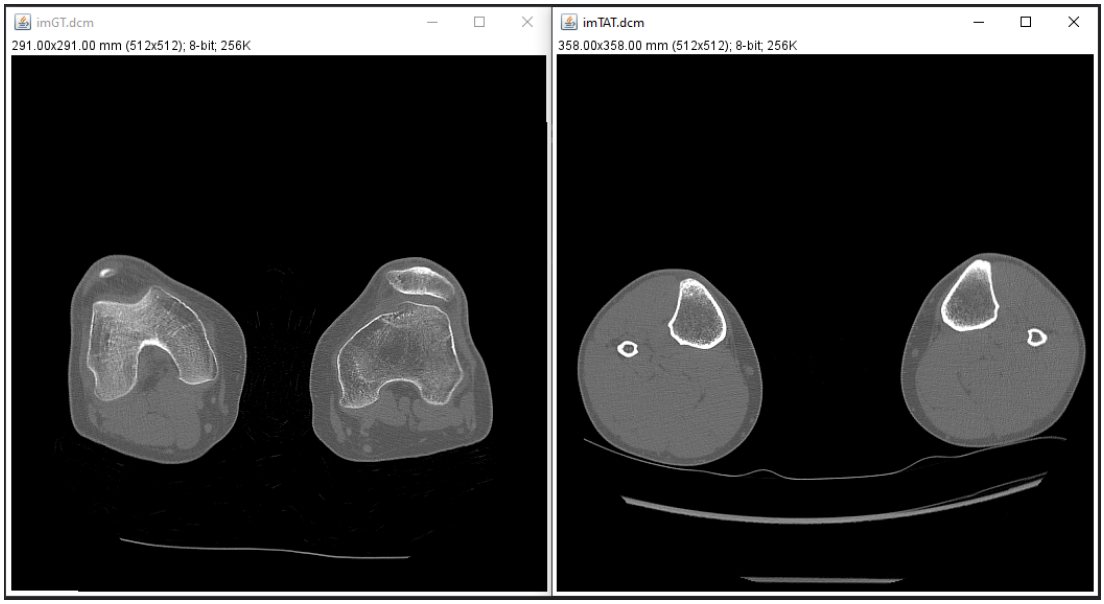
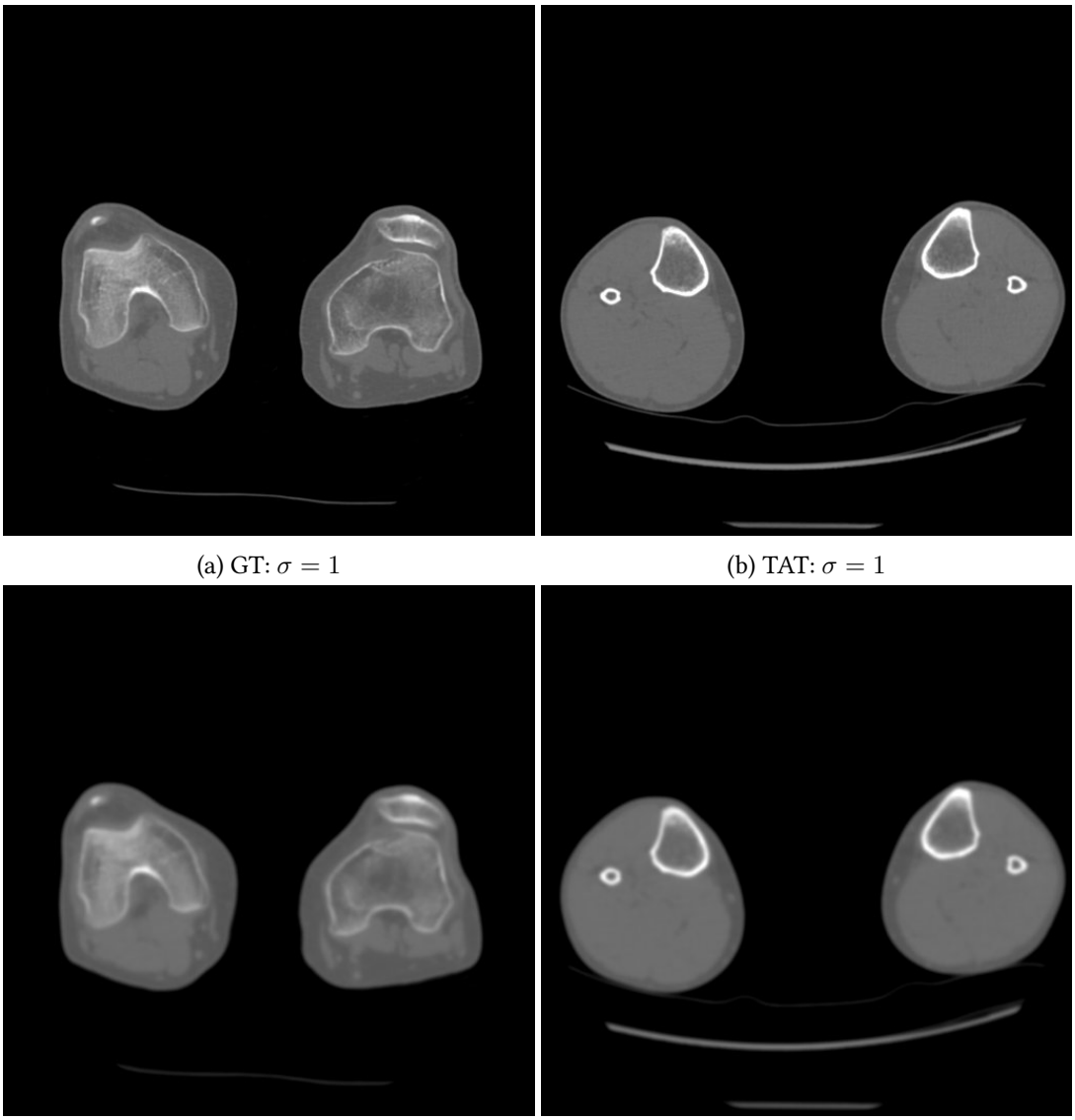


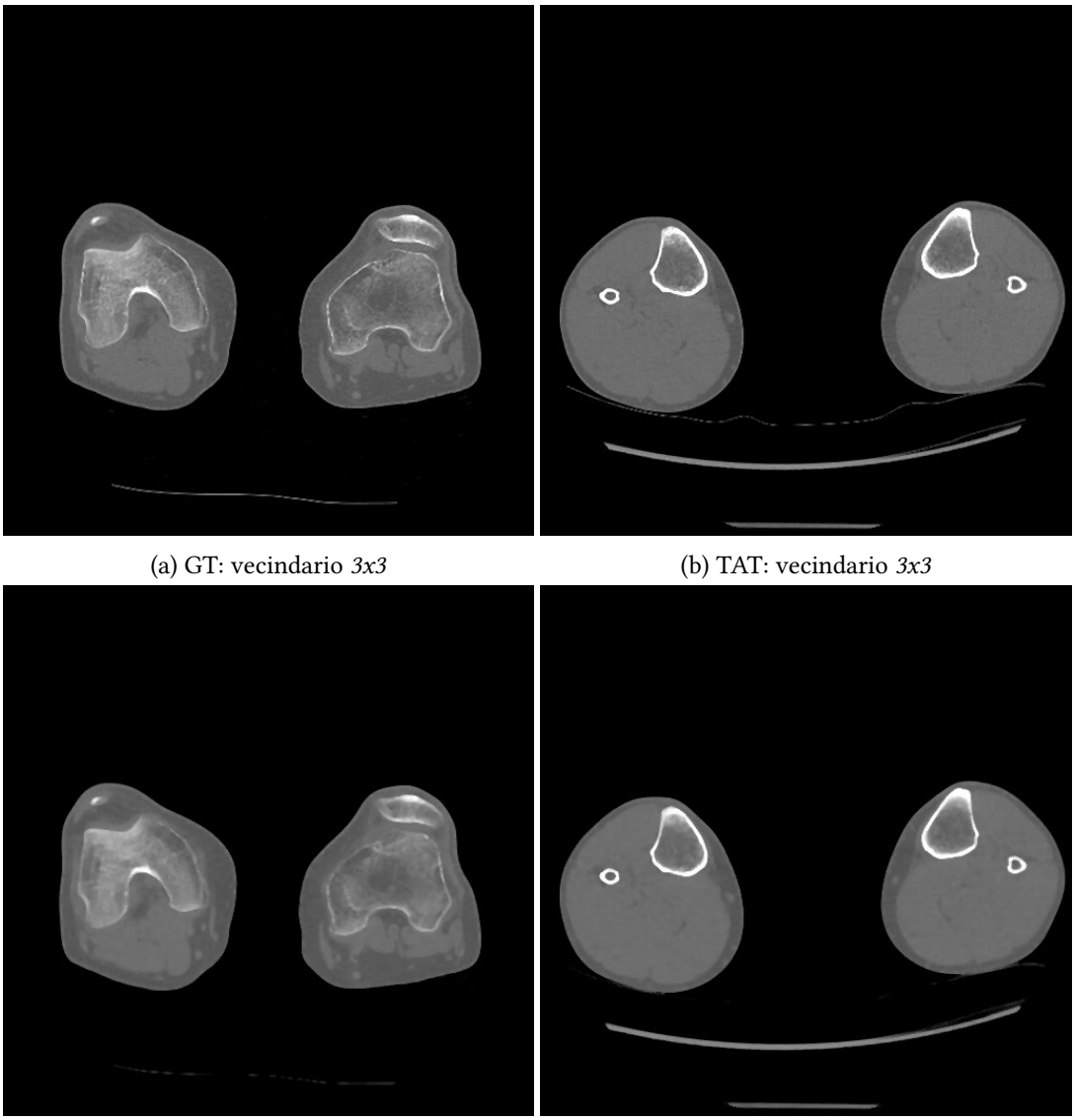
Figura 5.2: Carga de ficheros DICOM



(a) GT: $\sigma = 1$ (b) TAT: $\sigma = 1$

(c) GT: $\sigma = 2$ (d) TAT: $\sigma = 2$

Figura 5.3: Filtrado gaussiano



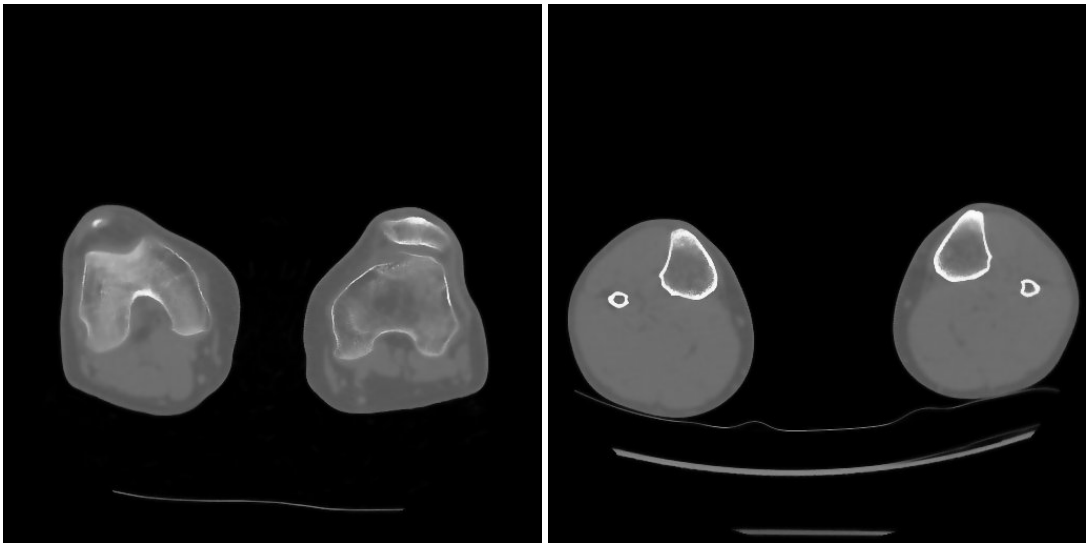
(a) GT: vecindario *3x3*

(b) TAT: vecindario *3x3*

(c) GT: vecindario *5x5*

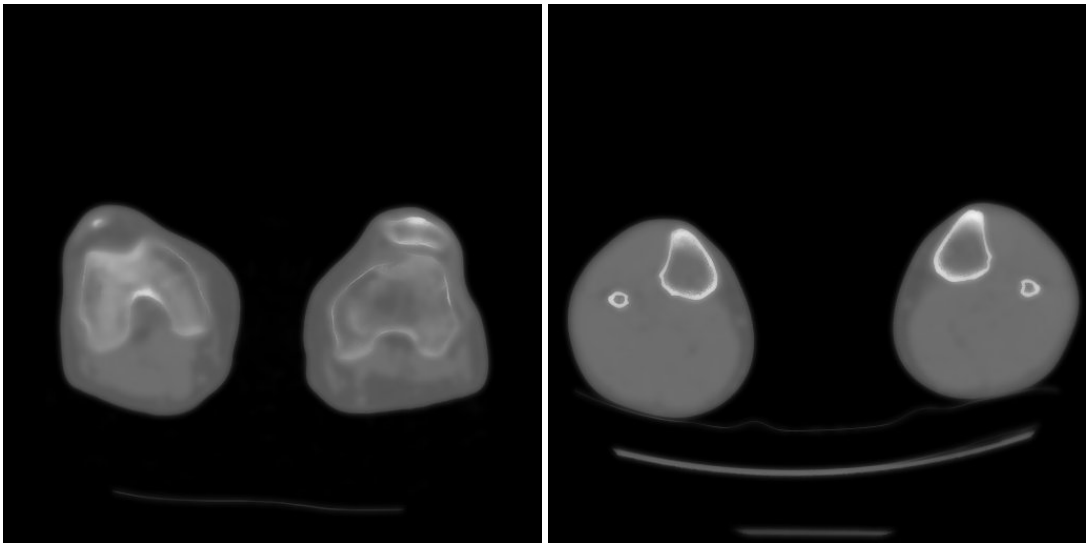
(d) TAT: vecindario *5x5*

Figura 5.4: Filtrado de medianas



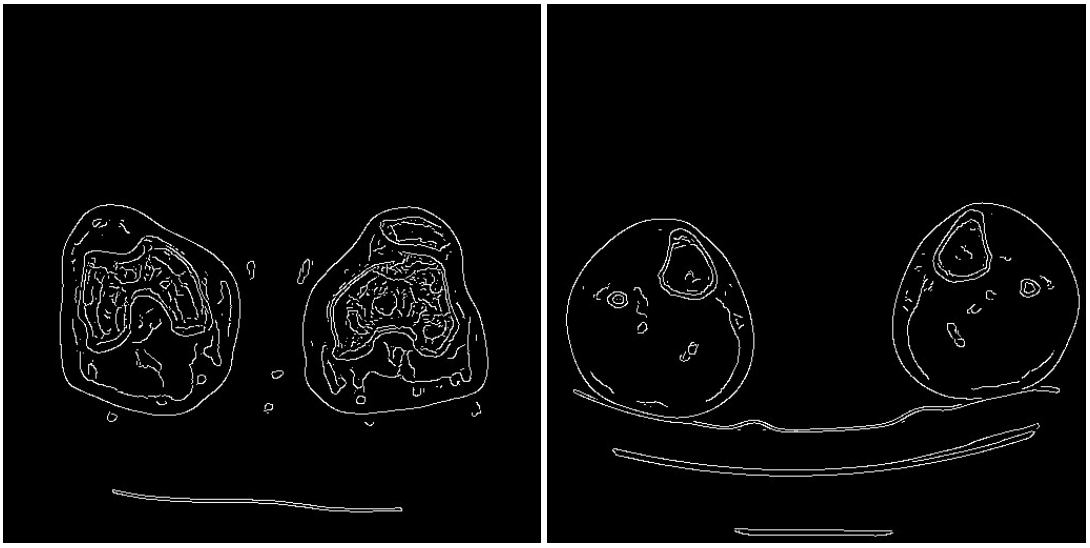
(a) GT (b) TAT

Figura 5.5: Filtrado bilateral: vecindario *9x9*, $\sigma = 150$



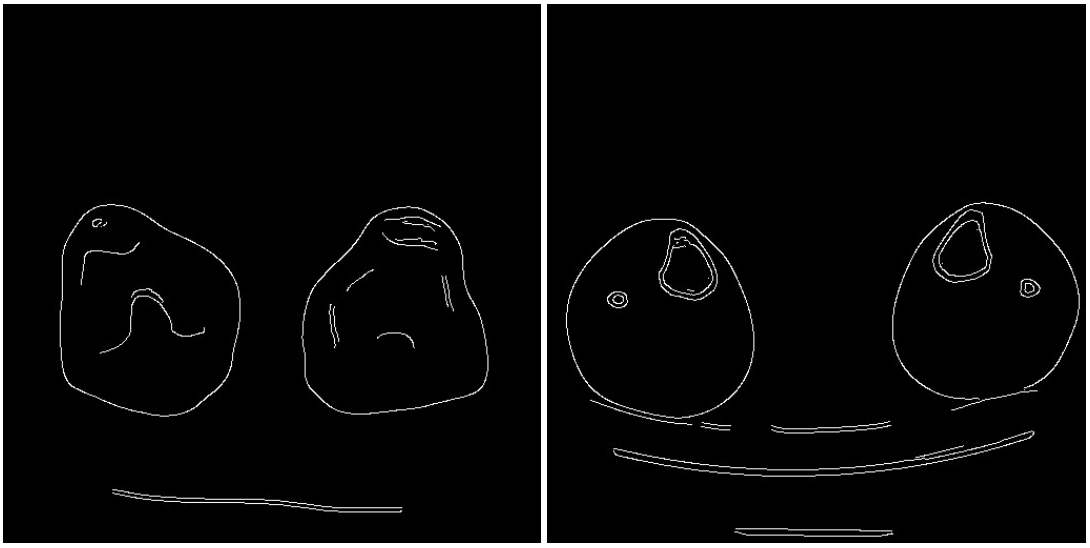
(a) GT (b) TAT

Figura 5.6: Filtrado bilateral: vecindario *15x15*, $\sigma = 300$



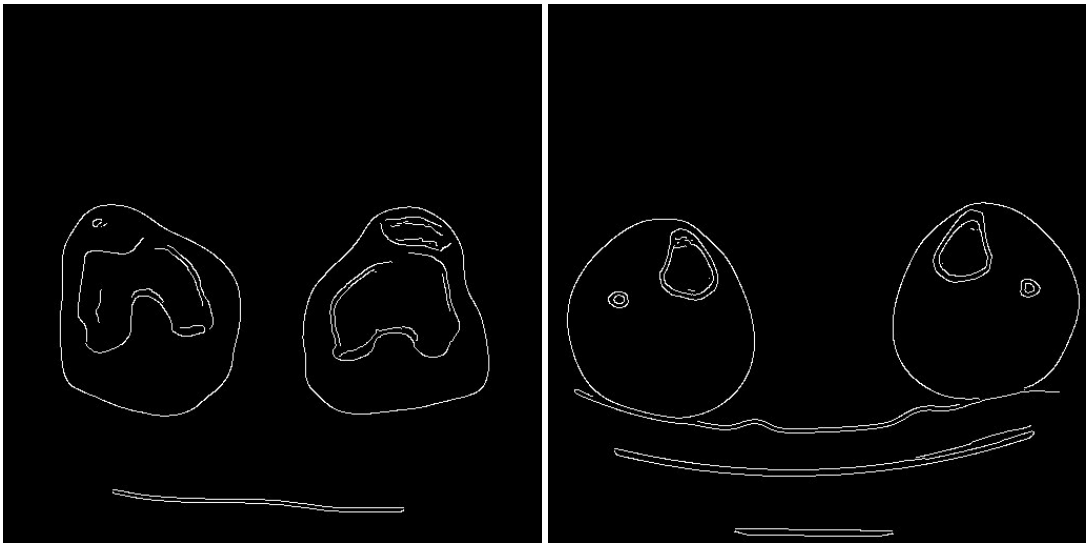
(a) GT (b) TAT

Figura 5.7: Detección de contornos: $\sigma = 1$, umbrales = 0,50, 0,30



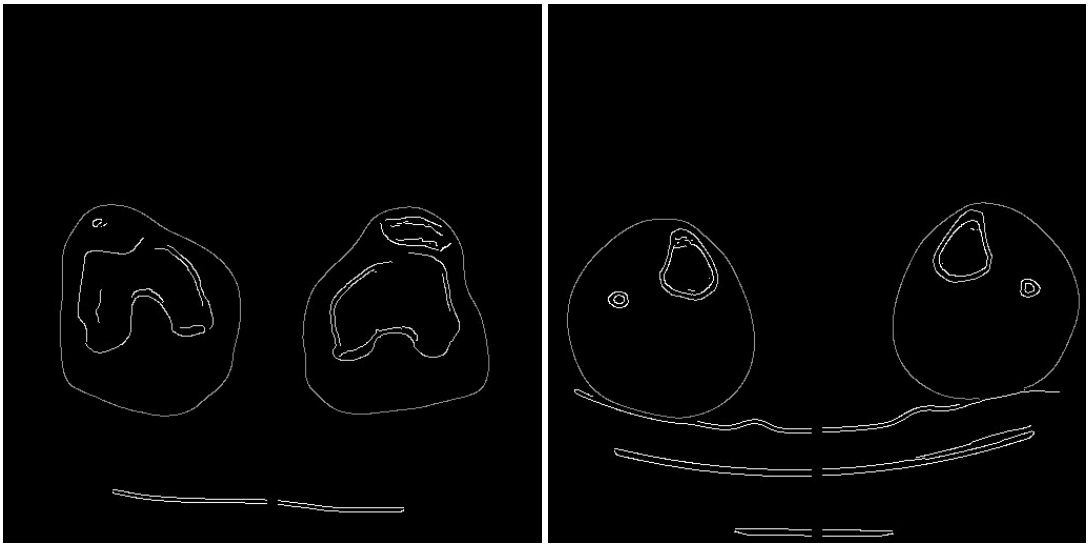
(a) GT (b) TAT

Figura 5.8: Detección de contornos: $\sigma = 3$, umbrales = 0,50, 0,30



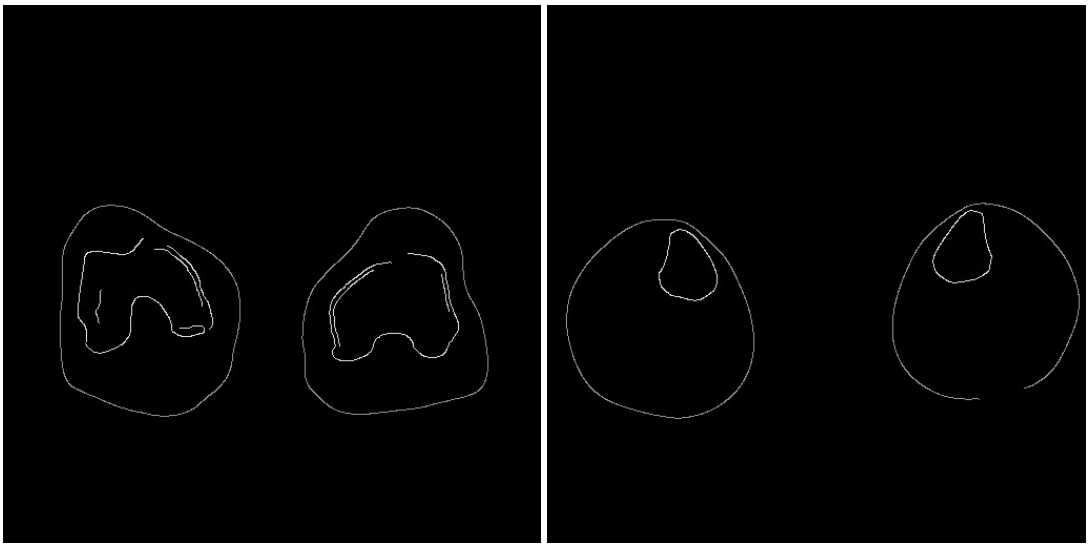
(a) GT (b) TAT

Figura 5.9: Detección de contornos: $\sigma = 3$, umbrales = 0,35, 0,15



(a) GT (b) TAT

Figura 5.10: Etiquetado



(a) GT (b) TAT

Figura 5.11: Supresión de bordes no deseados

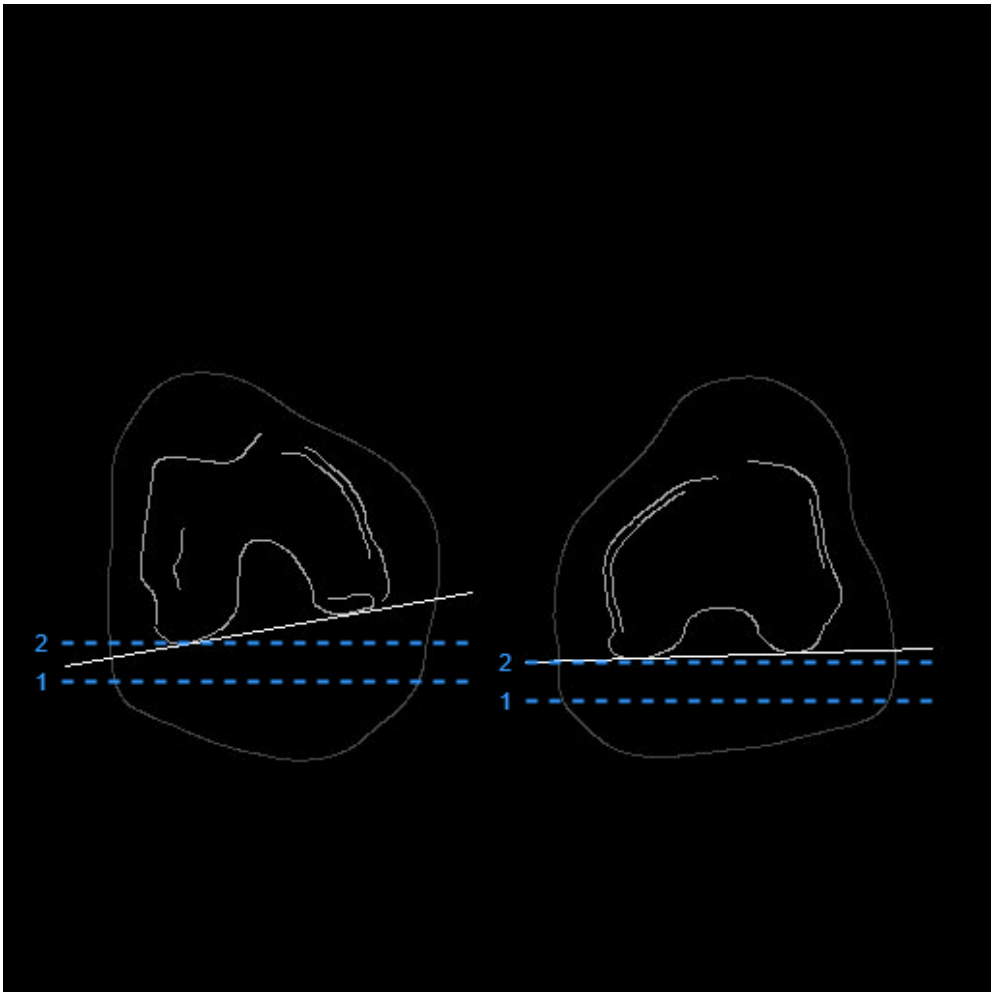


Figura 5.12: GT: Resultado del cálculo de la línea inferior

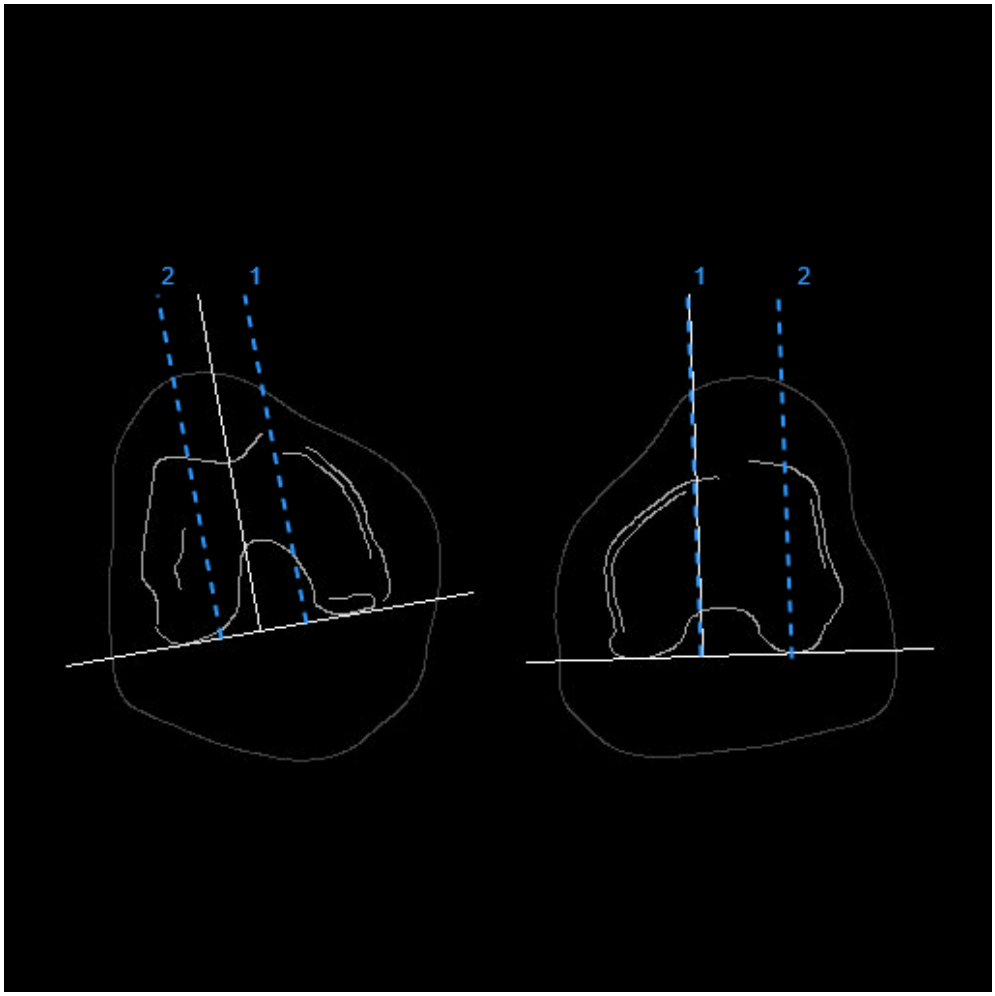


Figura 5.13: GT: Resultado del cálculo de la linea perpendicular

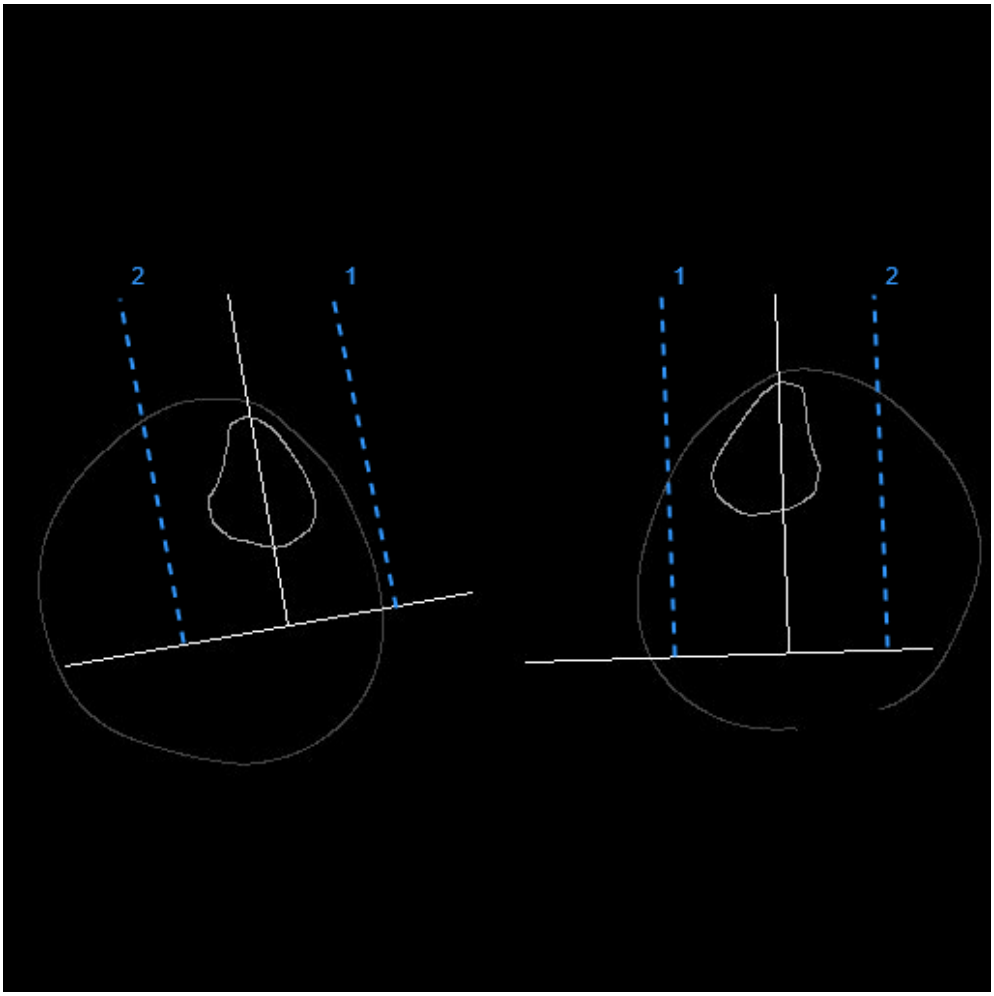
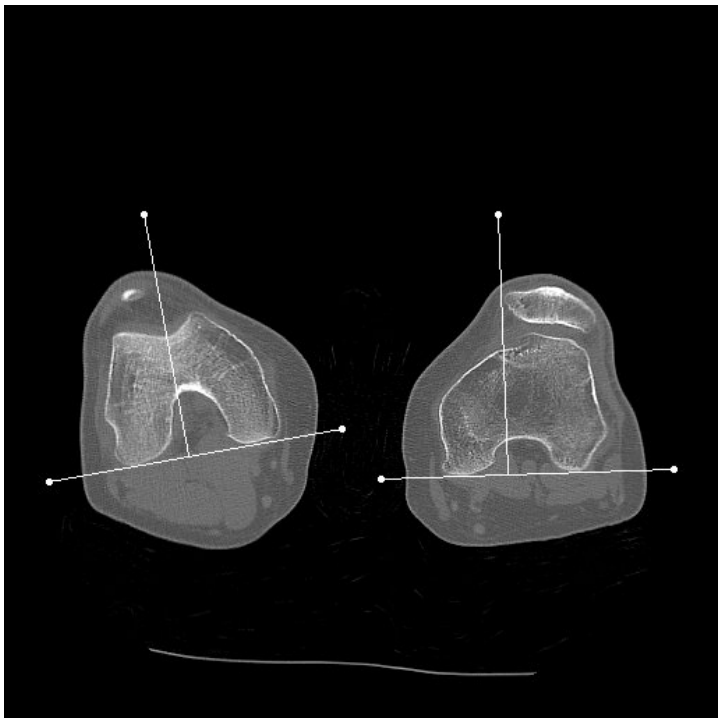
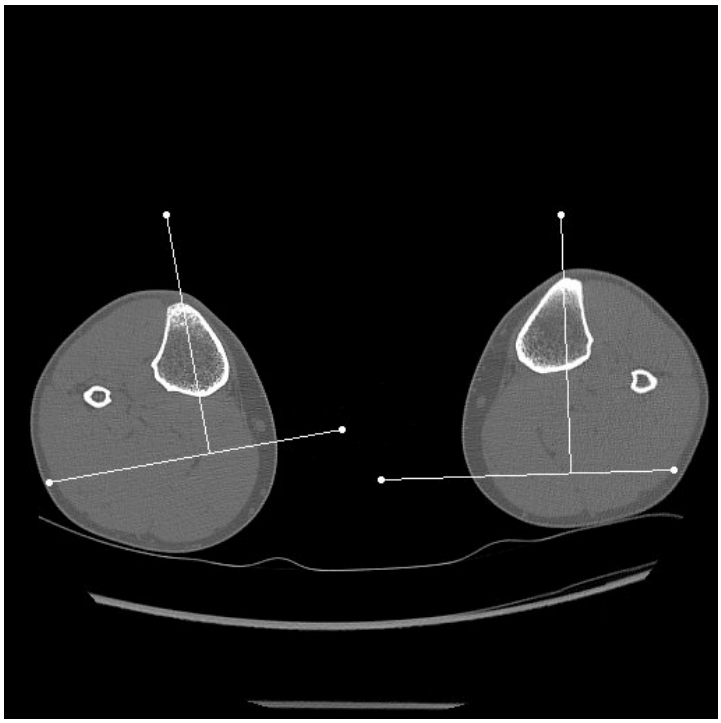


Figura 5.14: TAT: Resultado del cálculo de la línea perpendicular



(a) GT



(b) TAT

Figura 5.15: Medidas en imágenes originales

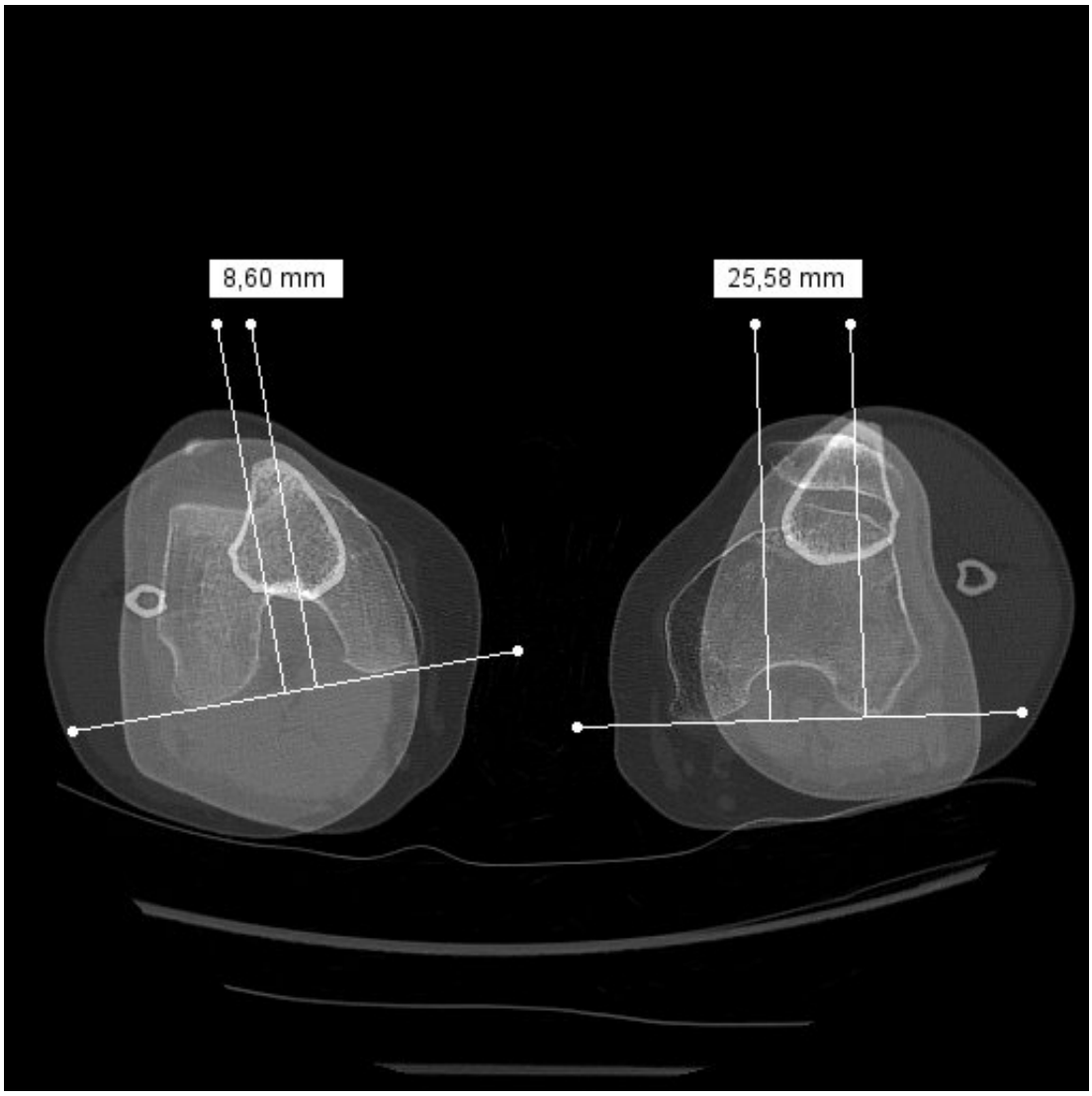
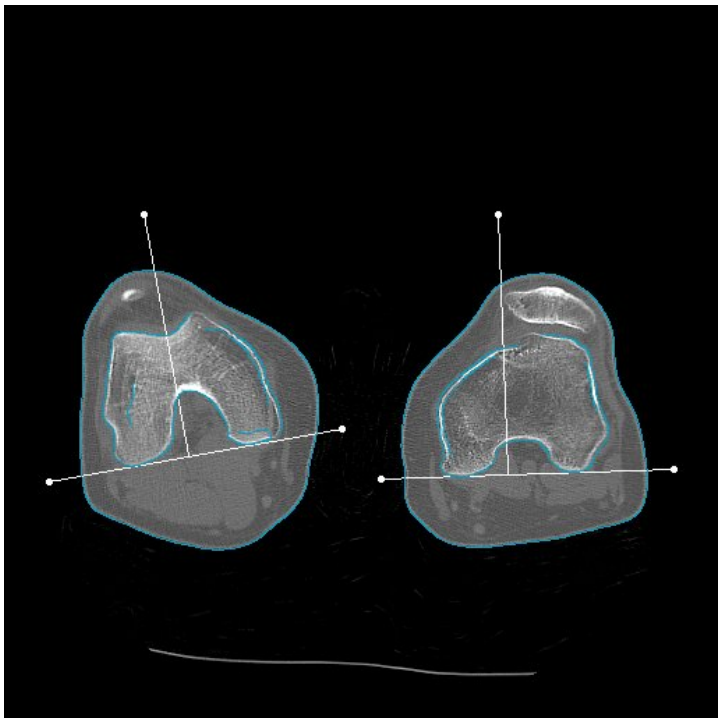
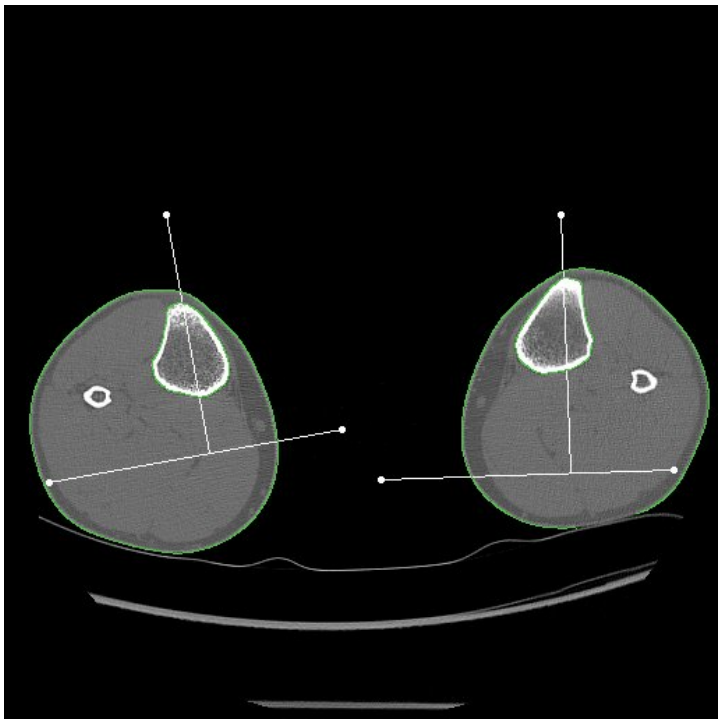


Figura 5.16: Medidas TA-GT en imágenes originales



(a) GT



(b) TAT

Figura 5.17: Medidas en imágenes con contornos superpuestos

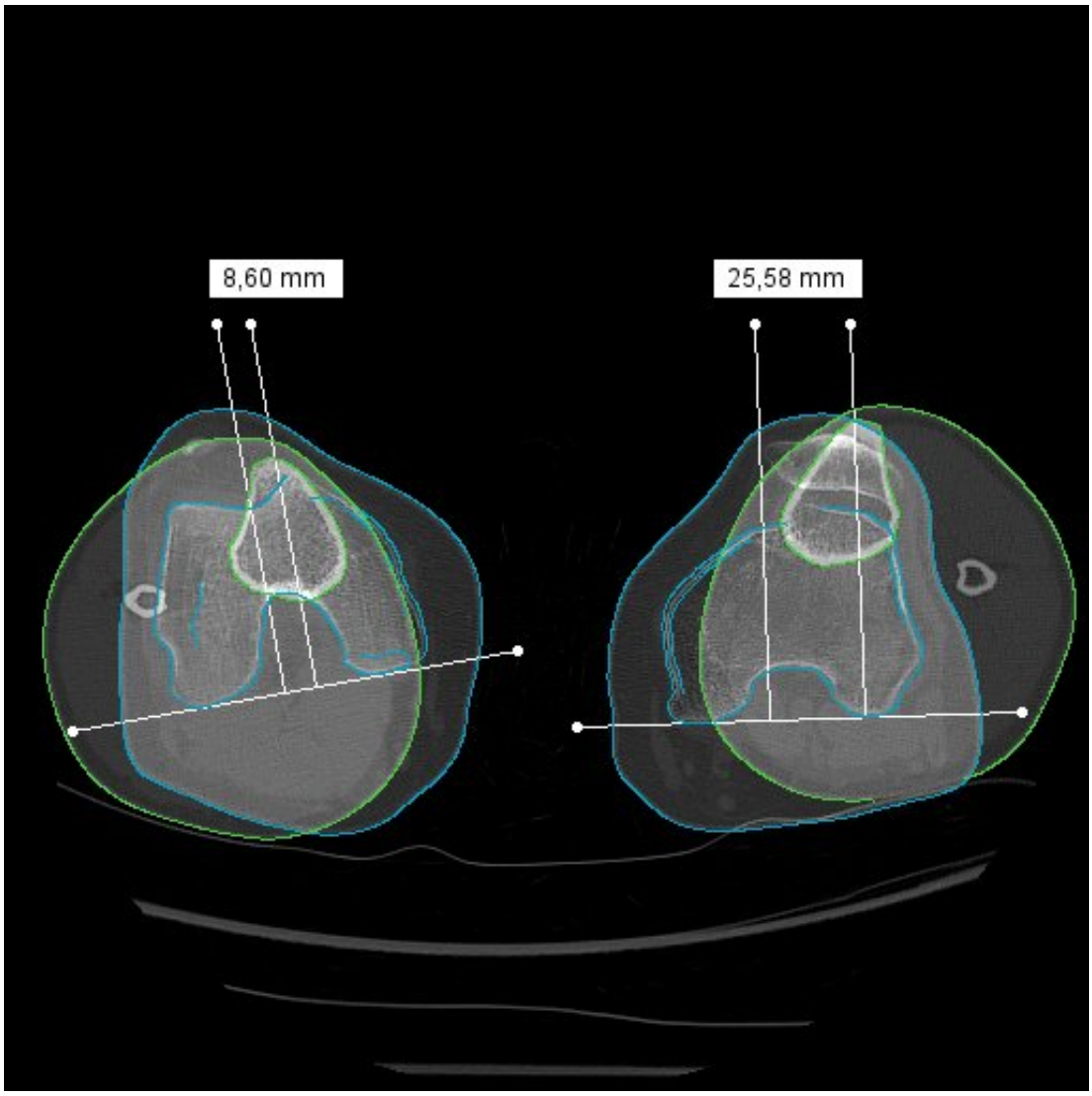
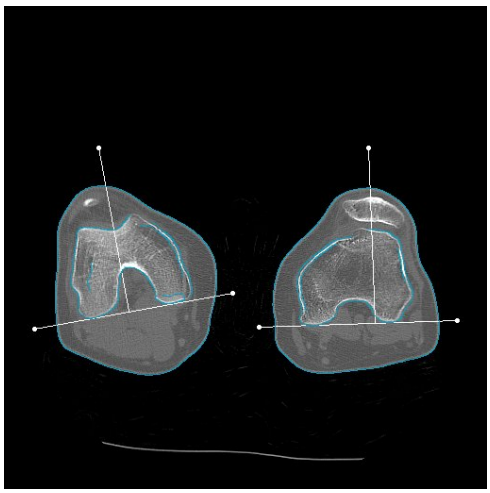
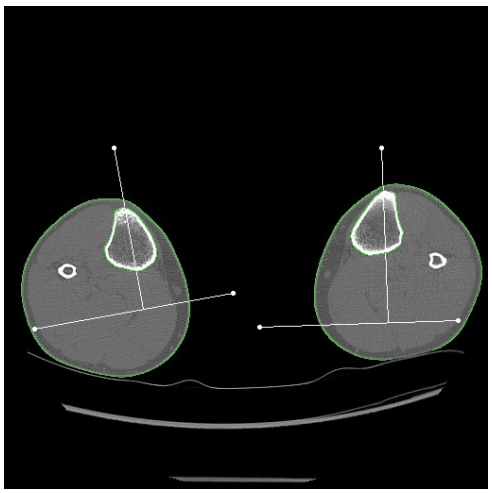


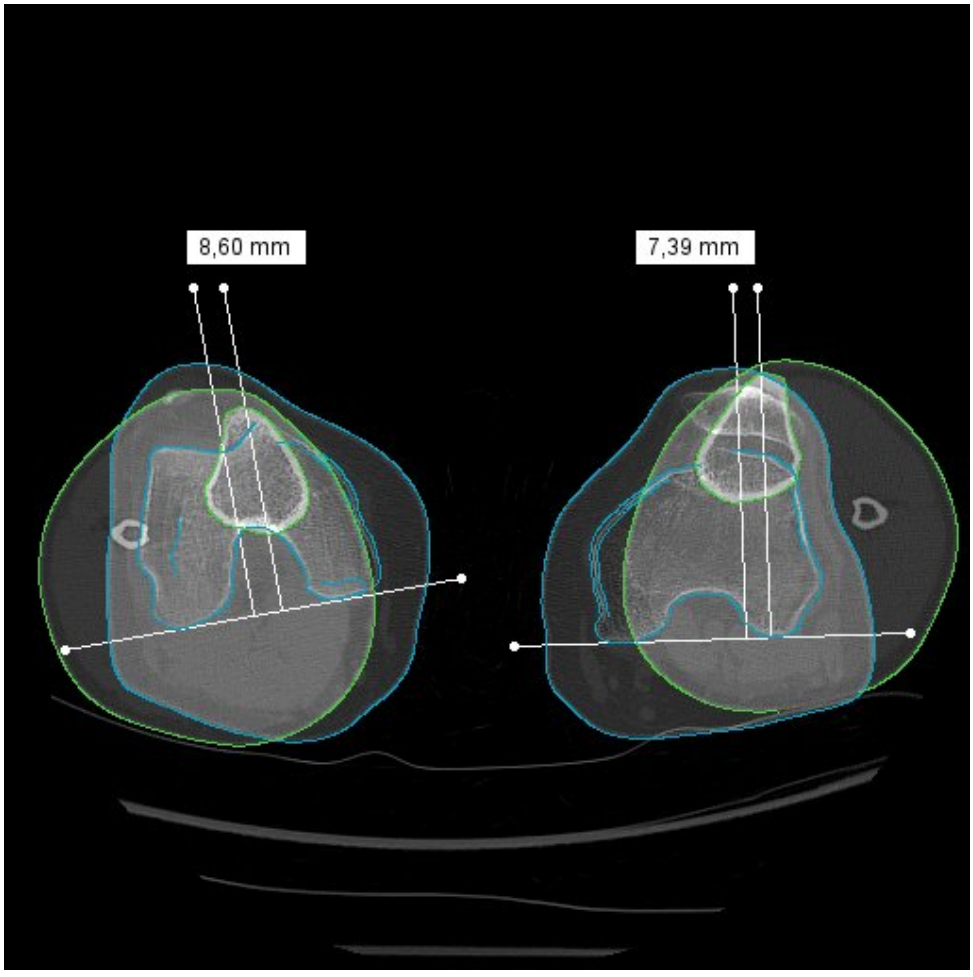
Figura 5.18: Medidas TA-GT en imágenes con contornos superpuestos



(a) GT



(b) TAT



(c) TA-GT

Figura 5.19: Medidas corregidas en imágenes con contornos superpuestos

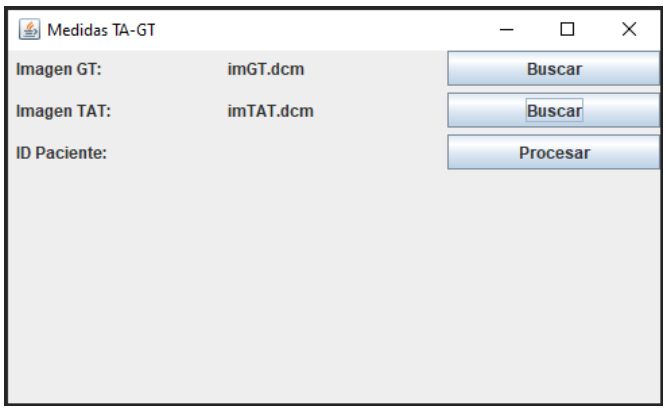


Figura 5.20: Interfaz de usuario: selección de imágenes

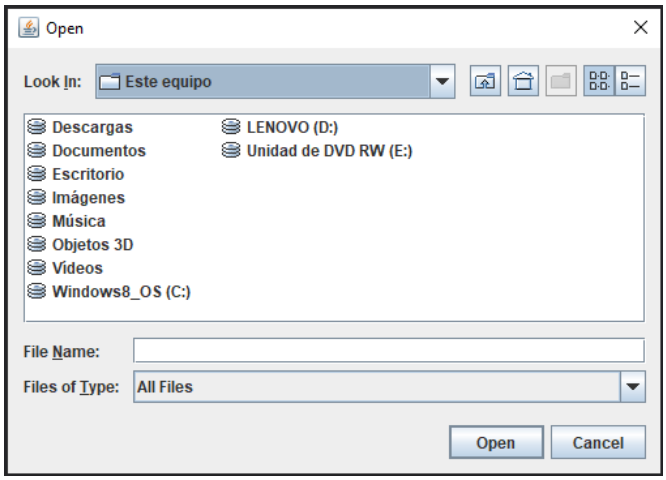


Figura 5.21: Interfaz de usuario: buscador de archivos

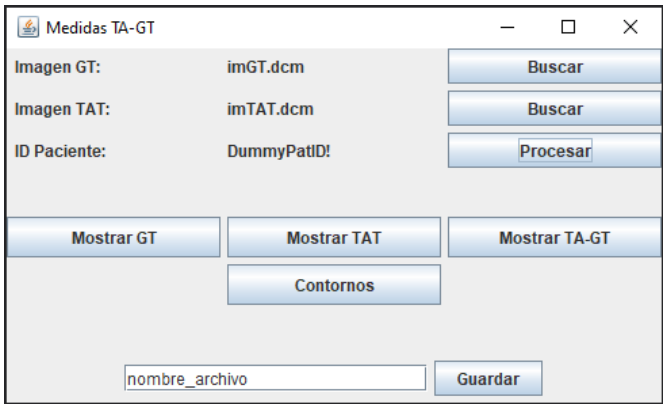
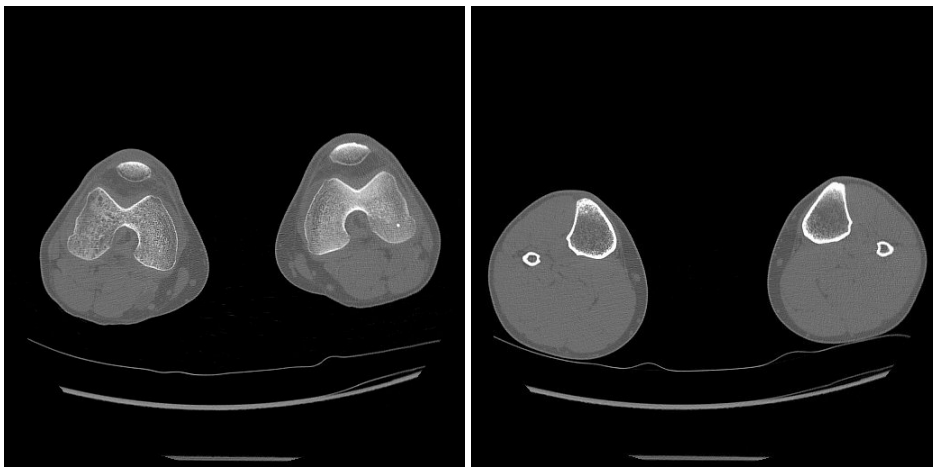


Figura 5.22: Interfaz de usuario: procesado de imagen



(a) GT (b) TAT

Figura 5.23: Imágenes de entrada 1

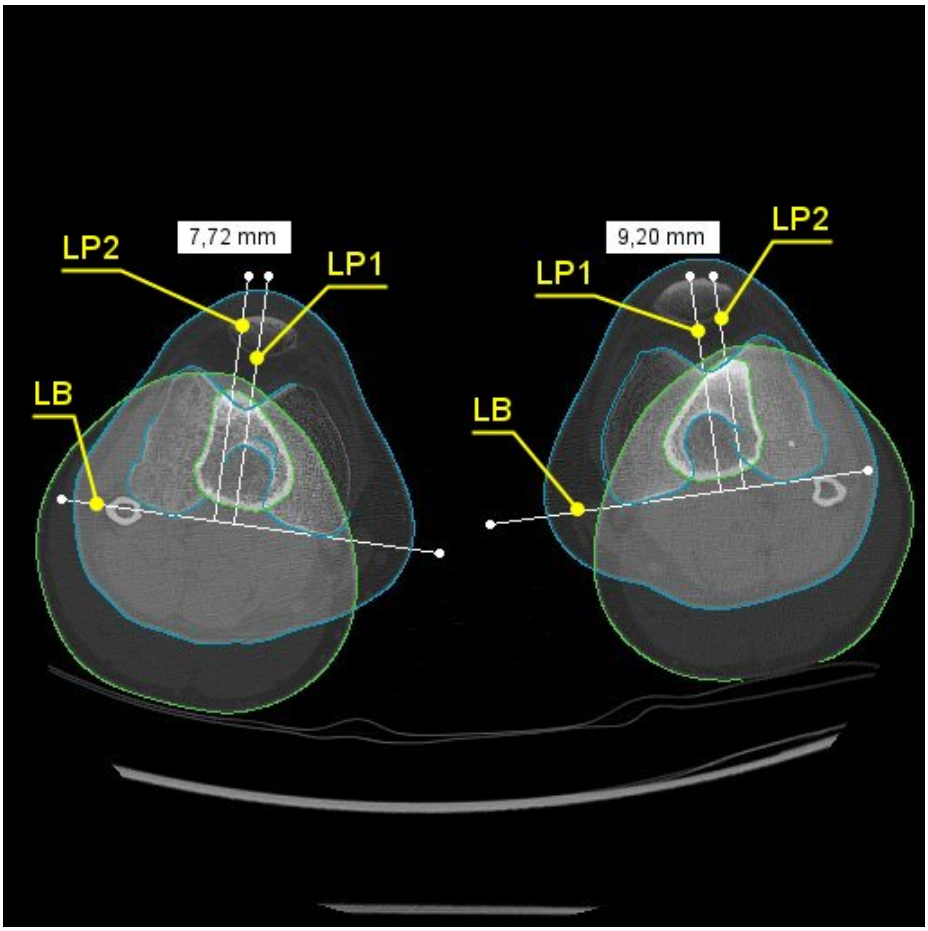
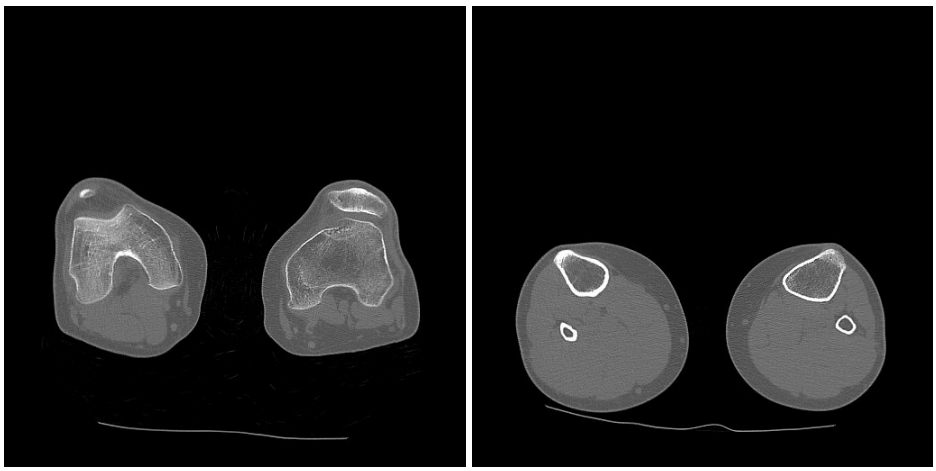


Figura 5.24: Medición TA-GT 1



(a) GT (b) TAT

Figura 5.25: Imágenes de entrada 2

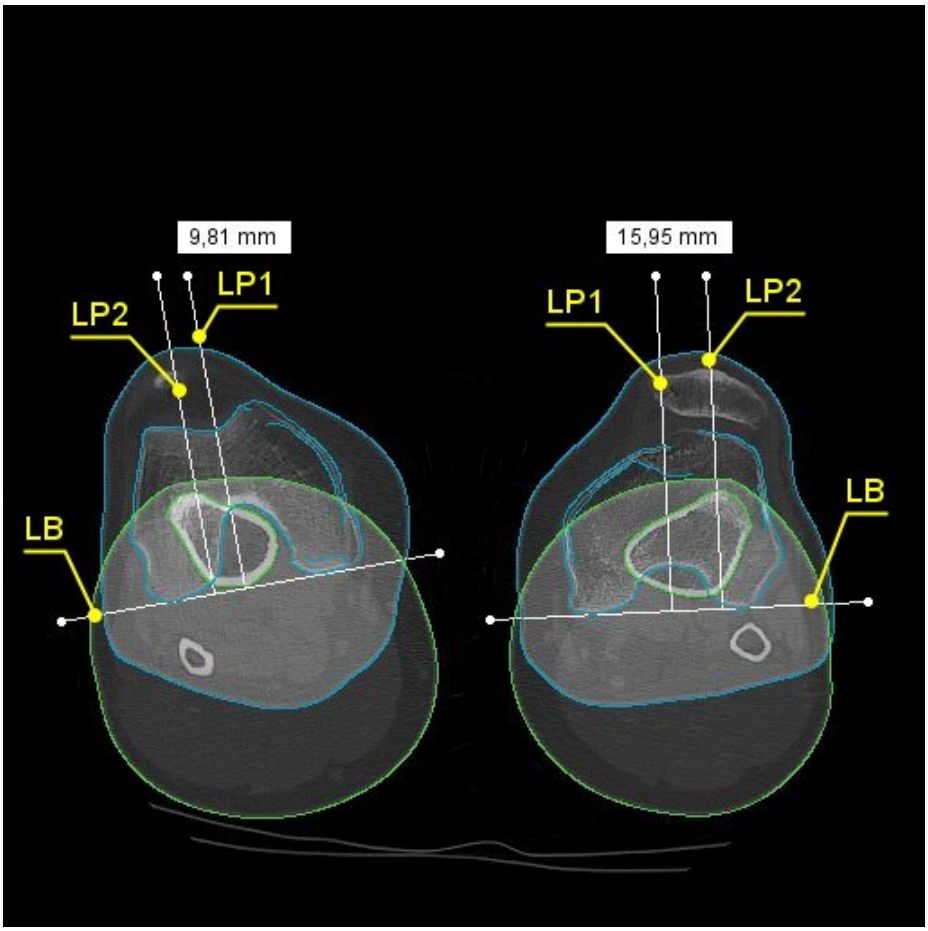
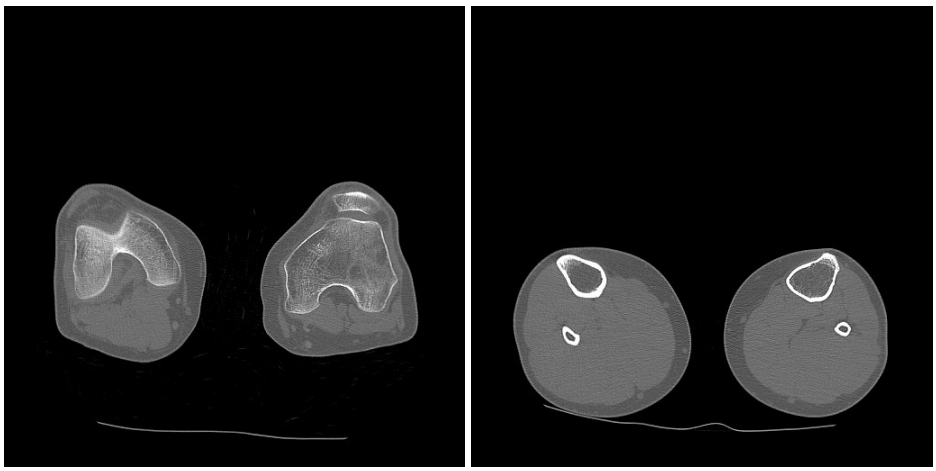


Figura 5.26: Medición TA-GT 2



(a) GT (b) TAT

Figura 5.27: Imágenes de entrada 3

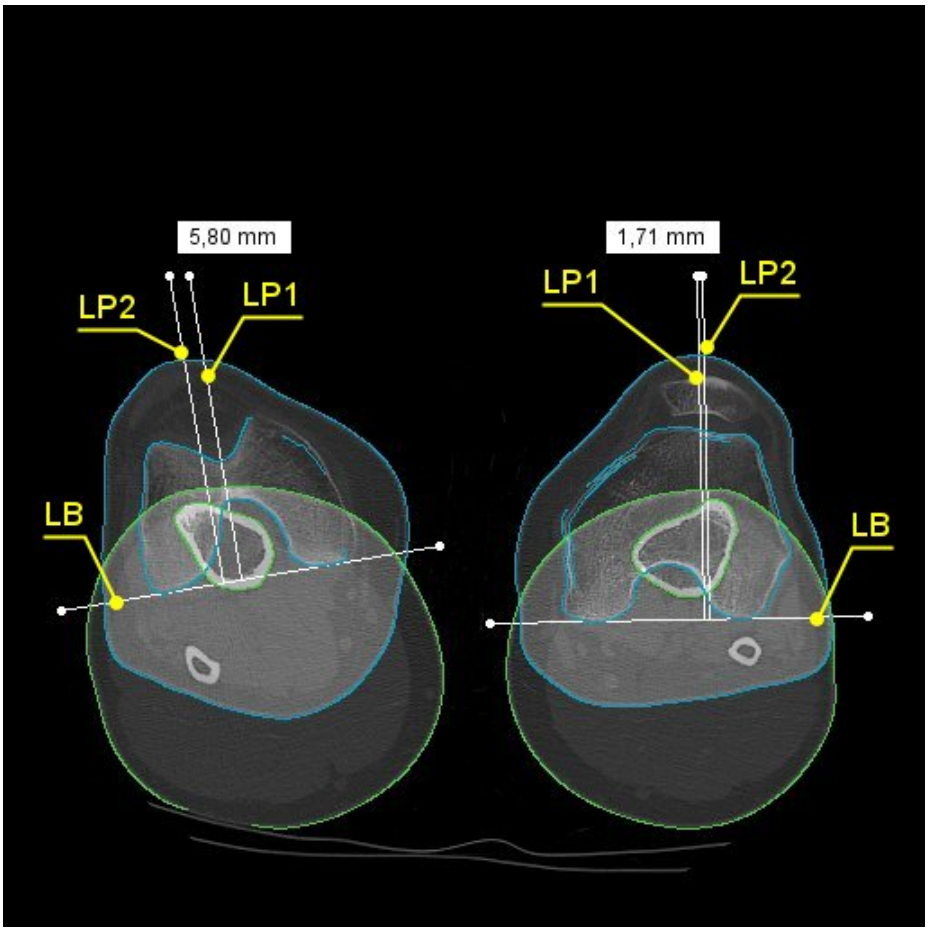
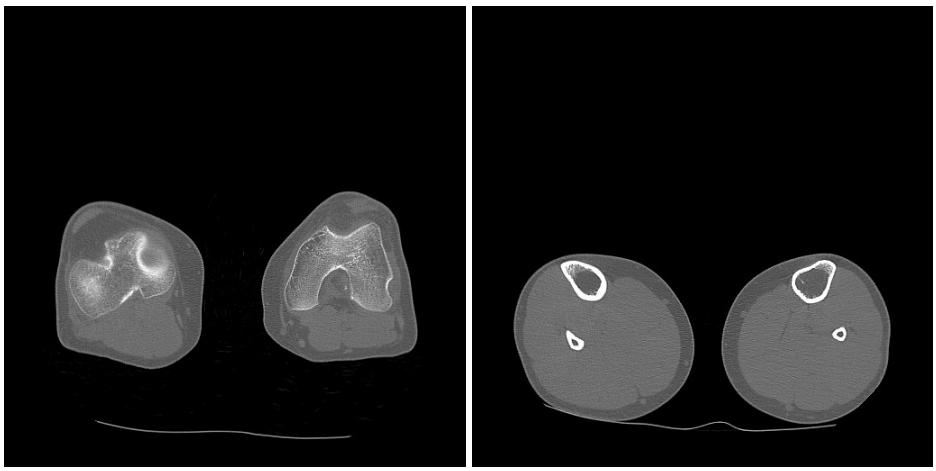


Figura 5.28: Medición TA-GT 3



(a) GT (b) TAT

Figura 5.29: Imágenes de entrada 4

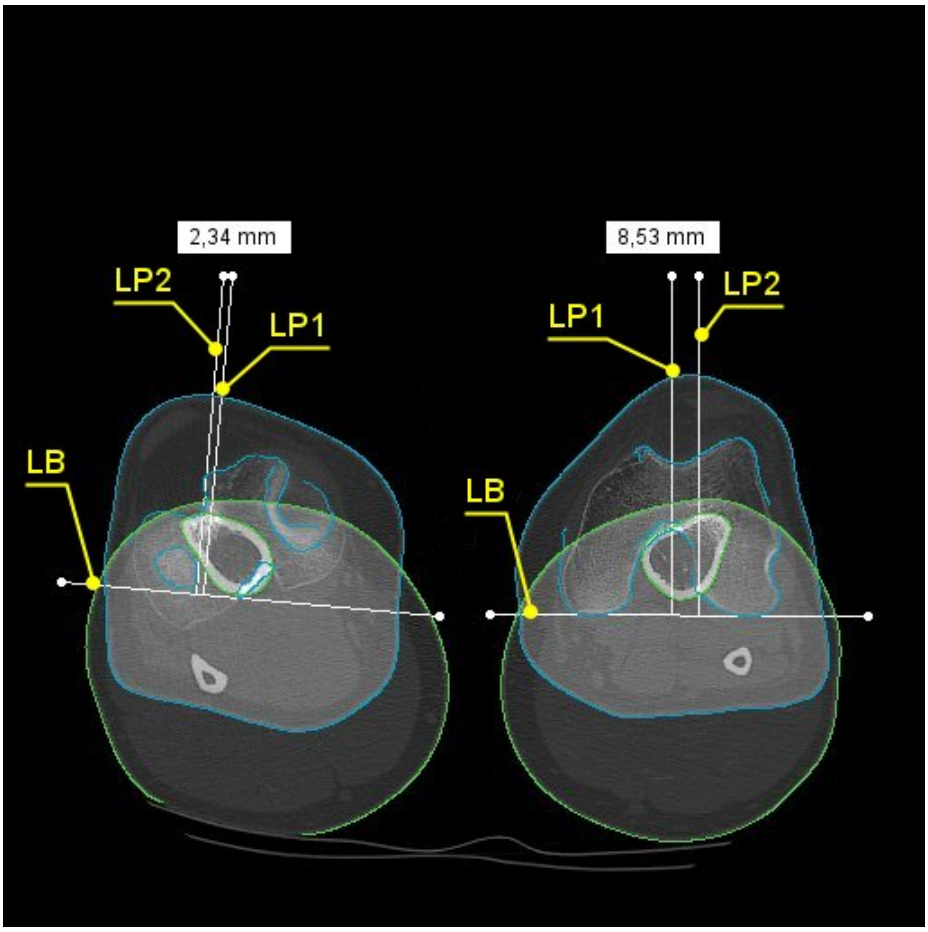
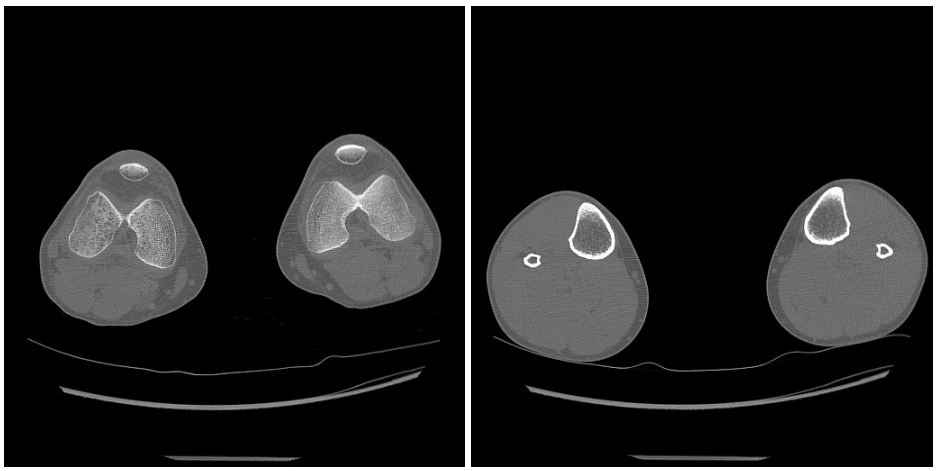


Figura 5.30: Medición TA-GT 4



(a) GT (b) TAT

Figura 5.31: Imágenes de entrada 5

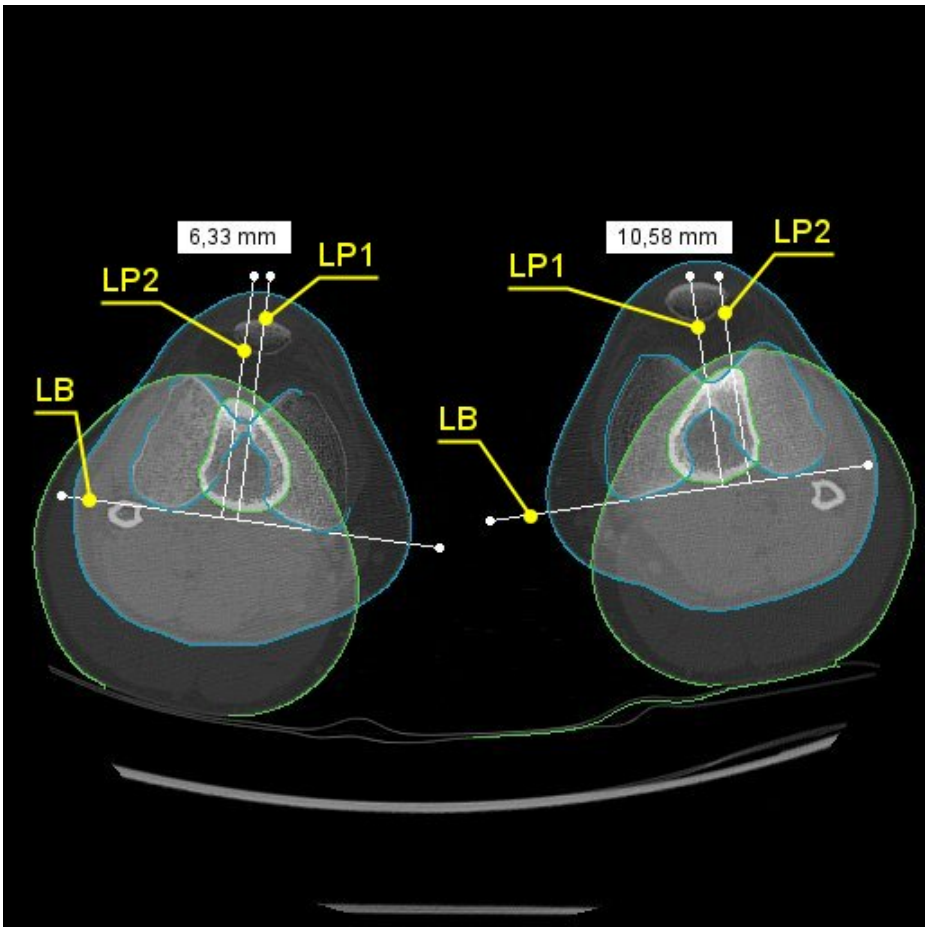
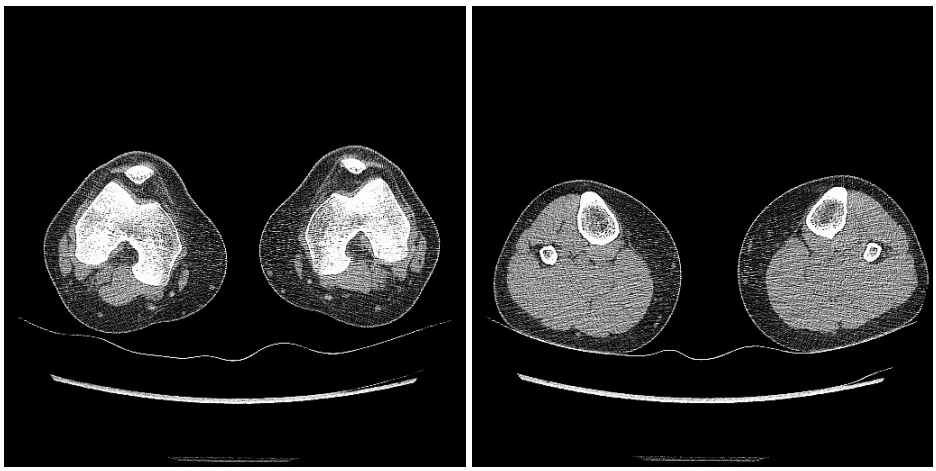


Figura 5.32: Medición TA-GT 5



(a) GT (b) TAT

Figura 5.33: Imágenes de entrada 6

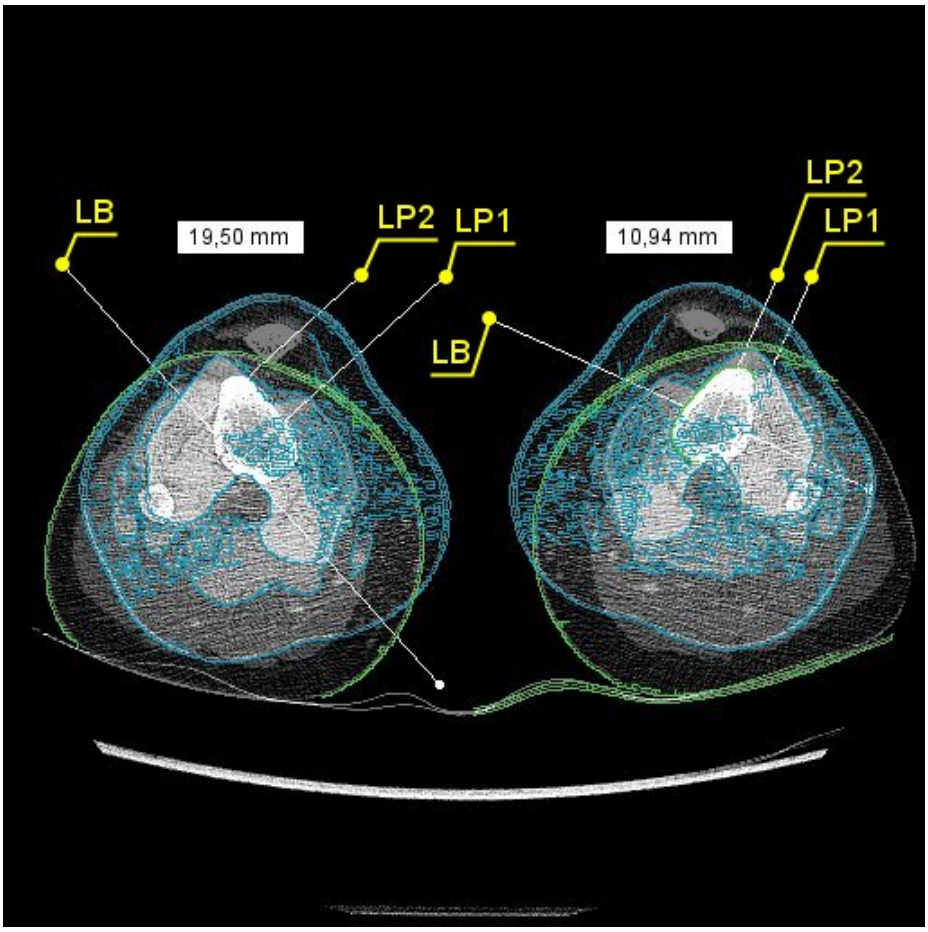


Figura 5.34: Medición TA-GT 6

Capítulo 6

Conclusiones y trabajo futuro

Último capítulo de la memoria en el se comentará el resultado final del sistema implementado y el trabajo futuro a realizar para mejorar algunas de sus funcionalidades.

6.1 Conclusiones

El objetivo de este trabajo era el de desarrollar un sistema que permitiera realizar un procesado sobre imágenes de un TAC de una rodilla. El resultado debía mostrar las mediciones correspondientes a la Tuberosidad Tibial Anterior - Garganta de la Tróclea sobre las dos imágenes de entrada de forma automatizada, pudiendo estas ser ajustadas por el usuario del programa.

El trabajo concluye con un sistema que proporciona una interfaz gráfica que permite al usuario escoger las imágenes sobre las cuales se quiere operar, realiza sobre ellas un procesado digital de forma automatizada y, por último, ofrece diferentes opciones de visualización, ajuste de forma manual y almacenamiento de los resultados.

La precisión en las mediciones automatizadas es la esperada, ofreciendo un porcentaje elevado de acierto en la localización de los puntos de interés y en el cálculo de las medidas. Esta precisión se vería afectada por la elección de las imágenes de entrada, ya que es el usuario el que debe escoger las dos imágenes en las que mejor se aprecie el surco troclear y el punto más saliente de la tibia. Además, nos encontramos con que los resultados del procesado en imágenes con mucho ruido se ven resentidos, siendo el usuario quien tiene que ajustar la medición de forma manual.

6.2 Trabajo Futuro

Para continuar ampliando la funcionalidad del sistema implementado en un futuro, el trabajo se debería centrar en dos aspectos: la mejora de la precisión y la inclusión de mecanismos

de Inteligencia Artificial para la elección de las imágenes de entrada.

La mejora de precisión del sistema se debería centrar principalmente en imágenes con mucho ruido, ya que la detección de contornos realizada no es la correcta. Para mejorar este aspecto habría que revisar, primero, la etapa de filtrado, de forma que analice los resultados y aumente el grado de suavizado o el método utilizado cuando fuese necesario. Como segundo paso, la detección de contornos tendría que ser revisada para los casos en los que la imagen de entrada tenía una fuerte distorsión.

En cuanto a la inclusión de una Inteligencia Artificial, el objetivo sería escoger las imágenes de entrada de forma automatizada. Este sistema recibiría todas las imágenes de un estudio en formato DICOM, escogiendo de forma automatizada aquella que mejor muestre el surco trocleano y aquella en la que se aprecie el margen más saliente de la tibia.

En este caso, el sistema debería ser entrenado con un conjunto amplio de estudios de imágenes, del cual no se disponía durante el desarrollo de este trabajo.

Apéndices

Lista de acrónimos

ACR American College of Radiology.

DICOM Digital Imaging and Communications in Medicine.

GT Garganta de la Tróclea.

NEMA National Electrical Manufacturers Association.

NIH National Institutes of Health.

TA-GT Tuberosidad Tibial Anterior - Garganta de la Tróclea.

TAC Tomografía Axial Computerizada.

TAT Tuberosidad Anterior de la Tibia.

TCP/IP Transmission Control Protocol/ Internet Protocol.

Glosario

Algoritmo. Conjunto ordenado de operaciones sistemáticas utilizado para realizar un cálculo y hallar la solución de algún problema.

Bit. En informática, unidad mínima de información que puede tener solo dos valores (cero y uno).

Contraste. Diferencia entre dos valores de intensidad en la escala de grises o en la de colores de una imagen.

Cóndilo. Elemento redondeado en la extremidad del hueso que forma la articulación al encajar con el hueco correspondiente de otro hueso.

Decúbito supino. Posición corporal acostado boca arriba, con cuello en posición neutra, extremidades superiores extendidas y pegadas al tronco con las palmas de las manos hacia arriba y extremidades inferiores extendidas con las puntas de los dedos hacia arriba.

Derivada. En una función es el límite hacia el cual tiende la razón entre el incremento en el valor de la función y el de la variable cuando el incremento tiende a cero. Es igual a la pendiente de la recta tangente a la gráfica de la función.

Epífisis. Cada uno de los extremos ensanchados de los huesos largos, situados a ambos lados de la parte larga central.

Función. En matemáticas, relación entre dos magnitudes por la que a cada valor de una de ellas le corresponde un valor determinado de la otra.

Gradiente. Variación de una magnitud en función de la distancia. Es una generalización de la derivada para funciones de varias variables que, al igual que esta, representa la pendiente de la recta tangente a la gráfica de una función.

Interpolación. Obtención de nuevos puntos en una imagen partiendo de un conjunto de puntos existente.

Pixel. Unidad básica de una imagen digital mostrada por pantalla como un conjunto de puntos de color o en escala de grises.

Plugin. Pequeño programa complementario que amplia las funciones de otra aplicación web o de escritorio.

Resonancia magnética. Técnica de imágenes médicas que utiliza un campo magnético y ondas de radio generadas por computadora para crear imágenes detalladas de los órganos y tejidos del cuerpo.

Sensor. Dispositivo que capta magnitudes físicas (como pueden ser los niveles de luz) en su entorno.

Tomografía Axial Computerizada. Procedimiento para el cual se utiliza una computadora conectada a una máquina de rayos X a fin de crear una serie de imágenes detalladas del interior del cuerpo.

Transformada de Fourier. Algoritmo matemático empleado para transformar señales entre el dominio del tiempo y el dominio de la frecuencia.

Tróclea. Articulación en forma de polea, que permite que un hueso adyacente pueda girar en el mismo plano.

Umbral. Valor mínimo del nivel de gris necesario para que algo sea perceptible.

Bibliografía

[1] L. Jullier, *La Imagen Digital: De la Tecnología a la Estética.* La Marca, 2004.

[2] M. Gómez, "Historia(s) de la imagen digital." [En línea]. Disponible en: https://interartive.org/2017/04/historias-de-la-imagen-digital-marisa-gomez

[3] "Reconocimiento facial en apps." [En línea]. Disponible en: https://baturamobile.com/blog/reconocimiento-facial-apps-procesamiento-imagen-2/

[4] "Los organismos encargados de la aplicación de la ley utilizan el reconocimiento facial informatizado, una tecnología relativamente nueva, para identificar a personas de interés para una investigación." [En línea]. Disponible en: https://www.interpol.int/es/Como-trabajamos/Policia-cientifica/Reconocimiento-facial

[5] Z. Mohammadzadeh, R. Safdari, M. Ghazisaeidi, S. Davoodi, and Z. Azadmanjir, "Advances in optimal detection of cancer by image processing; experience with lung and breast cancers," 2015. [En línea]. Disponible en: https://pubmed.ncbi.nlm.nih.gov/26320425/

[6] "Google ai for breast cancer detection beats doctors," 2020. [En línea]. Disponible en: https://towardsdatascience.com/google-ai-for-breast-cancer-detection-beats-doctors-65b8983352e0

[7] "Digital image processing basics," 2018. [En línea]. Disponible en: https://www.geeksforgeeks.org/digital-image-processing-basics/

[8] R. González, *Tratamiento Digital de Imágenes.* Addison-Wesley, 1996.

[9] M. Sonka, *Image Processing, Analysis and Machine Vision.* PWS Publishing, 1999.

[10] "Bilateral filtering for gray and color images." [En línea]. Disponible en: https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/MANDUCHI1/Bilateral_Filtering.html

[11] J. Valverde-Rebaza, "Detección de bordes mediante el algoritmo de canny," 2007. [En línea]. Disponible en: https://www.researchgate.net/publication/267240432_Deteccion_de_bordes_mediante_el_algoritmo_de_Canny

[12] "Canny edge detection step by step in python - computer vision," 2019. [En línea]. Disponible en: https://towardsdatascience.com/canny-edge-detection-step-by-step-in-python-computer-vision-b49c3a2d8123

[13] "Mediciones básicas de rodilla en tac," 2013. [En línea]. Disponible en: https://radiologiahcm.wordpress.com/2013/03/06/mediciones-basicas-de-rodilla-en-tac/

[14] "Dicom standard." [En línea]. Disponible en: https://www.dicomstandard.org/

[15] "Java." [En línea]. Disponible en: https://www.java.com/

[16] A. Sarangam, "History of java," 2021. [En línea]. Disponible en: https://www.jigsawacademy.com/blogs/java/history-of-java/

[17] "Imagej." [En línea]. Disponible en: https://imagej.net/software/imagej/

[18] "Bilateral filter." [En línea]. Disponible en: http://bigwww.epfl.ch/algorithms/bilateral-filter/

[19] T. Gibara, "Canny edge detector," 2013. [En línea]. Disponible en: https://imagej.nih.gov/ij/plugins/canny/index.html