



UNIVERSIDAD DE LAS PALMAS DE GRAN CANARIA
Escuela de Ingeniería Informática



Grado en Ingeniería Informática

Trabajo Fin de Grado

Software de Control de Calidad para Medicina Nuclear

Chaxiraxi Cabrera González

Tutor: **José Daniel Hernández Sosa**

Tutor: **David Armando Macías Verde**

Las Palmas de Gran Canaria

11 de julio de 2017

“El hombre nunca sabe de lo que es capaz hasta que lo intenta.”
Charles Dickens.

Agradecimientos

Quiero agradecer este proyecto a los compañeros del Hospital Negrín por darme la oportunidad de trabajar con ellos, ayudarme y asesorarme. También quiero agradecer a mi tutor académico por guiarme en el desarrollo de este proyecto. Para finalizar quiero agradecer a mi familia y mi pareja por su apoyo incondicional durante todos mis años de estudio. Sin todos ellos este proyecto no hubiera sido posible.

Resumen

En este proyecto se implementa una aplicación capaz de realizar las pruebas de control de calidad de las gammacámaras planares Siemens Symbia T y Ecam, de forma independiente a las herramientas facilitadas por la casa comercial Siemens para así corroborar su validez. Estas pruebas serán cuatro: Tamaño de Pixel, Uniformidad Intrínseca, Resolución Espacial Extrínseca y Uniformidad Tomográfica, y se encuentran recogidas en el documento español de control de calidad de instrumentación de Medicina Nuclear publicado en 2015. Para realizar la comprobación de dichas pruebas se utilizan técnicas de procesamiento de imágenes médicas, en concreto para imágenes en formato DICOM. La aplicación además integra algunas herramientas de ayuda para el usuario y esta implementada en MATLAB.

Índice general

1. Introducción	1
1.1. Estructura del documento	3
2. Estado actual del tema	5
2.1. Introducción al procesado de la imagen médica digital	5
2.2. Imágenes en medicina nuclear	6
2.2.1. Calidad de las imágenes	6
2.3. Procesado y realzado de imágenes médicas	7
2.3.1. Operaciones en el dominio espacial	7
2.3.2. Operaciones en el dominio de la frecuencia	10
2.4. Segmentación de imágenes médicas	10
2.4.1. Métodos basados en bordes	11
2.4.2. Métodos basados en píxeles	12
2.4.3. Métodos basados en regiones	13
3. Motivación y objetivos	15
3.1. Motivación	15
3.2. Objetivos principales	15
3.3. Competencias específicas cubiertas	16
4. Recursos utilizados	17
4.1. Hardware	17
4.1.1. Siemens Symbia T	17
4.1.2. E.Cam	17
4.1.3. Portátil	17
4.2. Software	18
4.2.1. MATLAB R2015a	18
4.2.2. Microsoft EXCEL	18
4.3. Librerías	18
4.4. Otras herramientas utilizadas	19
4.4.1. LATEX	19

4.4.2. StarUML	19
5. Análisis y Diseño	21
5.1. Plan de trabajo inicial	21
5.2. Diseño	22
5.3. Metodología	23
5.4. Estructura de la aplicación	24
6. Implementación de las pruebas	29
6.1. Descripción de funciones MATLAB utilizadas	29
6.1.1. Funciones internas de MATLAB	29
6.1.2. Procesamiento de imágenes	29
6.1.3. Ajuste de curvas	33
6.1.4. Importar y exportar datos	33
6.2. Prueba 1. Tamaño De Pixel	34
6.2.1. Objetivo de la prueba	34
6.2.2. Explicación de la prueba	34
6.2.3. Implementación de la prueba	35
6.2.4. Algoritmo	40
6.3. Prueba 2. Uniformidad Intrínseca	41
6.3.1. Objetivo de la prueba	41
6.3.2. Explicación de la prueba	41
6.3.3. Implementación de la prueba	44
6.3.4. Algoritmo	47
6.4. Prueba 3. Resolución Espacial Extrínseca	49
6.4.1. Objetivo de la prueba	49
6.4.2. Explicación de la prueba	49
6.4.3. Implementación de la prueba	52
6.4.4. Algoritmo	55
6.5. Prueba4. Uniformidad Tomográfica	57
6.5.1. Objetivo de la prueba	57
6.5.2. Explicación de la prueba	57
6.5.3. Implementación de la prueba	58
6.5.4. Algoritmo	60
7. Implementación de la interfaz	63
7.1. Interfaz Herramientas	64
7.2. Interfaz Pruebas	68

ÍNDICE GENERAL

8. Resultados	71
8.1. Tamaño de Píxel	71
8.2. Uniformidad Intrínseca	72
8.3. Resolución Espacial	72
8.4. Uniformidad Tomográfica	73
8.5. Resultados académicos	73
9. Conclusiones	75
9.1. Aportaciones	75
9.2. Trabajo Futuro	76
A. Manual de Usuario	77
A.1. Iniciar aplicación	77
A.2. Herramientas	77
A.2.1. Conversor de Nombres	77
A.2.2. Históricos	78
A.3. Pruebas	78
A.3.1. Tamaño De Píxel	78
A.3.2. Uniformidad Intrínseca	78
A.3.3. Resolución Espacial	78
A.3.4. Uniformidad Tomográfica	78
B. Versiones anteriores	81

ÍNDICE GENERAL

Índice de figuras

1.1. Siemens Symbia T	2
2.1. Máscaras de media y de media ponderada [8]	9
2.2. Otras máscaras[8]	9
5.1. Cuadro informativo de la planificación	22
5.2. Diagrama Aplicación	24
5.3. Diagrama Conversor de Imágenes	25
5.4. Diagrama Históricos	26
5.5. Diagrama Pruebas: Tamaño de Píxel y Resolución Espacial	27
5.6. Diagrama Pruebas: Uniformidad	28
6.1. Metadatos DICOM	30
6.2. Imagen Tamaño Píxel	34
6.3. Interfaz Tamaño Píxel	36
6.4. Interfaz Tamaño Píxel	37
6.5. Interfaz Tamaño Píxel	38
6.6. Interfaz Tamaño Pixel-Cargar Imágenes Error 1	38
6.7. Interfaz Tamaño Pixel-Cargar Imágenes Error 2	39
6.8. Interfaz Tamaño Pixel-Exportar	39
6.9. Excel - TamañoPixel	39
6.10. Imágenes Adquiridas	40
6.11. Colocación de la fuente respecto de la superficie del detector.[2]	42
6.12. Imagen de Uniformidad Intrínseca	42
6.13. Filtro [2]	43
6.14. Interfaz	45
6.15. Interfaz	46
6.16. EXCEL UniformidadIntrinseca	46
6.17. Imágenes Adquiridas 1024x1024	47
6.18. Imágenes Adquiridas 64x64	48

ÍNDICE DE FIGURAS

6.19. Colocación de la fuente lineal respecto de la superficie del detector.[2]	50
6.20. Imagen de Resolución Espacial	50
6.21. Análisis de la imagen: ROIs y perfiles.[2]	51
6.22. Perfil de las cuentas [2]	52
6.23. Interfaz	53
6.24. Perfil de las cuentas	54
6.25. Interfaz	54
6.26. Excel ResolucionEspacial	55
6.27. Imagen Binaria	56
6.28. Imagen ROI	56
6.29. Imagen Uniformidad Tomográfica.	57
6.30. Interfaz Uniformidad Tomográfica.	58
6.31. Histograma Uniformidad Tomográfica.	59
6.32. EXCEL Uniformidad Tomográfica	60
6.33. Imágenes Adquiridas 128x128	61
7.1. Interfaz Principal	63
7.2. Interfaz Herramientas	64
7.3. Interfaz Cambio Nombre.	65
7.4. Interfaz Históricos	66
7.5. Históricos Tamaño Pixel	66
7.6. Histórico Uniformidad Intrínseca	67
7.7. Histórico Uniformidad Tomográfica	68
7.8. Interfaz Pruebas	69
8.1. Resultados Tamaño Píxel	71
8.2. Resultados Uniformidad Intrínseca	72
8.3. Resultados Resolución Espacial	73
A.1. Icono EXCEL	78
B.1. Versión 1 Tamaño de píxel	81
B.2. Versión 2	82
B.3. Versión 2 Resolución Espacial	83
B.4. Versión 1 Uniformidad Tomográfica	83

Índice de cuadros

6.1. Tabla Valores Especificados por el fabricante.[11]	44
6.2. Tabla Tolerancias	44

ÍNDICE DE CUADROS

Acrónimos

DICOM: Digital Imaging Communications in Medicine.

FFT: Fast Fourier Transform.

LSF: Line Spread Function.

MTF: Modulation Transfer Function.

PET: Positron emission tomography.

PSF: Point Spread Function.

ROI: Region of interest.

SPECT: Single photon emission computed tomography.

CTVU: Campo Total Visión Útil

UFOV: Useful Field of View

CCVU: Campo Central Visión Útil

CFOV: Central Field of View

FWHM: Full Width at Half Maximum.

AIMA: Ancho intrínseco a mitad de altura

FWTM: Full Width at Tenth Maximum

AIDA: Ancho intrínseco a un décimo de altura

Capítulo 1

Introducción

La medicina nuclear es una técnica de obtención de imágenes médicas que consiste en obtener imágenes de la concentración de isótopos radiactivos en el cuerpo humano. En la medicina nuclear se administran radiofármacos que contienen isótopos radioactivos al paciente para poder obtener imágenes de la distribución de estos a lo largo del cuerpo humano, los isótopos inestables producen rayos gamma que permiten medir la concentración de estos radiofármacos en función de la posición y del tiempo, por lo que es una modalidad de imagen fisiológica, es decir, muestran el funcionamiento de los órganos y los tejidos del cuerpo.

El principal modo de detección de los fotones es mediante el uso de un cristal de centelleo con tubos foto-multiplicadores adjuntos que aumentan la señal, estos cristales convierten la energía de la radiación ionizante en luz visible, ya que el número de fotones emitidos es proporcional a la energía depositada. De esta forma es posible medir la concentración de los radiofármacos en función de la posición y del tiempo de emisión radiactiva. [12]

El Servicio de Radiofísica Hospitalaria del Hospital Universitario de Gran Canaria Dr. Negrín necesitaba un software capaz de realizar las pruebas de control de calidad de las gammacámaras planares Siemens Symbia T y Ecam, de forma independiente a las herramientas facilitadas por la casa comercial Siemens para así corroborar su validez. Las pruebas implementadas en el proyecto se basan en el análisis de imágenes DICOM que se obtienen de ensayos hechos con maniquís de control de calidad específicos, según el real decreto 1841/1997 y el actual documento español de control de calidad de instrumentación de Medicina Nuclear [2], publicado en 2015 por las sociedades españolas de Física Médica, Protección Radiológica y de Medicina Nuclear e Imagen Molecular.

En el presente trabajo se consideran las siguientes pruebas de validación, recogidas en el documento de control de calidad de Medicina Nuclear [2]:

- **GPL12: Tamaño de Píxel:** Consiste en obtener el factor de escala entre las dimensiones del objeto en la imagen.
- **GPL05: Uniformidad intrínseca:** Obtiene la diferencia entre el comportamiento ideal y el real del detector.
- **GPL10: Resolución espacial extrínseca:** Estima la capacidad del equipo de distinguir entre dos fuentes radiactivas pequeñas y cercanas.
- **GTM03: Uniformidad Tomográfica:** Determina la uniformidad de la reconstrucción de un maniquí con una distribución homogénea de actividad.

Las pruebas de calidad se realizan con el fin de indicar al usuario si el equipo examinado presenta fallos, es importante comprobar que todo el instrumental funciona correctamente ya que estos equipos son utilizados para diagnóstico y tratamientos.

El equipo más utilizado en este trabajo es Symbia T de la casa Siemens, este equipo es un sistema SPECT con diagnóstico CT para la localización anatómica precisa, dando a los médicos la calidad de imagen necesaria para realizar el diagnóstico.[11]



Figura 1.1: Siemens Symbia T

Para realizar este proyecto se utiliza la herramienta MATLAB ya que posee las librerías y toolbox necesarios para el procesamiento digital de imágenes médicas.

1.1. Estructura del documento

En este documento, para empezar, se dará una descripción sobre el área de estudio y las herramientas utilizadas para realizar el proyecto, acto seguido se comentarán los aspectos académicos del trabajo, los objetivos iniciales, las competencias adquiridas etc. A continuación, se expondrán los recursos utilizados para la realización y la planificación seguida. En los siguientes capítulos se detallarán las interfaces implementadas y los resultados obtenidos. Por último finalizamos el documento con las conclusiones y dos apéndices, el primero dedicado al manual de usuario mientras que en el segundo se exponen las versiones previas de la aplicación.

Capítulo 2

Estado actual del tema

En este capítulo abordamos el estudio teórico del procesamiento digital de imágenes médicas realizado para la implementación de la aplicación.

2.1. Introducción al procesamiento de la imagen médica digital

Una imagen se define como una función $f(x, y)$ bidimensional donde x e y representan las coordenadas espaciales de un punto y $f(x, y)$ la intensidad o nivel de gris dicho punto. Una imagen digital es aquella cuyas coordenadas espaciales son un número de elementos finito y cuya intensidad toma igualmente un rango de valores finito. Una imagen digital en dos dimensiones será por tanto una matriz de tamaño $m \times n$ con un número finito de elementos llamados píxeles [8] [9].

Los valores de gris de cada píxel tras la digitalización de una imagen dependen de la resolución específica con la que ha sido tomada; en el caso de imágenes tipo byte cuyos valores de gris son digitalizados con 8 bits, podemos llegar a tener 256 niveles de gris, donde el valor 0 se corresponde con el negro y el valor 255 con el blanco.

Tras la digitalización de la imagen, esta debe ser guardada en un formato específico como por ejemplo JPEG, GIF, en este proyecto utilizaremos el formato DICOM, específico para el uso de imágenes médicas. MATLAB dispone de librerías para la lectura de imágenes en cualquier formato, incluyendo imágenes DICOM.

El estándar DICOM se desarrolló en 1992 para poder intercambiar imágenes médicas en formato digital. Además, es un protocolo que permite la comunicación entre las entidades que generan las imágenes médicas y las que las procesan. Los rasgos principales que caracterizan a este estándar son una

cabecera(metadatos) en la que se incluyen datos relacionados con el paciente y otra parte con información de las imágenes que se han tomado del paciente. Actualmente este estándar tiene 18 partes que permiten el correcto intercambio de información de imágenes médicas. Entre estas 18 partes se pueden destacar los perfiles de seguridad de los datos, formato de los ficheros y almacenamiento de los ficheros, pero se añaden nuevas secciones cada pocos años, por lo que está en constante desarrollo.

2.2. Imágenes en medicina nuclear

Las técnicas de obtención de imágenes en medicina nuclear son la tomografía por emisión de fotón único (SPECT) y la tomografía por emisión de positrones (PET).

En PET los radionúcleos emiten dos rayos gamma en direcciones opuestas que son detectados de forma simultánea. Se utiliza un anillo formado por cristales para la detección de los positrones, por lo que no se requiere la rotación del cristal.

En SPECT los radionúcleos emiten solamente un rayo gamma en una dirección, se usa una gamma-cámara con un cristal detector que gira alrededor del paciente y se adquieren imágenes en los 360 grados.

2.2.1. Calidad de las imágenes

A continuación, se explican las tres características que definen la calidad en las imágenes médicas obtenidas mediante la Medicina Nuclear.

Ruido:

Las imágenes producidas con la técnica de medicina nuclear se caracterizan por tener un nivel alto de ruido. Esto se debe a que la cantidad de isótopos radiactivos en el cuerpo introducidos debe ser baja, por lo que el número de fotones detectados lo será también produciendo ruido con distribución de Poisson, tras la reconstrucción de las imágenes, sobretodo en el caso de SPECT, que las deteriora.

Contraste:

El contraste en la imagen está determinado por las características del radiofármaco introducido, además cuanto más tiempo se esté tomando la imagen más rayos gamma se obtendrán, por lo que el contraste mejorará al aumentar el tiempo de toma de la imagen. La dispersión de los fotones

produce radiación en sitios donde no corresponde, lo que reduce también el contraste obtenido en la imagen.

Resolución espacial:

La resolución espacial está limitada por el tamaño de los detectores individuales de cristal que reciben los rayos gamma. La forma de medirla es mediante la obtención de la MTF.

2.3. Procesado y realzado de imágenes médicas

Las operaciones de realzado y mejora de la imagen se pueden dividir en dos categorías:

- **Operaciones en el dominio espacial:** estas operaciones consisten en manipular los valores de los píxeles de las imágenes en el dominio espacial basándose en las distribuciones estadísticas de la imagen completa o de determinadas regiones. Se pueden destacar dentro de este grupo las operaciones de transformaciones del histograma, filtrado espacial y operaciones morfológicas.
- **Operaciones en el dominio frecuencial:** estas operaciones consisten en manipular la información en el dominio frecuencial basándose en las características de frecuencia de las imágenes. Se pueden destacar dentro de este grupo el filtrado en el dominio de la frecuencia y filtrado homomórfico.

2.3.1. Operaciones en el dominio espacial

Los métodos de procesado en el dominio espacial son métodos que realizan transformaciones píxel a píxel basándose en las propiedades estadísticas del histograma o en operaciones teniendo en cuenta los píxeles contiguos. Estos métodos normalmente son más rápidos pero menos eficientes que las operaciones en el dominio frecuencial.

Transformaciones y ecualización del histograma

El histograma de una imagen digital nos proporciona información sobre la distribución de las intensidades de los píxeles de la imagen. El histograma de una imagen digital con niveles de gris entre $[0\ 255]$ suponiendo que se tiene

una resolución de 8 bits será la función discreta $h(rk) = nk$, donde rk es el valor k -ésimo de nivel de gris y nk es el número de píxeles en la imagen que tienen ese valor de nivel de gris. Normalmente después se suele normalizar el histograma entre el número total de píxeles, obteniendo $p(rk) = \frac{nk}{n}$, que da una estimación de la probabilidad de ocurrencia de un nivel de gris k .

Con la ecualización del histograma se consigue expandir histogramas que tenían muchos niveles de grises coincidentes, aunque no se va a conseguir siempre un histograma totalmente uniforme. Con esto se podrá cambiar el rango de niveles de grises en la imagen, y se volverá más uniforme, ya que se fuerza el histograma para que los niveles de intensidad de gris se redistribuyan con igual probabilidad de ocurrencia.

Supongamos que una imagen tiene todos sus niveles de gris comprendidos entre $[a, b]$ donde el mínimo es a y el máximo es b . Usando un método de ecualización del histograma se cambiará el rango de valores desde el intervalo $[a, b]$ a un intervalo de rango $[c, d]$ mediante la ecuación:

$$z_{nuevo} = \frac{d - c}{b - a}(z - a) + c \quad (2.1)$$

Donde z , es el valor original del nivel de gris antes de la ecualización y z_{nuevo} es el nuevo valor del píxel tras la ecualización.

Filtrado espacial

Los métodos de filtrado espacial usan operaciones que requieren la convolución discreta de la imagen inicial con una máscara espacial con unos pesos específicos para realizar modificaciones en la imagen. Es decir, la máscara de filtrado se traslada y convoluciona a lo largo de toda la imagen para obtener la imagen final. Esta máscara suele ser cuadrada de tamaño 3x3 o 4x4. El valor de cada píxel será reemplazado por un nuevo valor que dependerá del píxel que se está tratando y de los píxeles a su alrededor, definiendo una región próxima a ese píxel con un tamaño especificado previamente. El valor nuevo del píxel se calculará tras aplicar la máscara en la zona centrada en el píxel del que se quiere hallar el nuevo valor. Dependiendo de los valores de los coeficientes de la máscara, se pueden realizar operaciones diferentes en la imagen como realzado o suavizado de la imagen.

La máscara de filtrado lineal más básica es la de media que reemplaza el valor de gris original por el valor medio de los píxeles de su entorno. Dentro de las máscaras de filtrado de media se puede usar la de la media ponderada que da más importancia al píxel central que la de media normal (figura 2.1). Con la máscara de media normal hay que usar un factor de escalado de 1/9 mientras que en la de media geométrica el factor de escalado será de 1/16.

1	1	1
1	1	1
1	1	1

1	2	1
2	4	2
1	2	1

Figura 2.1: Máscaras de media y de media ponderada [8]

Las máscaras de media se suelen usar para suavizar una imagen o para reducir el ruido, ya que el suavizado de la imagen es un método muy útil para mejorar la relación señal a ruido de la imagen. Sin embargo, esta mejora tiene un coste, ya que al suavizar la imagen se pierde resolución espacial.

De las máscaras de filtrado no lineal la más importante es la de mediana. Esta máscara reemplaza el valor de gris original por el valor mediano de los píxeles en el entorno especificado. Éstas son especialmente útiles para la reducción de ruidos de pico de la imagen como el ruido *sal y pimienta*. Sin embargo, producen pérdida también de detalles y efecto de emborronado en la imagen, pero conservan mejor los bordes que el filtrado de media.

Otro tipo de filtrado espacial son las máscaras de realzado y detección de bordes. El objetivo principal de este tipo de filtros es realzar o destacar los detalles de la imagen. Los bordes se detectan por ser puntos en los que los valores de gris de píxeles adyacentes de la imagen cambian de forma significativa. Este cambio se obtiene en el dominio discreto con una resta entre píxeles adyacentes. Se detecta el cambio con máscaras con operadores derivativos de primer orden, como la de Sobel, o de segundo orden, como las Laplacianas (figura 2.2). Se muestran a continuación en las dos primeras máscaras de la figura 2.2, las máscaras que calculan el gradiente horizontal o vertical de primer orden respectivamente. Estas máscaras que calculan los gradientes de las imágenes suelen tener un tamaño de 3x3 píxeles. Para realzado de los bordes basta con sumar a la imagen original la información del gradiente de la imagen y volver a escalar la imagen para que no supere los valores límite. Por lo tanto, se utilizará para detección y realzado de bordes las dos últimas de la figura 2.2 respectivamente.

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

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

-1	-1	-1
-1	8	-1
-1	-1	-1

-1	-1	-1
-1	9	-1
-1	-1	-1

Figura 2.2: Otras máscaras[8]

En la figura 2.2 la primera máscara que encontramos (de derecha a izquierda) es el operador de Sobel en la dirección x , y la segunda en la dirección y . La tercera máscara es un operador Laplaciano usando conectividad(adyacencia) de ocho píxeles y la última es una máscara de realzado de bordes.

2.3.2. Operaciones en el dominio de la frecuencia

Los métodos de filtrado en el dominio de la frecuencia procesan las imágenes obtenidas tras hacer la transformada de Fourier, para amplificar o atenuar determinadas componentes frecuenciales. Para ello se multiplica el resultado de la transformada de Fourier por la función de transferencia de un filtro determinado y posteriormente se hace la transformada de Fourier inversa para obtener la imagen de nuevo.

En primer lugar hay que explicar a qué rango de frecuencias corresponde cada parte de la imagen. Las componentes de alta frecuencia (mayor variación en poco espacio) son los bordes, los detalles y el ruido, mientras que las componentes de baja frecuencia corresponden a características que varían lentamente en la imagen como la intensidad media, o formas de objetos. Por todo esto el filtrado paso bajo elimina los bordes y los detalles, ya que atenúa los componentes de alta frecuencia.

En el dominio de la frecuencia, por tanto, el filtrado de paso bajo se conseguirá atenuando las componentes de alta frecuencia obtenidas en la Transformada de Fourier. Esto producirá, como se ha dicho anteriormente que los bordes se difuminen. El filtrado de paso alto implicará, por tanto, atenuar las componentes de baja frecuencia de la imagen. Como se sabe que los bordes y los cambios bruscos en los niveles de gris corresponden a las altas frecuencias se podrán conseguir realces en la imagen con filtrado paso alto.

2.4. Segmentación de imágenes médicas

La segmentación es el proceso por el cual se divide la imagen en diferentes partes u objetos, mediante la agrupación de los píxeles próximos basándose en criterios de similitud, que serán determinados usando propiedades específicas o características representativas de los píxeles de la imagen. Los métodos de segmentación de imágenes pueden ser clasificados en tres categorías:

- **Métodos basados en bordes:** la información que contienen los bordes de la imagen se usa para determinar las fronteras de los objetos.

Estas fronteras se analizan y modifican posteriormente para poder formar regiones cerradas que correspondan con los objetos segmentados detectados en las imágenes.

- **Métodos basados en píxeles:** la información estadística sobre los niveles de gris de una imagen se usa para formar regiones cerradas que correspondan con los objetos detectados en las imágenes. Existe una serie de técnicas basadas en el histograma, siendo una de las más sencillas la umbralización global.
- **Métodos basados en regiones:** la información que contienen los píxeles se analiza para realizar un proceso de expansión de regiones basado en criterios de similitud predefinidos, para formar regiones cerradas que correspondan con los objetos detectados en las imágenes.

Una vez se han separado mediante la segmentación las diferentes regiones de la imagen, se pueden obtener las características de cada región para su posterior caracterización, análisis y clasificación. Estas características pueden estar relacionadas con la forma, textura o propiedades estadísticas como valor medio o mediana de cada región.

2.4.1. Métodos basados en bordes

Para poder usar esta técnica las regiones que se quiere segmentar tienen que ser bastante homogéneas. Estos métodos se basan en el método de filtrado espacial utilizando máscaras derivativas para obtener información sobre el gradiente de primer o de segundo orden en la imagen, es decir, un operador local de derivación. La información de magnitud y de dirección del gradiente se obtienen convolucionando una máscara derivativa de primer orden como la de Sobel o de segundo orden (Laplacianas) en las direcciones verticales y horizontales. Posteriormente se obtiene la magnitud del gradiente con la ecuación:

$$M = \sqrt{G_x^2 + G_y^2} \approx |G_x| + |G_y| \quad (2.2)$$

Donde M es la magnitud del gradiente que se aproxima haciendo la suma de los valores absolutos de los gradientes horizontal y vertical.

Además hay que destacar que para poder hacer la segmentación con este método es necesario formar regiones cerradas tras la detección de los bordes. Este paso es el más complicado ya que los operadores de detección de bordes son muy sensibles al ruido y a los artefactos de la imagen.

2.4.2. Métodos basados en píxeles

Los métodos basados en la clasificación directa de los píxeles usan la información contenida en el histograma, es decir, los diferentes valores de gris o de color, para definir determinados umbrales que permitan clasificar a la imagen píxel por píxel. Este método de segmentación basándose en los píxeles se llama umbralización y es uno de los métodos más usados en la segmentación de imágenes. Generalmente una imagen umbralizada $g(x,y)$ se define como:

$$g(x, y) = \begin{cases} 1 & \text{si } f(x, y) > T \\ 0 & \text{si } f(x, y) \leq T \end{cases} \quad (2.3)$$

Por tanto, en una imagen umbralizada se pasará la imagen original a una imagen binaria con dos valores diferentes, según si el valor del píxel de la imagen original supera o no al valor del umbral escogido. Con esto ya se obtendrá una imagen segmentada en función de los valores de los píxeles de la imagen. Esta umbralización es la más sencilla de todas pues es la partición del histograma de una imagen utilizando un umbral único global, y es también una de las más usadas y efectivas.

Umbralización óptima

En el proceso de segmentación mediante umbralización óptima se elige un umbral automáticamente, siguiendo el procedimiento descrito a continuación: [8]

1. Se elige una estimación del umbral T inicial que suele ser el punto medio entre el valor máximo y mínimo de brillo de la imagen.
2. Se segmenta la imagen usando ese umbral T . Con esto se conseguirán dos grupos de píxeles, G_1 con los píxeles con intensidades $\geq T$ y G_2 con píxeles con valores de intensidad $< T$.
3. Se calculará la media μ_1 y μ_2 de los valores de intensidad de los grupos G_1 y G_2 respectivamente.
4. Se calculará el nuevo valor del umbral con :

$$T = \frac{1}{2}(\mu_1 - \mu_2) \quad (2.4)$$

5. Se repiten los pasos del 2 al 4 hasta que la diferencia entre los umbrales obtenidos en sucesivas iteraciones sea más pequeña que el parámetro T_0 definido previamente.

Una particularización de la umbralización óptima es el *Método de Otsu* [7], que produce resultados similares al método explicado anteriormente. En este caso también se selecciona en primer lugar un umbral que divida el histograma en dos partes G_1 y G_2 , cuyas medias son μ_1 y μ_2 . Posteriormente se definen las variables ω_1 y ω_2 que son la suma de todos los valores de probabilidad de nivel de gris de G_1 y G_2 respectivamente. Finalmente este método elige el umbral que maximiza la varianza entre clases de una imagen σ_B^2 , definida según la ecuación:

$$\sigma_B^2 = \omega_1 * (\mu_1 + \mu_T)^2 + \omega_2 * (\mu_2 + \mu_T)^2 \quad (2.5)$$

Donde μ_T es la intensidad de toda la imagen.

2.4.3. Métodos basados en regiones

Segmentación por crecimiento de regiones

Este procedimiento consiste en ir agrupando píxeles o subregiones dentro de otras regiones mayores en tamaño, examinando los píxeles o regiones contiguas. Se agruparán las regiones o los píxeles que tengan propiedades similares. Los criterios de similitud dependen de lo que se quiera segmentar y del tipo de imagen disponible. Así se podrán buscar criterios de similitud en función del color, de los niveles de gris, de las texturas. . .

Uno de los problemas de este método es saber en qué punto se empieza la segmentación. Además, también habrá que tener un criterio para saber cuándo se finaliza la segmentación. De forma general el crecimiento se parará cuando no haya más píxeles que satisfagan el criterio de similitud en la región. Sin embargo, los criterios de valores de intensidad, o color no tienen en cuenta la historia del crecimiento de la región, por lo que sería necesario incluir criterios adicionales que tengan en cuenta aspectos como el tamaño de la región o comparación del píxel con la media de la región ya obtenida.

Segmentación por división y fusión de regiones

Los métodos de segmentación por división de regiones examinan la heterogeneidad en una región de una propiedad predefinida, como su media, varianza, valor mínimo o máximo. Para ello se subdivide inicialmente la imagen en un conjunto de regiones arbitrarias. Posteriormente se examina cada región y si es considerada como heterogénea se divide a su vez en dos o más regiones hasta llegar a regiones de un tamaño que satisfagan el criterio de similitud individualmente.

Capítulo 3

Motivación y objetivos

En este capítulo explicaremos la motivación a la hora de escoger este proyecto y los objetivos y competencias básicas de este.

3.1. Motivación

La principal motivación a la hora de escoger este proyecto fue poder formar parte de un equipo de trabajo dedicado a la investigación donde poder formarme en este ámbito, otro de los motivos fue tener el reto de crear una aplicación con la que podría poner en práctica gran parte del conocimiento adquirido durante mis años de estudio a la par de ganar nuevo conocimiento.

3.2. Objetivos principales

Los objetivos principales del proyecto son:

- Diseñar e implementar una aplicación capaz de valorar, de forma independiente a la propuesta por las casas comerciales, parte de las pruebas de control de calidad de una gammacámara planar.
- Implementar una interfaz de usuario para el control de la aplicación y la presentación de resultados.
- Desarrollo de herramientas de ayuda adicionales para los usuarios.
- Estudio de posibles extensiones de trabajo a otros formatos de imagen o entornos.

3.3. Competencias específicas cubiertas

Las principales competencias cubiertas durante el desarrollo de este proyecto son las siguientes:

- **CII01:** *Capacidad para diseñar, desarrollar, seleccionar y evaluar aplicaciones y sistemas informáticos, asegurando su fiabilidad, seguridad y calidad, conforme a principios éticos y a la legislación y normativa vigente:*

Esta competencia se adquirió al desarrollar una aplicación funcional, esta aplicación debía ser fiable ya que se trabaja en el campo de la imagen médica, y como la aplicación se encarga de obtener los resultados de control de calidad de uno de los instrumentales médicos, es necesario que los valores proporcionados por esta sean fiables.

- **CII17:** *Capacidad para diseñar y evaluar interfaces persona computador que garanticen la accesibilidad y usabilidad a los sistemas, servicios y aplicaciones informáticas:*

Esta competencia envolvía la capacidad para diseñar una interfaz que el usuario pudiera manejar fácilmente.

- **CP05:** *Capacidad para adquirir, obtener, formalizar y representar el conocimiento humano en una forma computable para la resolución de problemas mediante un sistema informático en cualquier ámbito de aplicación, particularmente los relacionados con aspectos de computación, percepción y actuación en ambientes o entornos inteligentes:*

Esta competencia se alcanzó al ser capaz de realizar un algoritmo por medio de la información proporcionada por el documento de control de calidad [2] y reuniones con el cliente.

Capítulo 4

Recursos utilizados

En este capítulo se enumeran las herramientas que fueron necesarias para la realización de este proyecto.

4.1. Hardware

4.1.1. Siemens Symbia T

La Siemens Symbia T es una de las máquinas de las que dispone el hospital, y con la que más se ha trabajado a lo largo de esto proyecto, se trata de una solución de diagnóstico SPECT-CT avanzada y altamente integrada.[11]

4.1.2. E.Cam

La E.Cam es otra máquina disponible en el hospital, también de la casa Siemens, ofrece varios modelos para estudios óptimos de cuerpo entero, cardíaco y general, así como un alto rendimiento para cada tipo de modelo. Su diseño de pórtico abierto permite un fácil acceso para imágenes de pacientes en las camillas, en sillas de ruedas o de pie. La amplia conectividad DICOM permite la visualización de imágenes multimodales lado a lado con imágenes de medicina nuclear. [10]

En este proyecto no se trabajó mucho con imágenes de esta máquina, pero se comprobó que la aplicación también es capaz de procesar dichas imágenes obteniendo resultados válidos.

4.1.3. Portátil

Este proyecto comenzó a desarrollarse en un portátil cuyas prestaciones eran: procesador i3, 4 GB RAM. Posteriormente se acabó de desarrollar en

un portátil con prestaciones más potentes (procesador i5 12GB RAM), en ambos ordenadores la aplicación funciona correctamente. En ambos equipos se utilizó el Sistema Operativo Windows 10.

4.2. Software

4.2.1. MATLAB R2015a

MATLAB es un entorno de computación y desarrollo de aplicaciones totalmente integrado orientado para llevar a cabo proyectos en donde se encuentren implicados elevados cálculos matemáticos y la visualización gráfica de los mismos. Se decidió utilizar MATLAB al tratarse de una de las herramientas más potentes para el tratamiento de imágenes médicas que cuenta con un soporte para las imágenes DICOM.

4.2.2. Microsoft EXCEL

EXCEL es un programa desarrollado por Microsoft y permite trabajar con hojas de cálculo. Se optó pasar los resultados de la aplicación a EXCEL por requisito del cliente, ya que actualmente las pruebas de calidad las introducen en un fichero EXCEL.

4.3. Librerías

Las librerías utilizadas en MATLAB son las siguientes:

- **Toolbox Procesamiento de imágenes:** Para el procesamiento de imágenes DICOM.
- **Toolbox de Exportación e Importación de datos:** Se utilizó este toolbox para exportar los resultados a ficheros EXCEL.
- **Toolbox de Ajuste de Curvas:** Este toolbox se utilizó para ajustar funciones a gaussianas.
- **Herramientas GUIDE [3]:** Con estas herramientas se construyó la interfaz de usuario.

4.4. Otras herramientas utilizadas

4.4.1. LATEX

Para generar la memoria se utilizó LATEX con el entorno TeXStudio, este IDE proporciona un entorno moderno de escritura, con corrección ortográfica, plegado de código y visor de pdf entre otras características.

4.4.2. StarUML

StarUML es una herramienta para el desarrollo con UML, en este proyecto fue utilizada para la creación de los diagramas que se verán en el siguiente capítulo.

Capítulo 5

Análisis y Diseño

En este capítulo se describe el análisis y diseño realizado durante el proyecto.

5.1. Plan de trabajo inicial

Este proyecto surgió de mis prácticas externas donde lo comencé, acordamos realizar cuatro pruebas de control de calidad recogidas en el documento español de control de calidad de instrumentación de Medicina Nuclear [2], estas pruebas son: Tamaño de Píxel, Uniformidad Intrínseca, Resolución Espacial y Uniformidad Tomográfica.

Como se puede observar en la figura 5.1 el primer paso fue realizar un estudio previo del proyecto con el fin de desarrollar un plan de trabajo inicial, el plan que se elaboró consistía en realizar una interfaz para cada prueba por separado siguiendo el ciclo: Diseño e implementación \rightarrow validación. Al finalizar, las cuatro interfaces se integrarían en una interfaz que englobaría todas las pruebas y las herramientas desarrollada.

El siguiente paso fue el análisis de requerimientos que veremos más detalladamente en el próximo apartado.

Por último antes de abordar la aplicación fue necesario estudiar conceptos básicos sobre el procesamiento de imágenes y aprender la realización de interfaces en MATLAB.

Fases	Duración Estimada (horas)	Tareas (nombre y descripción, obligatorio al menos una por fase)
Estudio previo / Análisis	80	Tarea 1.1: Organización inicial del proyecto Se desarrollará un plan inicial de trabajo
		Tarea 1.2: Análisis de requerimientos Se estudiará la realización de las pruebas de calidad que se desean incluir en la aplicación y las necesidades de uso por parte de los usuarios. Se tendrá presente la flexibilidad para adaptarse a nuevos test/entornos.
		Tarea 1.3: Estudio de técnicas específicas Métodos de procesamiento de imágenes y visión por computador.
Diseño / Desarrollo / Implementación	170	Tarea 2.1: Estudio de las herramientas y entornos alternativos de desarrollo Estudio de las herramientas necesarias para realizar el proyecto.
		Tarea 2.2: Diseño y desarrollo de las pruebas Se diseñará una interfaz para cada prueba de calidad deseada.
		Tarea 2.3: Validación En diferentes ciclos se irán validando tanto los resultados de los controles de calidad como la usabilidad de las interfaces.
		Tarea 2.4: Integración Se creará una interfaz común que englobe todas las interfaces anteriormente diseñadas.
Evaluación / Validación / Prueba	20	Tarea 3.1: Depuración y pruebas Se realizará pruebas de la interfaz final en busca de errores, y se probará con usuarios.
		Tarea 3.2: Se estudiarán posibles extensiones.
Documentación / Presentación	30	Tarea 4.1: Redacción de la memoria
		Tarea 4.2: Elaboración de la presentación
		Tarea 4.3: Envío de una comunicación (póster) al Congreso Conjunto SEFM-SEPR, Girona 2016.

Figura 5.1: Cuadro informativo de la planificación

5.2. Diseño

En la fase de diseño se consideraron varias opciones de implementación como el uso de Python, C, Java, Matlab, tras estudiar dichas opciones, decidimos utilizar Matlab al tratarse de una herramienta potente para el tratamiento de imágenes, por su adecuación al desarrollo rápido de prototipos, por tener una curva de aprendizaje menos pronunciada y por contar con librerías ya desarrolladas que permiten simplificar notablemente la implementación, entre ellas la librería de procesamiento de imágenes que cuenta con funciones para el formato DICOM que trataremos en este proyecto.

Como se describió previamente el diseño de esta aplicación fue descompuesto en un número de interfaces que fueron creándose incrementalmente validándolas con el equipo de trabajo y el tutor académico una vez acabado cada prototipo recogiendo de estas validaciones nuevas mejoras e implementándolas.

5.3. Metodología

Se ha seguido una metodología de desarrollo en espiral basada en la generación incremental de prototipos funcionales que son evaluados para determinar la corrección de las pruebas y las posibles mejoras a introducir.

En las primeras reuniones con el cliente se explicó lo que debía realizar cada prueba y lo que el usuario quiere ver en la interfaz. En posteriores reuniones se acordaron los requisitos del sistema que describiremos a continuación para cada prueba:

- **Tamaño de Píxel:** Debe mostrar al usuario la distancia en píxeles entre dos cargas puntuales y el tamaño de píxel calculado indicando al usuario cuando dichos datos sobrepasan la tolerancia establecida por el fabricante, además se mostrarán las imágenes de cada detector y eje (cuatro en total). También debe exportar a EXCEL los resultados.
- **Uniformidad Intrínseca:** Se mostrará al usuario la imagen de cada detector y el resultado del cálculo de la prueba indicando si hay algún error en dichos resultados, además se mostrará las imágenes procesadas y exportará a EXCEL los resultados.
- **Resolución Espacial:** Al igual que en el tamaño de píxel se mostrará las cuatro imágenes correspondientes a los detectores y los ejes, el usuario podrá ver los resultados de la prueba para cada imagen. Además, el usuario podrá ver las gráficas generadas por la prueba y exportar los resultados a EXCEL.
- **Uniformidad Tomográfica:** Esta prueba fue la que más reuniones con el cliente necesitó y la que más veces se cambiaron los requisitos, inicialmente solo era necesario visualizar las imágenes (128 cortes), y mostrar un histograma para cada una. Posteriormente se añadió la capacidad de mostrar el resultado calculado internamente por el sistema, comprobar si se produce algún error, y exportar a EXCEL los valores calculados y la media de estos. En una última reunión se acordó que el usuario pudiera ver un resumen de errores (en el caso de que

se produzcan) y poder navegar hasta el primer error localizado en la prueba.

Igualmente se trataron en las reuniones las herramientas a desarrollar: renombrado de ficheros y visualización de histórico, decidiendo como nombrar los ficheros y como debían ser las gráficas de los históricos.

5.4. Estructura de la aplicación

A continuación, analizaremos la estructura que tiene nuestra aplicación.

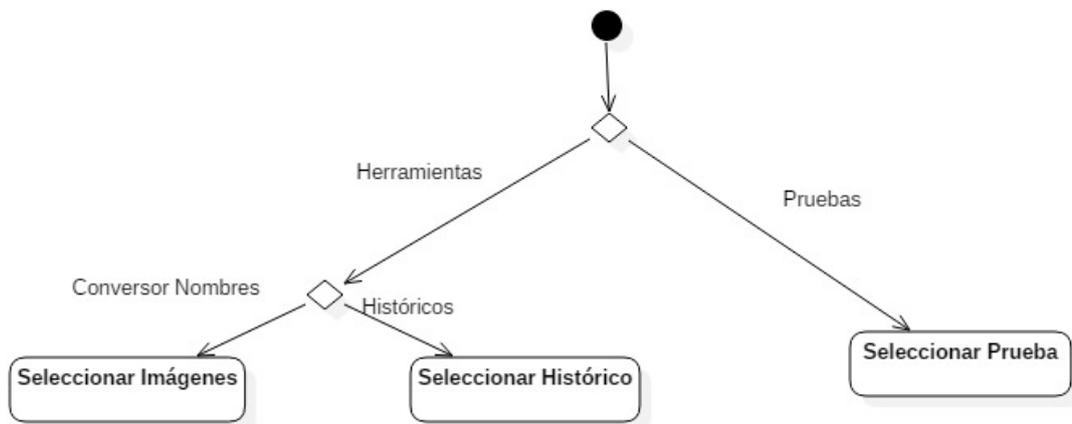


Figura 5.2: Diagrama Aplicación

Nuestra aplicación, como vemos en la figura 5.2 comienza abriendo una interfaz por la cual el usuario puede seleccionar dos acciones, la primera acción es el uso de *Herramientas*, en la cual el usuario podrá utilizar el conversor de nombres o acceder a los históricos.

Ahora desglosaremos cada parte de la aplicación, empezando por el conversor de imágenes.



Figura 5.3: Diagrama Conversor de Imágenes

En el anterior diagrama vemos cómo funciona internamente el conversor de imágenes, en primer lugar, al usuario se le da la opción de seleccionar las imágenes a las que desea aplicar el cambio de nombre, a continuación internamente se leen los metadatos de la cabecera DICOM buscando los elementos que compondrán el nuevo nombre como: el nombre de la prueba, la fecha de adquisición, etc. Por último, se aplica el cambio de nombre a la imagen dejándola en el directorio donde se encuentra la imagen original.

En la figura 5.4, tenemos el diagrama de las actividades que sigue los históricos, independientemente de la prueba todos poseen la misma estructura, la única diferencia reside en el fichero EXCEL que se lee en cada caso. La estructura de un histórico es la siguiente: primero se lee del fichero EXCEL los resultados de las pruebas, tras esto se genera la gráfica de la prueba.

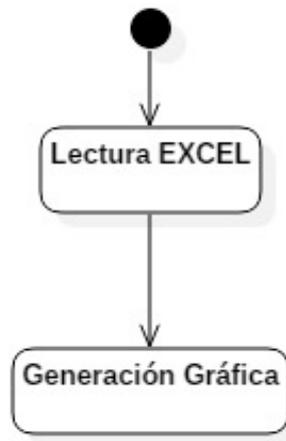


Figura 5.4: Diagrama Históricos

Para las pruebas no usamos una estructura genérica como en el caso de los históricos, en su lugar, en las pruebas de *Tamaño de píxel* y la *Resolución espacial* utilizamos una estructura distinta que para las pruebas de Uniformidad, esto se debe a que las dos primeras pruebas utilizan cuatro imágenes y es necesario comprobar que las imágenes que carga el usuario pertenecen a la misma sesión de prueba.

Para las pruebas de *Tamaño de píxel* y *Resolución espacial*, se utiliza una estructura como la de la figura 5.5, podemos ver que la aplicación una vez abierta da al usuario la opción de seleccionar las imágenes que quiere procesar, una vez seleccionadas las cuatro imágenes se comprueba si pertenecen a la misma prueba (si fueron adquiridas el mismo día), en el caso de que esta condición no se cumpla, saltará un error. Una vez se han seleccionado las imágenes se procesan internamente, se calcula los resultados y posteriormente se muestran al usuario, si este desea guardar los resultados a EXCEL deberá pulsar el botón correspondiente. Una vez acabado este proceso, el usuario puede decidir seguir realizando más pruebas o simplemente cerrar la aplicación.

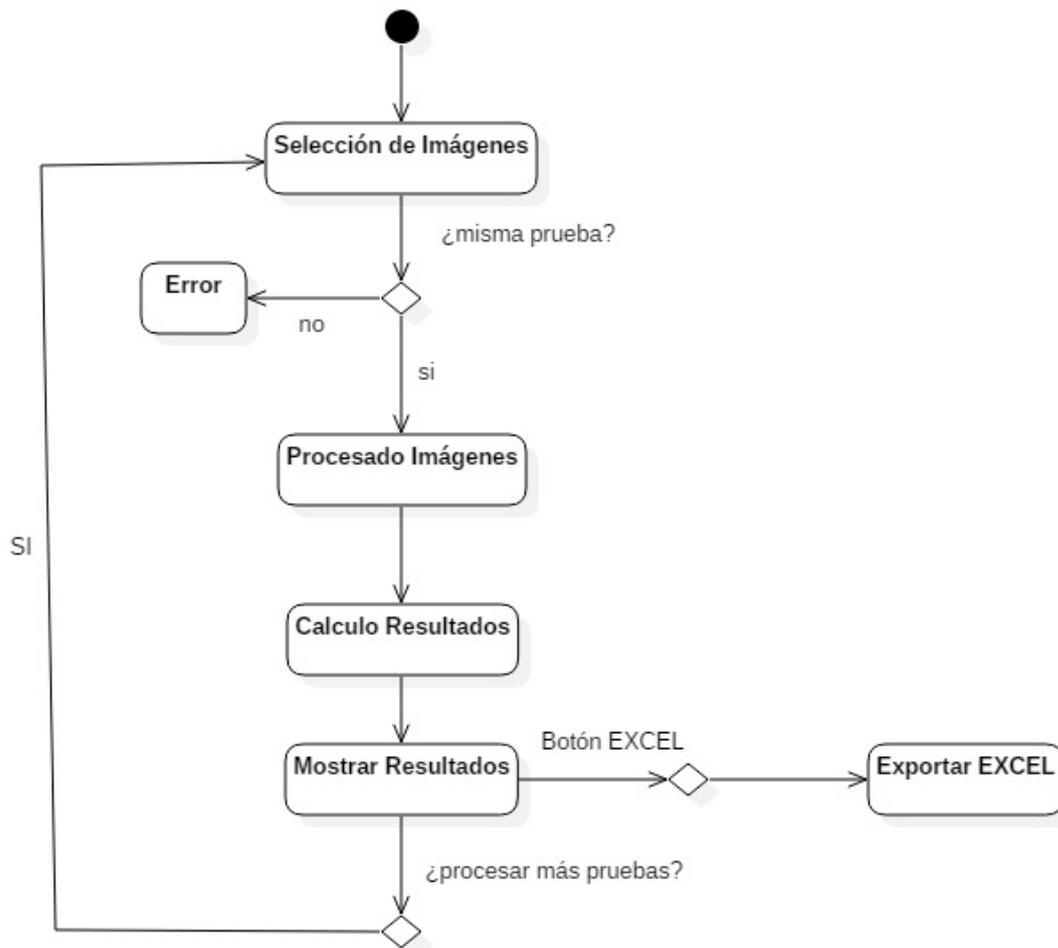


Figura 5.5: Diagrama Pruebas: Tamaño de Píxel y Resolución Espacial

Para las pruebas de *Uniformidad Intrínseca* y *Uniformidad Tomográfica*, se utiliza otra estructura en la que no se comprueba la fecha de la imagen, ya que en estas pruebas solo se procesa una sola imagen por prueba. Como se puede apreciar en la 5.6, la aplicación comienza seleccionando la imagen a la que realizarle la prueba, acto seguido se aplica el procesado y se calculan los resultados que se mostrarán al usuario en su cuadro correspondiente, por último si este desea exportar los datos calculados a EXCEL deberá pulsar el icono correspondiente. Tras acabar este ciclo el usuario puede seguir cargando nuevas imágenes.

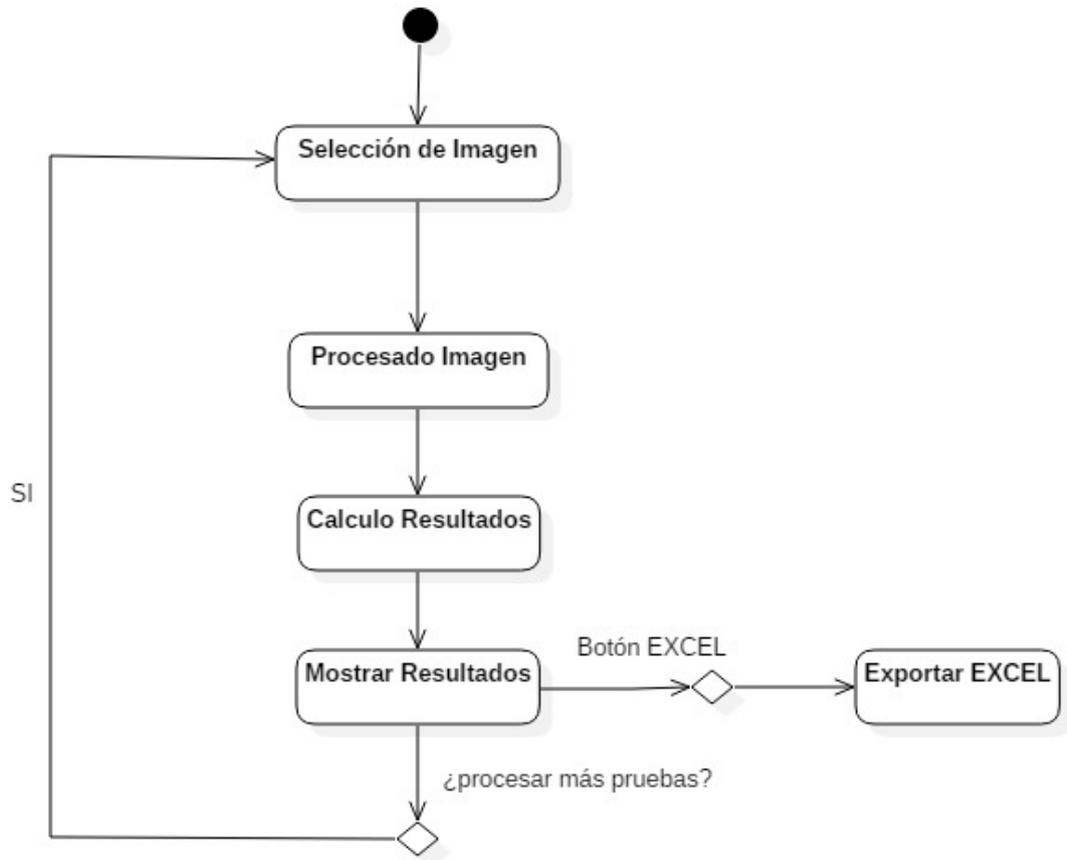


Figura 5.6: Diagrama Pruebas: Uniformidad

Capítulo 6

Implementación de las pruebas

En este capítulo describiremos como se han implementado cada una de las pruebas, describiendo para cada una los fundamentos teóricos y los elementos importantes recogidos en el documento de control de calidad [2] para su implementación.

6.1. Descripción de funciones MATLAB utilizadas

Antes de describir cada prueba y como se implementó, se describirá las funciones de MATLAB y sus Toolbox más importantes que fueron utilizadas para el desarrollo de este proyecto.

6.1.1. Funciones internas de MATLAB

- **fft**

Realiza la transformada rápida de Fourier.

$$\text{fft}(x);$$

- **histogram**

Esta función genera la gráfica de un histograma correspondiente a una imagen.

6.1.2. Procesamiento de imágenes

A continuación se describirán las funciones pertenecientes al Toolbox de procesamiento de imágenes de MATLAB [6].

Funciones para imágenes Dicom

MATLAB nos proporciona herramientas para leer archivos DICOM y escribirlos, en este proyecto solo nos interesa la lectura de dichos archivos.

■ **dicominfo**

Los archivos DICOM contienen metadatos que proporcionan información sobre los datos de la imagen, como el tamaño, las dimensiones, los ajustes del equipo utilizados para capturar la imagen etc. Para leer estos metadatos de un archivo DICOM, se utiliza la función *dicominfo*, que devuelve la información en una estructura MATLAB (struct), donde cada campo contiene un metadato.

Ejemplo: `filename = 'Spatial Resolution20161010Det1 X.dcm';`
`dicominfo(filename);`

Obteniendo la siguiente estructura en MATLAB:

Field ▲	Value
Filename	'C:\Users\Chaxi\Documents\TFG\V5\PRUEBAS\RESOLUCIONES...
FileModDate	'13-oct.-2016 12:24:08'
FileSize	12590068
Format	'DICOM'
FormatVersion	3
Width	1024
Height	1024
BitDepth	16
ColorType	'grayscale'
FileMetaInformationGroupLength	192
FileMetaInformationVersion	[0;1]
MediaStorageSOPClassUID	'1.2.840.10008.5.1.4.1.1.20'
MediaStorageSOPInstanceUID	'1.3.12.2.1107.5.6.1.38107.30000116101006511723400000002'
TransferSyntaxUID	'1.2.840.10008.1.2.1'
ImplementationClassUID	'1.3.12.2.1107.5.1.4'
ImplementationVersionName	'SIEMENS_S5VB38B'
SpecificCharacterSet	'ISO_IR 100'
ImageType	'ORIGINAL\PRIMARY\STATIC\EMISSION'

Figura 6.1: Metadatos DICOM

■ **dicomread**

Para leer la imagen DICOM se utiliza la función *dicomread*. Como en este proyecto trabajamos con imágenes DICOM con varios cortes la sintaxis de esta función es la siguiente:

```
dicomread(filename, 'frames', x);
```

Donde x es un número entero que indica el corte de la imagen.

Visualización y exploración de imágenes

- **imshow**

La función *imshow*, muestra la imagen pasada como parámetro.

```
imshow(imageName);
```

Transformaciones geométricas

- **imresize**

La función *imresize*, cambia el tamaño de la imagen pasada como parámetro, podemos indicarle una escala o bien el número de columnas y filas.

```
imresize(image, scale);  
imresize(image, [nrows, ncols])
```

- **imcrop**

Con la función *imcrop* podemos recortar una imagen, solo tenemos que pasarla como parámetro junto con un vector de posición de cuatro elementos que especifica la posición y tamaño del rectángulo a recortar.

```
rect = [xmin ymin width height]; imcrop(image, rect);
```

Aplicar filtros

- **imfilter**

Esta función aplica un filtro a la imagen, este filtro puede ser creado con *fspecial* o bien ser uno de los filtros de media explicados anteriormente.

```
fil = [1,2,1; 2,4,2; 1,2,1]; imfilter(image, fil);
```

Mejora de la imagen

- **imadjust**

La función *imadjust* ajusta los valores de intensidad de la imagen pasada como parámetro.

```
imadjust(image);
```

- **bwareaopen**

Esta función elimina los objetos (componentes conectados) que tienen menos de un número de píxeles pasado como parámetro, de una imagen binaria. También se puede especificar la conectividad entre 8 o 4 (píxeles adyacentes).

```
bwareaopen(image, nP, conn);
```

El resultado es otra imagen binaria.

Conversión de imágenes

- **im2bw**

Esta función transforma una imagen a binaria basándose en un umbral.

```
im2bw(image, level);
```

- **graythresh**

La función *graythresh* calcula el umbral de una imagen pasada como parámetro utilizando el método Otsu.

```
level = graythresh(image);
```

Análisis de imágenes

- **regionprops**

Esta función se encarga de medir las propiedades de las regiones de la imagen pasada como parámetro. Estas son algunas de las propiedades de los objetos que devuelve:

- Area: Devuelve el número de píxeles del objeto.

- **BoundingBox**: Devuelve el rectángulo más pequeño que contiene el objeto.
- **Centroid**: Devuelve el vector que especifica el centro de masa del objeto.

```
regionprops(image, prop);
```

■ **improfile**

Con esta función podemos trazar el perfil de cuentas de una imagen DICOM, recupera los valores de intensidad de píxeles a lo largo de una línea.

```
x = [19 427 416 77]; y = [96 462 37 33]; improfile(image, x, y);
```

6.1.3. Ajuste de curvas

A continuación se describirán las funciones pertenecientes al Toolbox de ajuste de curvas de MATLAB [4].

Regresión lineal y no lineal

■ **fit**

Esta función ajusta la curva o la superficie de los datos a un modelo especificado (en nuestro caso lo utilizaremos para ajustar a una gaussiana).

```
fit(x, y, 'gauss1');
```

6.1.4. Importar y exportar datos

Ahora describiremos las funciones que pertenecen al Toolbox de importar y exportar datos, estas funciones las utilizaremos para importar y exportar datos a un documento EXCEL.[5]

■ **xlswrite**

La función `xlswrite` escribe una tabla en un fichero EXCEL.

```
xlswrite(table, 'file.xlsx');
```

■ **xlsread**

Esta función lee un fichero EXCEL.

```
xlsread('file.xlsx');
```

6.2. Prueba 1. Tamaño De Pixel

6.2.1. Objetivo de la prueba

Según el documento de control de calidad de la instrumentación de medicina nuclear [2] el objetivo de la prueba Tamaño de Píxel es determinar el tamaño que un píxel representa en el cristal de centelleo y que corresponde a las dimensiones de las celdas de la matriz utilizada. Se trata de obtener el factor de escala entre las dimensiones del objeto y de su imagen. Se sitúan cargas puntuales separadas una distancia de 200 mm (maniquí de cuadrantes), obteniéndose dos imágenes por detector (ejes X e Y). La interfaz calculará la distancia en píxeles entre dichas cargas dando como resultado el tamaño de píxel buscado. La aplicación informa si el tamaño calculado está dentro de la tolerancia establecida con respecto al dato proporcionado por el fabricante.

6.2.2. Explicación de la prueba

En el documento de control de calidad de la instrumentación de medicina nuclear [2] se encuentra toda la explicación de dicha prueba, en esta memoria explicaremos los conceptos clave necesarios para realizar su implementación.

Adquisición de datos

Se obtendrán tras realizar la prueba 4 imágenes con dos fuentes puntuales (dos por cada detector y por cada eje), dichas imágenes tendrán la siguiente apariencia:

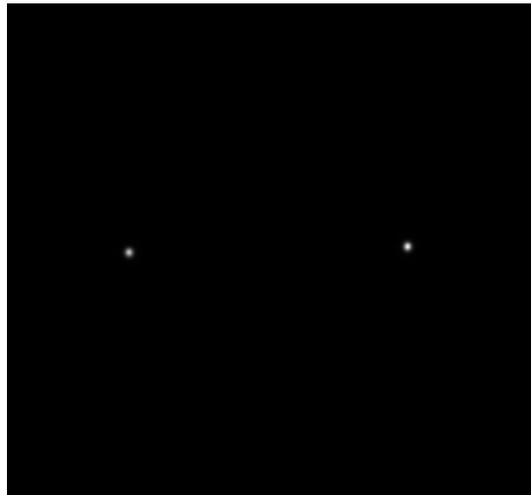


Figura 6.2: Imagen Tamaño Píxel

Se puede apreciar en la figura 6.2 las dos cargas puntuales comentadas anteriormente.

Procesado y análisis del resultado

Se calculará el centro de gravedad para cada una de las imágenes adquiridas, obteniendo cuatro centros:

- (x^+, y_1) y (x^-, y_2) para las dos posiciones extremas sobre el eje X.
- (x_1, y^+) y (x_2, y^-) para las dos posiciones extremas sobre el eje Y.

Acto seguido obtendremos las distancias, en número de píxeles, entre las imágenes de los puntos x^+ y x^- y entre y^+ e y^- .

El tamaño de píxel se obtiene dividiendo la distancia de las fuentes puntuales en mm por la distancia correspondiente en píxeles (ecuación 6.1).

$$TP = \frac{d_{obj}(mm)}{d_{imagen}(pixels)} \quad (6.1)$$

Los valores obtenidos se compararán con las referencias del fabricante.

Tolerancias

Según el fabricante el tamaño de píxel nominal es 0.6mm[11], el valor obtenido al realizar la prueba no puede superar a este valor nominal en un 5%.

Además la diferencia del valor medido en la dirección X no debe diferir en más del 5% del medido en la dirección Y.

6.2.3. Implementación de la prueba

La interfaz tendrá el siguiente diseño:

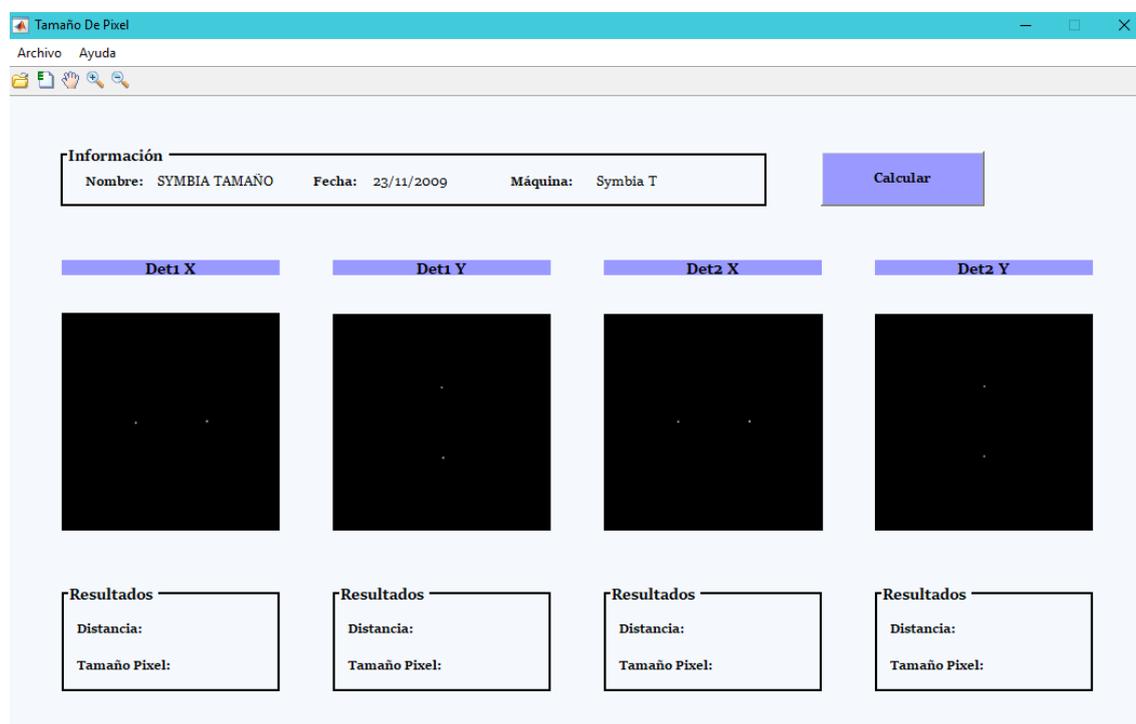


Figura 6.3: Interfaz Tamaño Píxel

Inicialmente se abrirá la prueba más reciente almacenada en el directorio de la aplicación llamado *PRUEBAS*. En la interfaz de la figura 6.3 podemos distinguir en la parte superior un cuadro de información donde se mostrará el nombre de la imagen, la fecha de adquisición y la máquina usada, todos estos datos se sacarán de la cabecera de la imagen DICOM. El usuario podrá ver las cuatro imágenes obtenidas durante la adquisición de datos, cada imagen se encuentra encabezada por el detector y el eje correspondiente a su adquisición. Bajo cada imagen encontramos un cuadro donde se mostrarán los resultados de la prueba (en el caso de que el resultado sobrepasara los límites de tolerancia se mostrará en rojo). El botón calcular situado en la esquina superior derecha, calcula los resultados y mostrándolos en su correspondiente cuadro de resultados. Observamos que además disponemos de una barra de herramientas en la que tenemos los siguientes iconos (de izquierda a derecha):

- Abrir imágenes: al pulsar sobre este icono se abrirá una ventana donde el usuario podrá seleccionar las cuatro imágenes para ser procesadas (figura 6.5).
- Exportar a EXCEL: Exporta el resultado de un cálculo a EXCEL.

- Moverse por una imagen ampliada.
- Iconos de Zoom: Aumentan o disminuyen las imágenes.

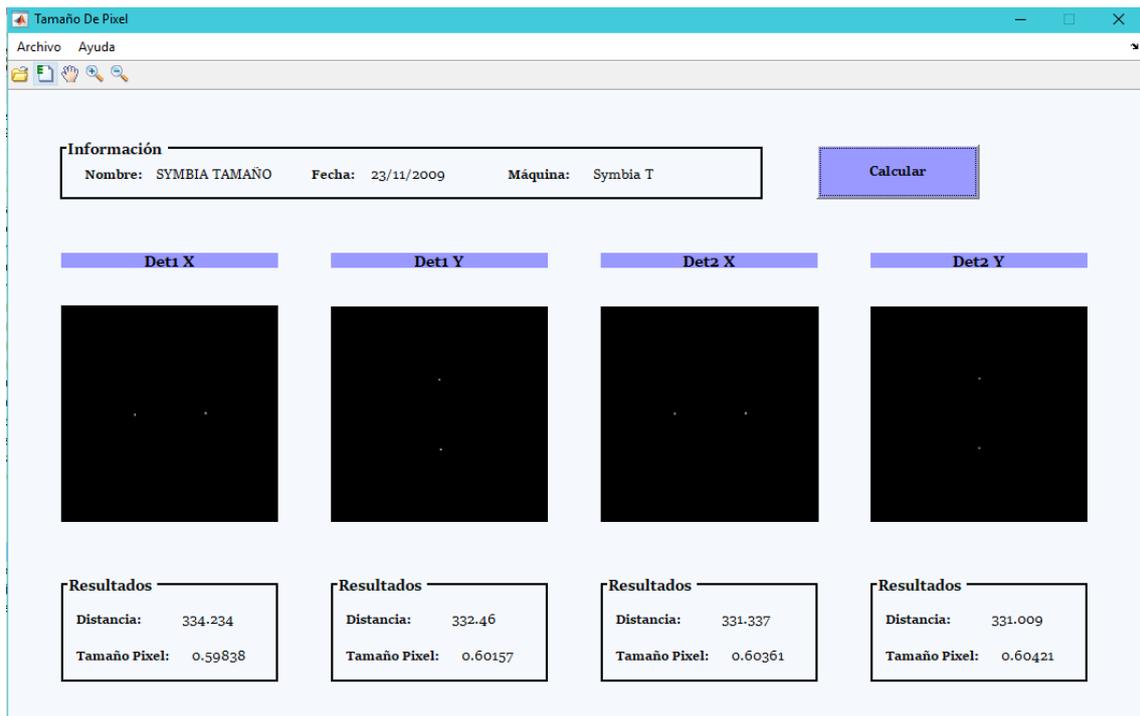


Figura 6.4: Interfaz Tamaño Píxel

En la figura 6.4 tenemos un ejemplo de la interfaz con los valores calculados. Podemos observar que se muestra por cada imagen dos resultados: la distancia en número de píxeles entre fuentes y el tamaño de píxel en mm calculado.

Además de la barra de herramientas, también disponemos de un menú donde nos encontramos con dos categorías: Archivo y Ayuda. En el menú de Ayuda el usuario se encuentra con una pequeña explicación de varios aspectos de la aplicación. En Archivo disponemos de la herramienta para seleccionar las imágenes a procesar *Abrir Imágenes* (que también puede utilizarse con el atajo de teclado CTRL + O), esta herramienta abrirá el siguiente diálogo donde el usuario seleccionará las imágenes a tratar:

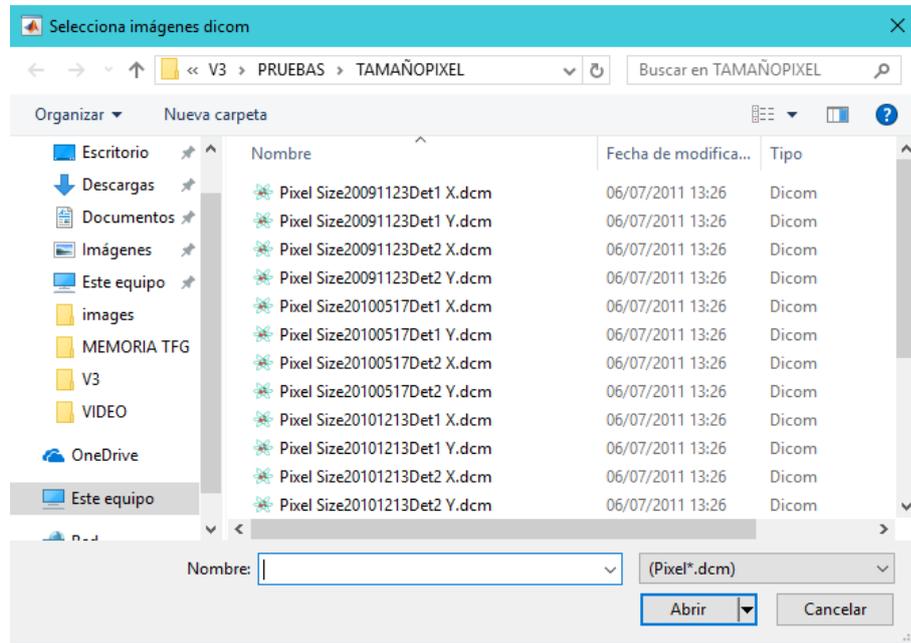


Figura 6.5: Interfaz Tamaño Píxel

Hay que destacar que estas imágenes han de seleccionarse de cuatro en cuatro, en el caso de que el usuario no seleccione las cuatro imágenes saltará el error que podemos ver en la figura 6.6.

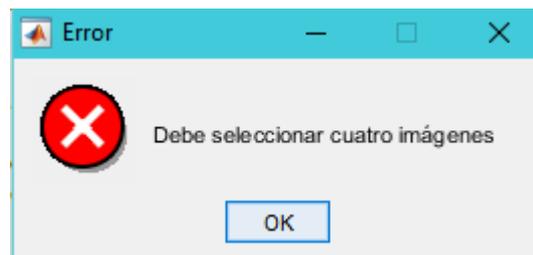


Figura 6.6: Interfaz Tamaño Pixel-Cargar Imágenes Error 1

Además para evitar el procesamiento de imágenes que no correspondan a misma sesión de la prueba, saltará un error como el que se puede ver en la figura 6.7 si el usuario introduce cuatro imágenes que no pertenecen a la misma prueba (esto se comprueba mediante la fecha de obtención de las imágenes, ya que imágenes correspondientes a la misma sesión serán adquiridas el mismo día)

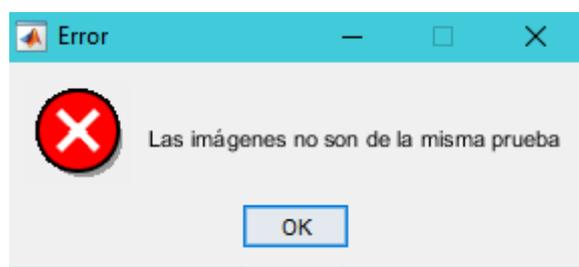


Figura 6.7: Interfaz Tamaño Pixel-Cargar Imágenes Error 2

Si el usuario desea guardar los resultados del cálculo de una prueba exportándolos a un fichero EXCEL, basta con pulsar sobre el icono correspondiente en la barra de herramientas o bien en el menú Archivo->Exportar, también accesible con el atajo CTRL+E. Una vez seleccionada esta opción de la aplicación, se realizará la exportación y una vez acabada se mostrará al usuario un mensaje notificándole que se ha terminado con éxito (figura 6.8).

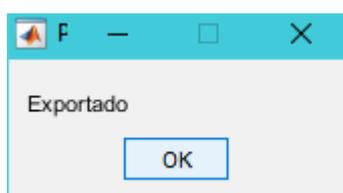


Figura 6.8: Interfaz Tamaño Pixel-Exportar

Dependiendo de la máquina que detecte la aplicación (Symbia T o E.Cam) se exportará a un fichero EXCEL específico para la prueba y la máquina. En el caso de la máquina Symbia T el fichero se llamará *SymbiaTamañoPixel.xlsx* y estará situado en el directorio *EXCEL/Symbia*, si la máquina es E.Cam se exportará a un fichero en *EXCEL/Ecam* denominado *EcamTamañoPixel.xlsx*. En la figura 6.9 podemos ver la estructura de estos ficheros:

	A	B	C	D	E	F	G	H	I	J
#	Nombre	Fecha	DetectorEje	Distancia	NumeroPixeles	TamPixel	TamPixelNominal	DiferenciaEjes	DesviacionValorNominal	Resultado
	Pixel Size20091123Det1 X.dcm	23/11/2009	Det1 X	200	334,2341837	0,59838284		0,6	-0,533573744	-0,169693717 Correcto
	Pixel Size20091123Det1 Y.dcm	23/11/2009	Det1 Y	200	332,460263	0,601575653		0,6	0,530741844	0,362974586 Correcto
	Pixel Size20091123Det2 X.dcm	23/11/2009	Det2 X	200	331,3370091	0,603615034		0,6	-0,099062518	0,703211581 Correcto
	Pixel Size20091123Det2 Y.dcm	23/11/2009	Det2 Y	200	331,0091031	0,60421299		0,6	0,098964481	0,802970718 Correcto

Figura 6.9: Excel - TamañoPixel

En la primera columna nos encontramos con el nombre de la imagen procesada, le siguen la fecha de adquisición de la imagen y el par detector-eje específico de esta. A continuación tenemos los resultados de la prueba: *Distancia* que se corresponde con la distancia real indicada por el fabricante

en mm que hay entre las fuentes, tras este valor tenemos la distancia calculada en nuestra aplicación (en número de píxeles) y el tamaño de píxel calculado junto al nominal que nos proporciona el fabricante ambos en mm. Por último tenemos el cálculo de la diferencia entre los ejes y la desviación con respecto al valor nominal (estos datos fueron requeridos por el cliente). Para terminar nuestro fichero tenemos una etiqueta que indica si el resultado calculado es correcto o no, es decir, si la prueba se pasa con éxito o no.

6.2.4. Algoritmo

El algoritmo de esta prueba como se ha indicado anteriormente consiste en procesar una imagen como que vimos en la figura 6.2, en ella podemos apreciar las dos cargas puntuales como dos puntos blancos sobre el fondo negro, la prueba consiste en medir la distancia entre estas dos cargas, para ello como primer paso tras leer la imagen DICOM con la función *dicomread*, ajustamos la intensidad de los píxeles con *imadjust*, que nos deja una imagen como la que podemos ver en la figura 6.10(a).

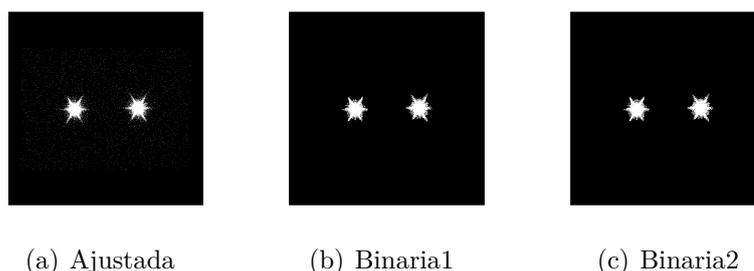


Figura 6.10: Imágenes Adquiridas

En la figura 6.10(a) podemos apreciar varios tonos blancos, muchos puntos sueltos y algunos más concentrados, estos se corresponden con las cargas puntuales. El siguiente paso es transformar la imagen a binario para ello utilizaremos la función *im2bw*, dejándonos con una imagen como la vista en la figura 6.10(b) en la que podemos ver tonos solo negros y blancos en contraste con la anterior que era una imagen de niveles de gris. A continuación eliminaremos los objetos pequeños de la imagen (si los hubiera) con la función *bwareaopen*, lo que nos deja solo los dos objetos grandes, en este ejemplo al pasar a binario solo nos quedan dos objetos por ello no hay diferencia entre las figuras 6.10(b) y 6.10(c).

Por último con *regionprops* obtenemos la información de ambos objetos como su área y el centro de cada uno, a partir de los centros de cada objeto

calculamos su distancia (en número de píxeles) que posteriormente pasamos a mm con la ecuación 6.1 y mostramos el resultado en el cuadro correspondiente de la interfaz.

6.3. Prueba 2. Uniformidad Intrínseca

6.3.1. Objetivo de la prueba

Como queda recogido en [2] esta prueba refleja la capacidad que tiene una gammacámara de reproducir, sin alteraciones, la distribución de actividad del paciente. Es un indicador que depende de distintos factores; entre otros de la linealidad espacial y del cálculo de la energía.

Para el cálculo, seguimos los pasos indicados en [2] para esta prueba, aplicando los filtros de imagen correspondientes y corrigiendo por distancia a la fuente puntual, que en el caso de esta gammacámara se coloca cercana a los detectores. Tras generar los resultados, la aplicación exporta a una hoja de Microsoft Excel y comprueba que las tolerancias de los valores sean correctas.

6.3.2. Explicación de la prueba

En el documento de control de calidad de la instrumentación de medicina nuclear [2] se encuentra toda la explicación de dicha prueba, a continuación, explicaremos los conceptos clave necesarios para realizar su implementación.

Adquisición de datos

En la adquisición hay que tener en cuenta que la fuente puntual se coloca a una distancia mayor o igual a 5 veces la diagonal del campo total de visión (CTVU), sobre la recta perpendicular del cristal como podemos ver en la figura 6.11.

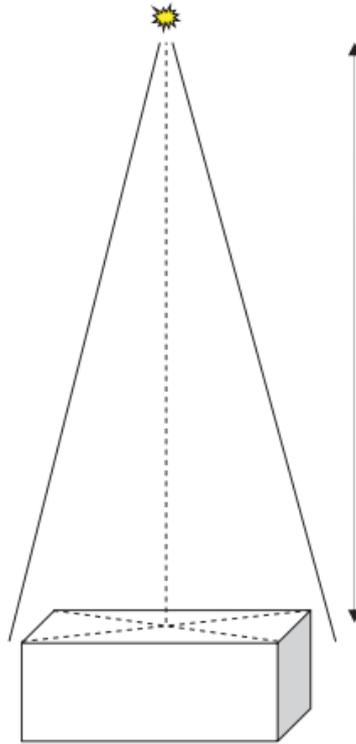


Figura 6.11: Colocación de la fuente respecto de la superficie del detector.[2]

Tras realizar la prueba se obtendrá una imagen como la siguiente, cada imagen DICOM tendrá dos cortes (frames) correspondientes con el detector:



Figura 6.12: Imagen de Uniformidad Intrínseca

En estas imágenes hay que tener en cuenta que tendremos dos áreas de estudio, por un lado el CTVU o UFOV (campo total de visión útil) que se

corresponde con la zona de la imagen capturada, por el otro lado el CCVU o CFOV (campo central de visión útil) que es el 70 % del área anterior.

Procesado y análisis de resultado

Para comenzar el análisis se eliminan los píxeles pertenecientes a los bordes de la siguiente forma: se asigna cero cuentas a todos los píxeles del borde del CTVU (campo total de visión útil) que tengan menos del 75 % de las cuentas promedio, con cuentas nos referimos al valor que toma un píxel concreto, este valor se corresponde con el conteo de radiación captado durante la prueba. A continuación, los píxeles que ahora poseen uno de los cuatro píxeles adyacentes con cero cuentas, se pondrán a cero también. Acto seguido se suaviza la imagen aplicando el filtro de 9 puntos (figura 6.13).

1	2	1
2	4	2
1	2	1

Figura 6.13: Filtro [2]

Se define la uniformidad integral, UI, como la diferencia entre el máximo y mínimo número de cuentas registradas por píxel en el área de estudio, dividida por la suma de ambas y multiplicada por cien (ecuación 6.2).

$$UI = \frac{C_{max} - C_{min}}{C_{max} + C_{min}} * 100 \quad (6.2)$$

La uniformidad diferencial, UD, se define como el valor máximo de todos los obtenidos al aplicar la ecuación 6.2 a cada conjunto de 5 píxeles contiguos en cada fila o columna del área a estudiar.

Para comprobar si la prueba es correcta calcularemos los siguientes valores:

- UICT (Uniformidad Integral en el Campo Total) y UICC (Uniformidad Integral en el Campo Central).
- UDCT (Uniformidad Diferencial en el Campo Total) y UDCC (Uniformidad Diferencial en el Campo Central).

En resumen se calcularán los valores de la uniformidad integral y diferencial para las dos áreas anteriormente explicadas (CTVU y CCVU).

Tolerancias

En esta prueba los valores obtenidos deben ser menores o iguales a los proporcionados por el fabricante. En el manual de Siemens[11] nos encontramos con la tabla 6.1 que nos da dichos valores.

	Valor	
	CFOV	UFOV
UI	$\leq 2.94\%$	$\leq 3.74\%$
UD	$\leq 2.54\%$	$\leq 2.74\%$

Cuadro 6.1: Tabla Valores Especificados por el fabricante.[11]

En las tolerancias de esta prueba el valor de los datos no puede exceder en un 10% a los valores anteriormente comentados, al aplicar esta regla obtenemos la siguiente tabla de tolerancias con la que comprobaremos que la prueba es o no correcta.

	Tolerancia	
	CFOV	UFOV
UI	$\leq 3.2\%$	$\leq 4.1\%$
UD	$\leq 2.8\%$	$\leq 3.0\%$

Cuadro 6.2: Tabla Tolerancias

Ambas tablas comparten la misma estructura, en el margen izquierdo tenemos el valor que estamos calculando (UI o UD) mientras que en el margen superior nos encontramos con el área de estudio.

6.3.3. Implementación de la prueba

Como podemos ver en la figura 6.14 los elementos que conforman el menú son similares a los de la anterior prueba.



Figura 6.14: Interfaz

En la parte superior nos encontramos con el cuadro de información donde recogeremos el nombre de la imagen, la fecha de adquisición y la máquina con la que fue obtenida. Junto a este cuadro de información tenemos el botón calcular, que al igual que el caso anterior, será el encargado de realizar los cálculos de la prueba.

En esta prueba tendremos una imagen por cada detector (es una sola imagen con dos cortes, uno por cada detector) y una caja de resultados para cada uno de ellos. Los resultados se mostrarán en forma de tabla indicando la zona de la imagen analizada (CFOV o UFOV) y el valor calculado (UI o UD). Al igual que pasaba en la prueba anterior, si algún valor obtenido sobrepasa la tolerancia especificada en la tabla 6.2, este se mostrara en rojo al usuario.

Para calcular los resultados basta con que el usuario pulse el botón *Calcular*, una vez realizado dicho cálculo se mostrarán los resultados en su tabla correspondiente y aparecerá junto a cada imagen, la que ha resultado durante el proceso.

Inicialmente se abrirá la imagen más reciente, en el caso de que el usuario quiera realizar la prueba de otro día puede o bien pulsar el primer icono de

la barra de tareas o dirigirse a la opción *Abrir imagen* en el menú.

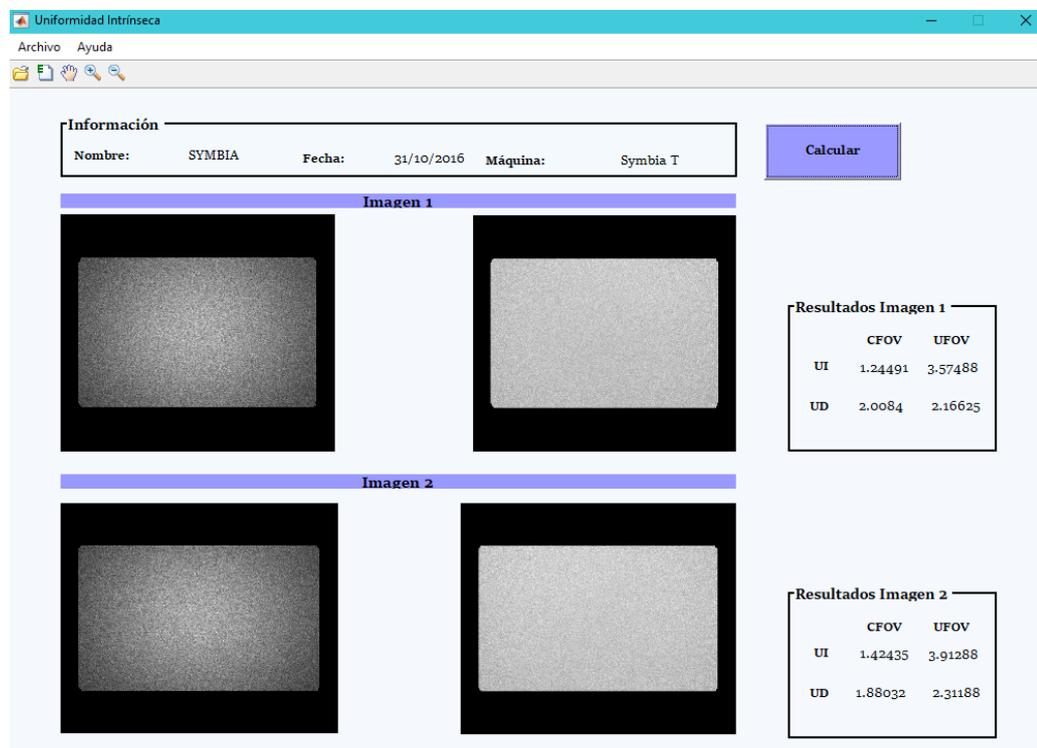


Figura 6.15: Interfaz

La figura 6.15 nos muestra la interfaz una vez se ha realizado una prueba, podemos observar la colocación de los resultados en su tabla correspondiente.

Si el usuario deseara exportar a EXCEL los resultados obtenidos, al igual que, en las demás pruebas, solo tiene que pulsar el icono correspondiente (el segundo en la barra de herramientas) o seleccionar en el menú la opción *Exportar*. Como en la prueba anteriormente comentada el fichero EXCEL al que se exporte dependerá de la máquina detectada por la aplicación; para la máquina Symbia T el fichero será *SymbiaUniformidadIntrínseca.xlsx* mientras que para la máquina E.Cam este será *EcamUniformidadIntrínseca.xlsx*, ambos ficheros tendrán la misma estructura:

Nombre	Fecha	Detector	CFOVUI	CFOVUD	UFOVUI	UFOVUD	Resultado
Intrinsic Verification Flood20161031Detector ver. in.dcm	31/10/2016	1	1,24491	2,008455	3,574885	2,166253	Correcto
Intrinsic Verification Flood20161031Detector ver. in.dcm	31/10/2016	2	1,424347	1,880319	3,912879	2,311878	Correcto

Figura 6.16: EXCEL UniformidadIntrinseca

Como en la anterior prueba tenemos en la primera columna el nombre de la imagen procesada seguido por la fecha de adquisición y el detector

al que se corresponde, a continuación, tenemos un grupo de datos que se corresponde con los resultados calculados durante la ejecución, estos datos son: **CFOVUI**(Uniformidad Integral del CFOV), **CFOVUD**(Uniformidad Diferencial del CFOV), **UFOVUI**(Uniformidad Integral del UFOV) y **UFOVUD**(Uniformidad Diferencial del UFOV). En la última columna se encuentra la etiqueta que indica si los valores obtenidos durante la prueba son o no correctos.

6.3.4. Algoritmo

En esta prueba el objetivo es analizar una imagen en tonos de grises como la figura 6.12 anteriormente mostrada en la página 42. A partir de esta imagen el primer paso a seguir (que no se encuentra en el guion propuesto[2]), es normalizar la imagen con respecto a la distancia de la carga al detector, tras aplicar dicha normalización que no es más que cambiar los valores de los píxeles obtenidos calculando su distancia hacia la fuente, obtenemos la figura 6.17(a).

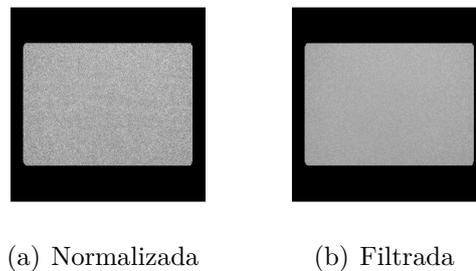


Figura 6.17: Imágenes Adquiridas 1024x1024

A continuación, como la imagen tiene un gran nivel de ruido se aplica el filtro anteriormente comentado 6.13 con la función *imfilter*, así obtenemos una imagen con menos nivel de ruido como vemos en la figura 6.17(b). Ahora y como se especifica en el documento [2] es necesario redimensionar la imagen de 1024x1024 a 64x64, esta redimensión se realiza con la función *imresize*, obteniendo la imagen 6.18(a).



(a) Redimensión



(b) Binaria



(c) Recortada(UFOV)

Figura 6.18: Imágenes Adquiridas 64x64

A continuación, tenemos que quedarnos solo con el CTVU (área total útil de la imagen) por lo que la transformamos a una imagen binaria con la función *im2bw* (figura 6.18(c)), tras esta transformación nos quedamos con el objeto mayor obteniendo sus bordes con *regionprops* utilizando la propiedad *BoundingBox*, con estos bordes recortamos la imagen utilizando la función *imcrop* obteniendo el centro de esta (figura 6.18(c))

A partir de aquí aplicamos el procedimiento explicado en el apartado

anterior recogido en el documento [2] obteniendo la uniformidad integral y diferencial de esta área por medio de las fórmulas y mostraremos los resultados al usuario en la interfaz.

Para obtener los resultados del área central (CCVU) se aplica el mismo algoritmo pero en este caso lo realizamos con el 70 % de la imagen útil, es decir, en lugar de aplicar el algoritmo a la imagen completa como anteriormente, lo aplicamos solo al 70 % del área de la imagen centrada.

6.4. Prueba 3. Resolución Espacial Extrínseca

6.4.1. Objetivo de la prueba

La resolución espacial indica la mínima distancia a la que pueden estar dos fuentes (puntuales o lineales) para poder registrarlas como dos imágenes distintas. La resolución espacial se valora mediante la medida de la anchura a mitad de altura del perfil de cuentas de la imagen de una línea (FWHHM). En nuestro caso, se traza un perfil automático perpendicular a la imagen de una fuente lineal (capilar) alineada en cada uno de los ejes X e Y, para ambos detectores, buscando la zona de mayor actividad de la imagen (mayores valores de píxel). La aplicación ajusta estos valores a una gaussiana y halla su anchura a mitad de altura. También calcula la transformada de Fourier de la misma para obtener la función de transferencia de modulación (MTF) de ambos detectores. Finalmente comprueba que los resultados estén dentro de la tolerancia establecida.

6.4.2. Explicación de la prueba

En el documento de control de calidad de la instrumentación de medicina nuclear [2] se encuentra toda la explicación de dicha prueba, en los siguientes apartados se explicará los conceptos clave necesarios para realizar su implementación.

Adquisición de datos

Esta adquisición es similar a la realizada en Tamaño de Píxel, se coloca la fuente lineal en el CCVU, a 10 cm de la superficie accesible del detector, centrada y alineada con el eje X y en el eje Y como en la figura 6.19.

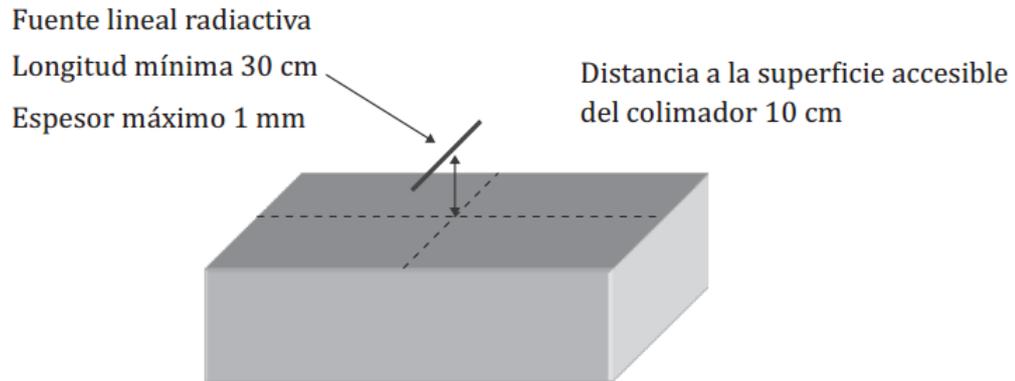


Figura 6.19: Colocación de la fuente lineal respecto de la superficie del detector.[2]

Se obtendrán cuatro imágenes, una por cada detector y eje, como la siguiente:



Figura 6.20: Imagen de Resolución Espacial

Podemos ver en la figura 6.20 la fuente lineal (en blanco), en este caso en el eje X, sobre el fondo negro.

Procesado y análisis de resultado

Una vez obtenida la imagen en el CCVU se dividen en ROIs rectangulares de 30 mm en la dirección paralela a la orientación de la fuente y de, como mínimo, cuatro veces la AIMA esperada en la dirección perpendicular. Para cada ROI se suma el valor de los píxeles de cada fila (o columna) paralela a

la dirección en la que esta colocada la fuente, obteniendo un perfil en forma gaussiana que representa la función de dispersión lineal de la fuente.

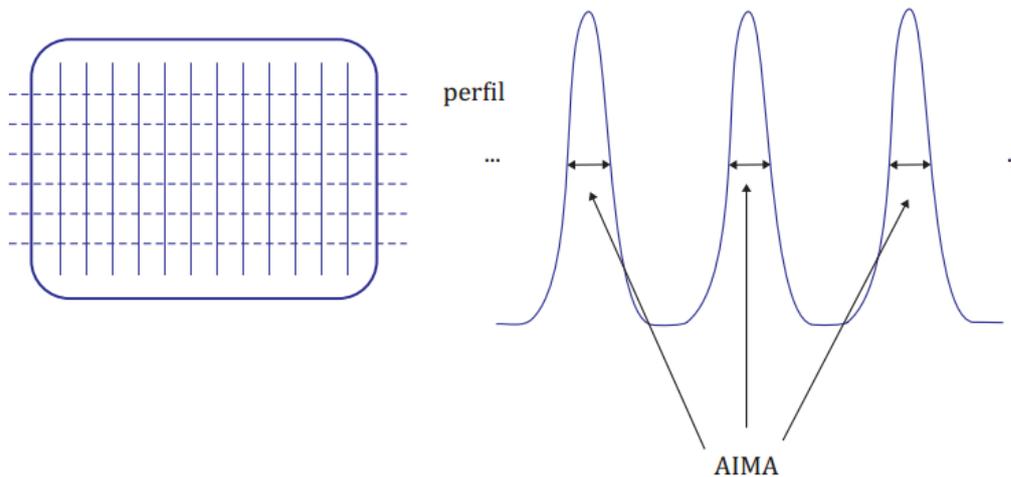


Figura 6.21: Análisis de la imagen: ROIs y perfiles.[2]

Se obtiene la AIMA y la AIDA de la siguiente forma:

- Si el tamaño del píxel es menor o igual que 0.1 AIMA , el píxel con el máximo número de cuentas es el correspondiente al pico.
- Si el tamaño de píxel es mayor que 0.1 AIMA se determina la altura de cada pico mediante un ajuste parabólico de tres puntos entre el valor máximo y los valores contiguos. El máximo de la parábola se toma como el valor del pico.

Una vez hallado el valor del pico, se calcula el valor de su mitad y de su décima parte. A continuación, se obtienen, por interpolación lineal de los puntos vecinos, los puntos de corte de la función de dispersión con la recta que representa la mitad del máximo (o su décima parte, para la AIDA). La distancia entre estos puntos es el valor buscado (figura 6.22)

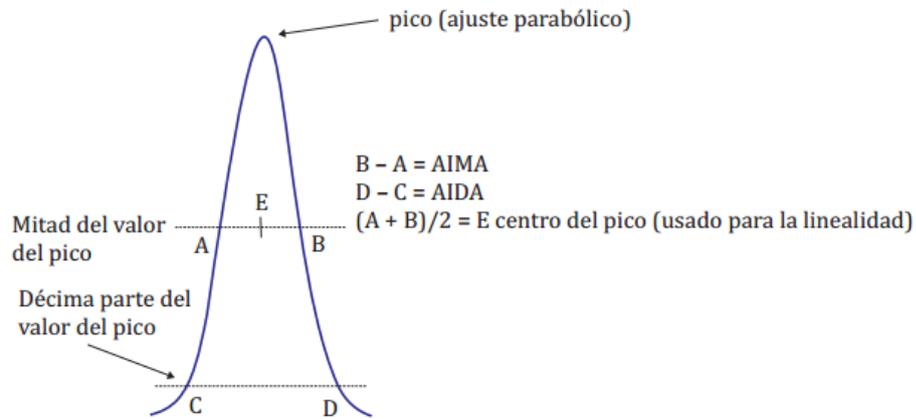


Figura 6.22: Perfil de las cuentas [2]

Si se puede realizar el ajuste a una gaussiana, también se puede calcular la anchura a mitad de la misma. La resolución espacial es el resultado de promediar los valores obtenidos para cada función de dispersión en cada perfil, en X y en Y, expresados en mm y con una aproximación de 0.1mm.

La conversión entre píxeles y milímetros se hará midiendo en píxeles la distancia entre los picos de las dos funciones de dispersión lineal que estén más alejadas dentro de la misma ROI, y dividiendo por este valor la distancia real en mm a la que se encuentran las rendijas correspondientes.

Tolerancias

Los valores obtenidos deben ser menores o iguales que los que el fabricante establece en sus especificaciones técnicas. [11] La resolución espacial, dada como el ancho completo a mitad de altura (FWHA), especificado por el fabricante es 7.8 mm, mientras el ancho a un décimo de altura es 14.9mm.

El valor del FWHA no debe sobrepasar en más de un 5% el valor establecido por el fabricante, debiéndose calibrar el sistema en caso contrario.

6.4.3. Implementación de la prueba

La interfaz de esta prueba se asemeja mucho a la vista en Tamaño de Píxel, las cuatro imágenes se le mostrarán al usuario a la vez, identificando cada una con una etiqueta superior indicando el detector y el eje, inicialmente se cargarán las imágenes de la prueba cuya fecha es la más reciente en el directorio de imágenes de la aplicación, el usuario puede cargar un conjunto

de imágenes nuevas si lo desea utilizando para ello el icono o la opción del menú correspondiente.

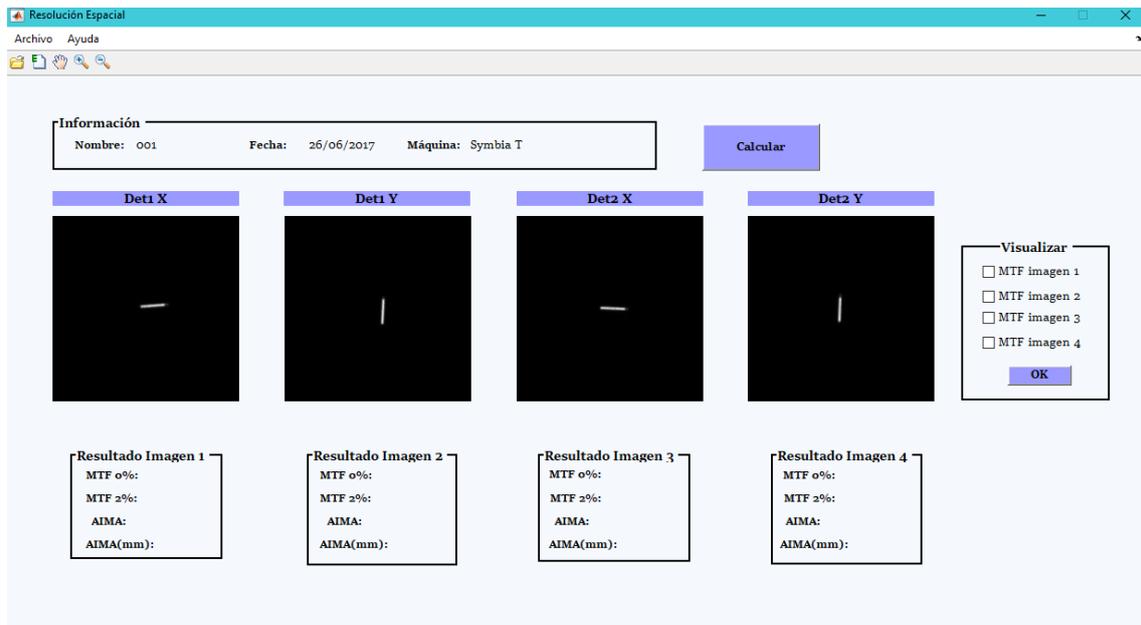


Figura 6.23: Interfaz

En la figura 6.23 podemos apreciar la vista de las cuatro imágenes, al igual que, en las interfaces anteriores, cada imagen tiene un cuadro de resultados asociado donde se mostrarán los valores calculados cuando el usuario pulse el botón calcular. En la parte superior también podemos apreciar el cuadro de información ya conocido de las otras interfaces y la barra de herramientas y menú, que posee los mismos elementos.

La diferencia con las otras interfaces reside en el cuadro que nos encontramos a la derecha de la interfaz, este cuadro sirve para visualizar la gaussiana y la MTF asociada a una imagen, está compuesto por varios *checkbox*, cada uno asociado a una imagen, cuando el usuario selecciona uno de estos *checkbox* puede o bien pulsar el botón *OK* mostrando las dos gráficas asociada a la imagen como vemos en la figura 6.24. Esta opción también es compatible con el botón *Calcular*, en el caso de que el usuario quiera realizar el cálculo y la visualización al mismo tiempo.

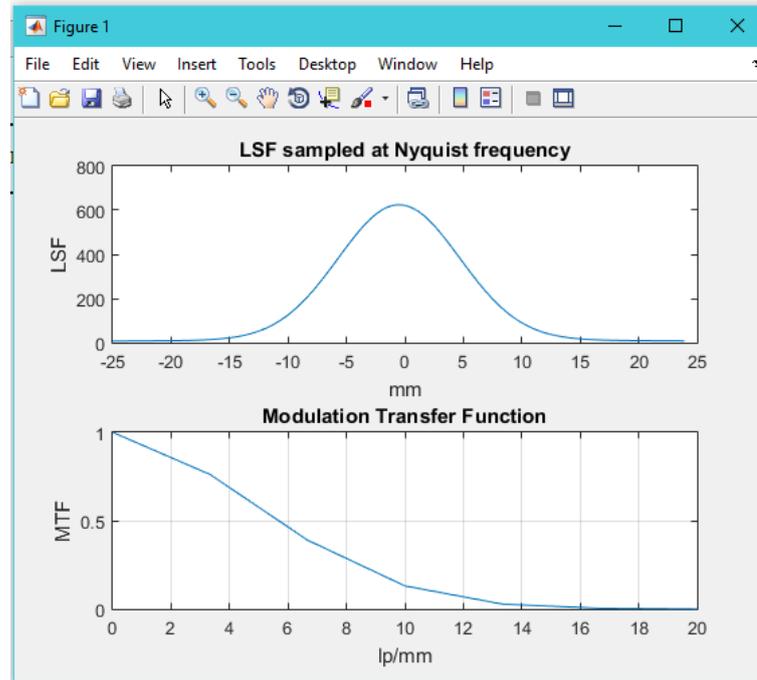


Figura 6.24: Perfil de las cuentas

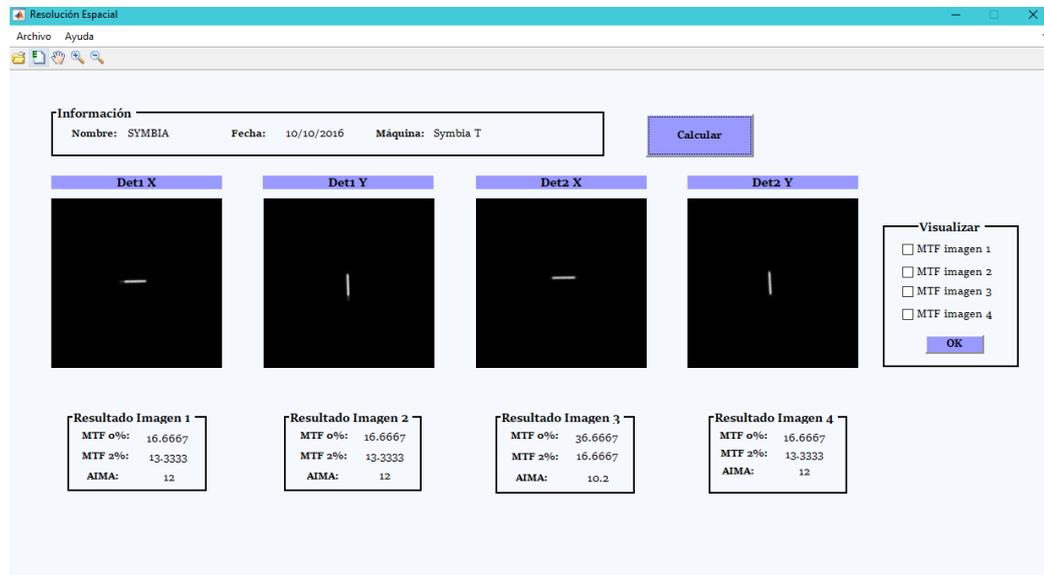


Figura 6.25: Interfaz

En la figura 6.25 podemos ver como se le muestran los resultados al usuario, como se ha comentado anteriormente, estos resultados se mostrarán

en el cuadro correspondiente a la imagen, al igual que en otras pruebas, si estos valores sobrepasaran las tolerancias recomendadas se mostrarán en rojo.

Al igual que en las demás pruebas la exportación a EXCEL se realizará cuando el usuario seleccione o bien el icono asociado a esta acción o la opción del menú correspondiente. El fichero de EXCEL para la máquina Symbia T se denominará *SymbiaResoluciónEspacial.xlsx* mientras que el fichero asociado a la máquina E.Cam se llamará *EcamResoluciónEspacial.xlsx*, ambos ficheros tendran la siguiente estructura similar a los anteriores:

Nombre	Fecha	Detector	MTF1	MTF2	AIMA	AIMAmm	Resultado
Spatial Resolution20161010Det1 X.c	10/10/2016	Det1 X	16,66667	13,33333	12,3	7,38	Correcto
Spatial Resolution20161010Det1 Y.c	10/10/2016	Det1 Y	16,66667	13,33333	12,3	7,38	Correcto
Spatial Resolution20161010Det2 X.c	10/10/2016	Det2 X	16,66667	13,33333	11,4	6,84	Correcto
Spatial Resolution20161010Det2 Y.c	10/10/2016	Det2 Y	16,66667	13,33333	12,3	7,38	Correcto

Figura 6.26: Excel ResolucionEspacial

Tendremos en la primera columna los nombres de las imágenes procesadas, a continuación, tendremos la fecha de adquisición de cada una y el detector correspondiente, los siguientes visibles datos son el valor de la MTF a 0% y al 2% y el valor del AIMA (anchura intrínseca a mitad de altura), el valor del AIMA es importante ya que es el que determinará si la prueba se pasa o no con éxito, los valores de la MTF han sido incluidos por petición del usuario. Por último, al igual que en las pruebas anteriores, disponemos de una etiqueta que indicará si se pasa la prueba o no.

6.4.4. Algoritmo

En este algoritmo, al igual que en los anteriores, se realiza el cálculo sobre un fragmento de la imagen que recortaremos con las técnicas explicadas anteriormente.

Inicialmente tendremos una imagen como la vista en la figura 6.20, para obtener el ROI a computar debemos pasarla a binario con *im2bw*, obteniendo la siguiente figura:

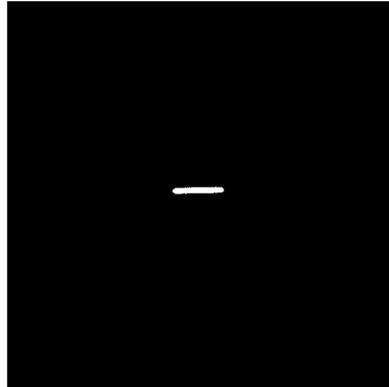


Figura 6.27: Imagen Binaria

En esta figura podemos apreciar claramente la fuente lineal (en blanco) sobre el fondo negro, a partir de esta imagen detectaremos el objeto (fuente) y obtendremos sus bordes con *regionprops* especificando la propiedad *BoundingBoxBounds*. Una vez conocemos los bordes de la fuente, recortamos la imagen con 30 píxeles de margen con respecto a dichos bordes, esta acción nos deja una imagen como la que podemos apreciar en la figura 6.28.

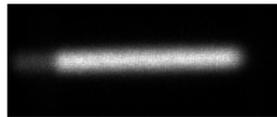


Figura 6.28: Imagen ROI

Ahora que tenemos el ROI, el primer paso es aplicarle a esta imagen la función *improfile* con la que obtendremos un perfil de las cuentas de la imagen, tras esto se calculará la distancia de cada píxel hacia aquel con mayor número de cuentas, una vez obtenido dicho cálculo se aproximará a una gaussiana por medio de la función *fit* utilizando la opción *gauss1*. Por último para calcular la MTF sólo hay que aplicar la función *fft* a la gaussiana anterior.

6.5. Prueba4. Uniformidad Tomográfica

6.5.1. Objetivo de la prueba

La uniformidad tomográfica, evalúa los posibles artefactos producidos en la reconstrucción tomográfica de la imagen.

El estudio adquirido está compuesto por 128 proyecciones. La aplicación diseñada calcula la uniformidad de cada imagen y genera su histograma de niveles de gris, que se intercompara entre todas las imágenes. Finalmente, exporta todos los datos a un a una hoja de Microsoft Excel. Los resultados numéricos obtenidos permiten valorar cuantitativamente el estado de la prueba frente a la valoración subjetiva señalada en [2].

6.5.2. Explicación de la prueba

En el documento de control de calidad de la instrumentación de medicina nuclear [2] se encuentra toda la explicación de dicha prueba, en esta memoria explicaremos los conceptos clave necesarios para realizar su implementación.

Adquisición de datos

Se realizará como en la prueba de Uniformidad Intrínseca, obteniendo 128 cortes en la imagen DICOM como el siguiente:

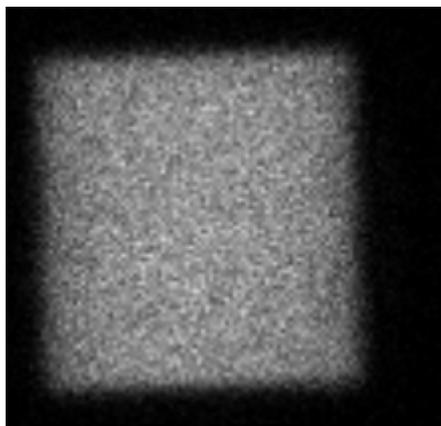


Figura 6.29: Imagen Uniformidad Tomográfica.

Procesado y análisis de resultado

El procesado y análisis de esta prueba lo hace originalmente el usuario a ojo tratándose de una prueba subjetiva, el usuario observa todos los cortes en

busca de posibles artefactos circulares (lo que indicaría que hay un detector dañado). Para ayudar a este análisis, mostraremos el histograma de cada corte y los valores de uniformidad calculados, lo cual ayudará al usuario a comprobar si la imagen observada es o no uniforme.

Tolerancias

Como el análisis de la prueba subjetivo, dicha prueba será incorrecta si el usuario detecta un artefacto en algún corte de la imagen. Con el fin de ayudar al usuario y debido a que no se dispone de unas tolerancias específicas, en nuestra aplicación calculará los valores de uniformidad para cada corte y se realizará la media de estos, si la uniformidad de uno de los cortes superara el valor medio significaría que uno de los cortes tiene un cambio brusco de uniformidad lo cual indicaría un posible error en el detector.

6.5.3. Implementación de la prueba

En esta interfaz, al igual que en las anteriores, disponemos de un menú y una barra de herramientas con los mismos elementos explicados en interfaces anteriores.

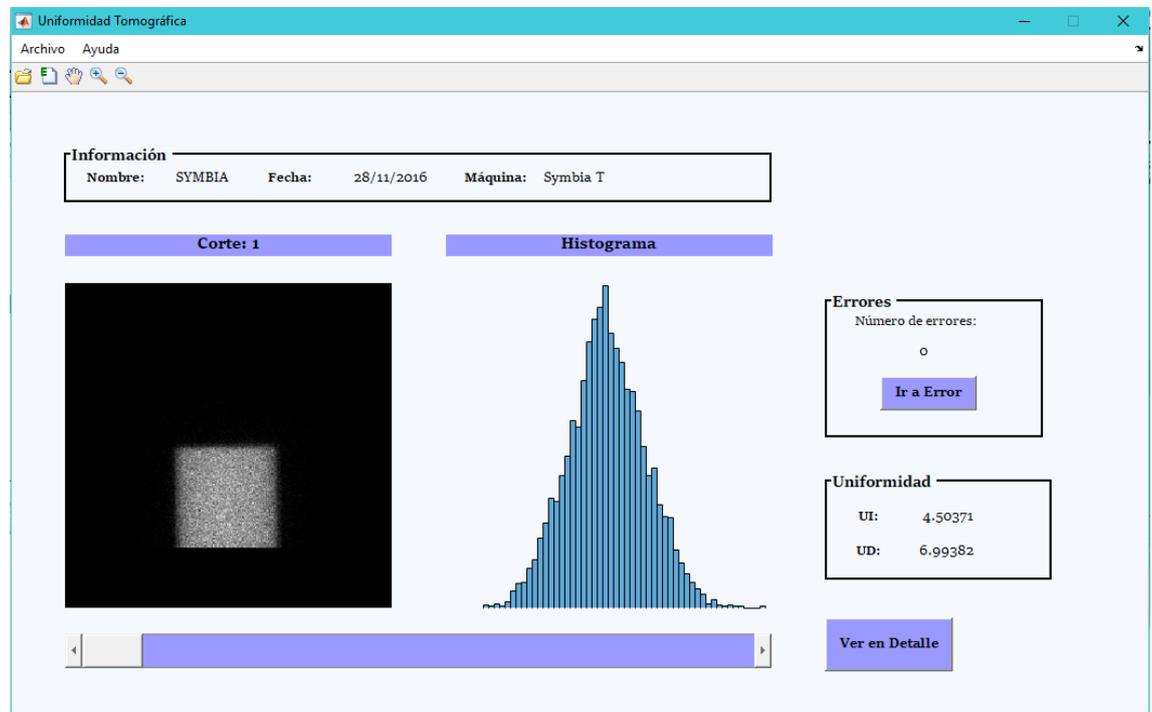


Figura 6.30: Interfaz Uniformidad Tomográfica.

En la figura 6.30 podemos ver la interfaz podemos apreciar un cuadro de información disponible en la parte superior, donde se indica el nombre, fecha y máquina correspondiente a la imagen procesada actualmente. También el usuario verá la imagen con una etiqueta superior indicándole a qué corte pertenece, junto a esta imagen el usuario puede visualizar el histograma de dicho corte. Como en esta prueba se procesa una imagen DICOM con 128 cortes (frames) se optó por incluir un *scroll* por medio del cual el usuario puede variar el corte a visualizar y su histograma asociado.

En esta interfaz no se dispone de un botón *Calcular*, los resultados se muestran al mismo tiempo que la imagen, cuando el usuario varía el corte con el *scroll*, también se mostrarán los resultados asociados a dicho corte, lo que hace la interfaz más cómoda e intuitiva para el usuario. Además, disponemos de un cuadro en la parte superior que indica al usuario el número de errores encontrados y el botón *Ir a Error* redirige al usuario al primer corte en el que se encontró un error.

Como el histograma mostrado en la aplicación tiene una escala considerable, en el caso de que el usuario desee verlo en más detalle bastará con pulsar en el botón *Ver en Detalle* generando una figura como la 6.31.

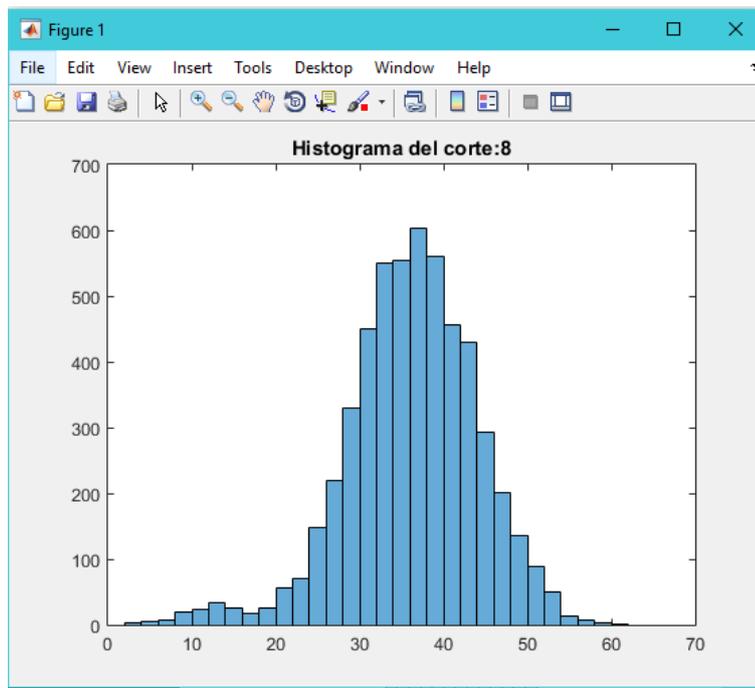


Figura 6.31: Histograma Uniformidad Tomográfica.

Esta figura mostrará el histograma con sus ejes, y un título que indica al

usuario a que corte pertenece.

Al igual que todas las pruebas anteriormente descritas se puede exportar a EXCEL en el fichero *SymbiaUniformidadTomográfica* para la máquina Symbia T o *EcamUniformidadTomográfica* para la Ecam. El fichero EXCEL tendrá la siguiente estructura:

Nombre	Fecha	Uniformidad	Corte1	Corte2	Corte3	Corte4	Corte5
Tomo Uniformity2	31/05/2016	Diferencial	18,02882	21,55139	19,42507	18,18078	20,93472
Tomo Uniformity2	31/05/2016	Integral	15,0462	14,74008	15,17134	14,04199	14,28288

(a) Principio

Corte122	Corte123	Corte124	Corte125	Corte126	Corte127	Corte128	Media
20,91579	18,582	21,43483	17,07324	22,46141	20,02388	19,71911	19,83183
15,00276	13,79382	14,57953	14,25014	14,59351	14,25885	14,88912	14,50189

(b) Final

Figura 6.32: EXCEL Uniformidad Tomográfica

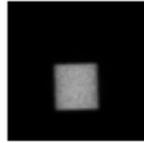
Como podemos ver en las figuras 6.32(a) y 6.32(b) en el fichero EXCEL se escribe el nombre de la imagen junto a cada corte y la uniformidad medida (diferencial o integral). Al final de los 128 cortes se incluye el valor de la media de los valores, el valor medio de los valores de las uniformidades calculados es importante ya que es el que usaremos para estimar si hay algún error en uno de los cortes, en el caso de que se encuentre algún artefacto en el corte este cambiará notablemente el valor de la uniformidad que se distinguirá del resto y no rondará la media.

6.5.4. Algoritmo

El algoritmo de esta prueba es similar al descrito en la Uniformidad Intrínseca sin la normalización con respecto a la distancia del proyector y cambiando la redimensión de 64x64 por una de 128x128.

En primer lugar leeremos la imagen DICOM indicando el corte/frame específico con: *dicomread(filename, 'frames', x)* donde x es el frame indicado (1-128). Las imágenes obtenidas de cada corte serán similares a la vista en la página 57.

A continuación redimensionamos la imagen anteriormente obtenida a 128x128 quedándonos una imagen como la que se puede ver en la figura 6.33(a).



(a) Redimensión



(b) Binaria1



(c) Recortada

Figura 6.33: Imágenes Adquiridas 128x128

Esta imagen la transformamos a binario con *im2bw* (figura 6.33(b)) y nos quedamos con los bordes del objeto que encontremos en ella utilizando como en las otras pruebas *regionprops*. A partir de los bordes recortamos la imagen de 128x128 y realizamos los cálculos a esta imagen recortada que podemos apreciar en la figura 6.33(c).

Capítulo 7

Implementación de la interfaz

En este capítulo explicaremos como está conformada la aplicación, empezado a describir la interfaz generada y como se accede a las distintas herramientas realizadas.

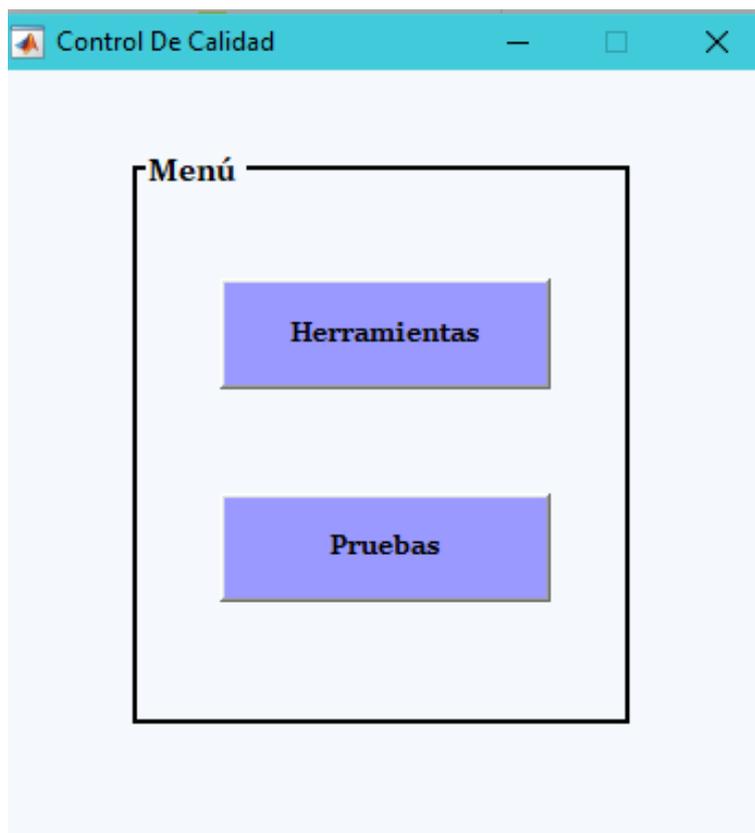


Figura 7.1: Interfaz Principal

Para finalizar la interfaz y como se comentó anteriormente, se unificaron las interfaces anteriormente documentadas en una sola.

La interfaz inicial al abrir la aplicación podemos verla en la figura 7.1, como podemos observar esta interfaz es bastante sencilla, está formada por dos botones, *Herramientas*, el cual abre un conjunto de herramientas que han sido desarrolladas como ayuda al usuario, estas herramientas en concreto son: un conversor de nombres que asocia un nombre más intuitivo a las imágenes relacionándolo con las pruebas, la otra herramienta está dedicada a los históricos de cada prueba. El botón *Pruebas* abre la interfaz donde el usuario podrá realizar las pruebas. Se comentarán a continuación ambas opciones.

7.1. Interfaz Herramientas

La interfaz que se abre cuando el usuario pulsa el botón *Herramientas* podemos verla en la figura 7.2

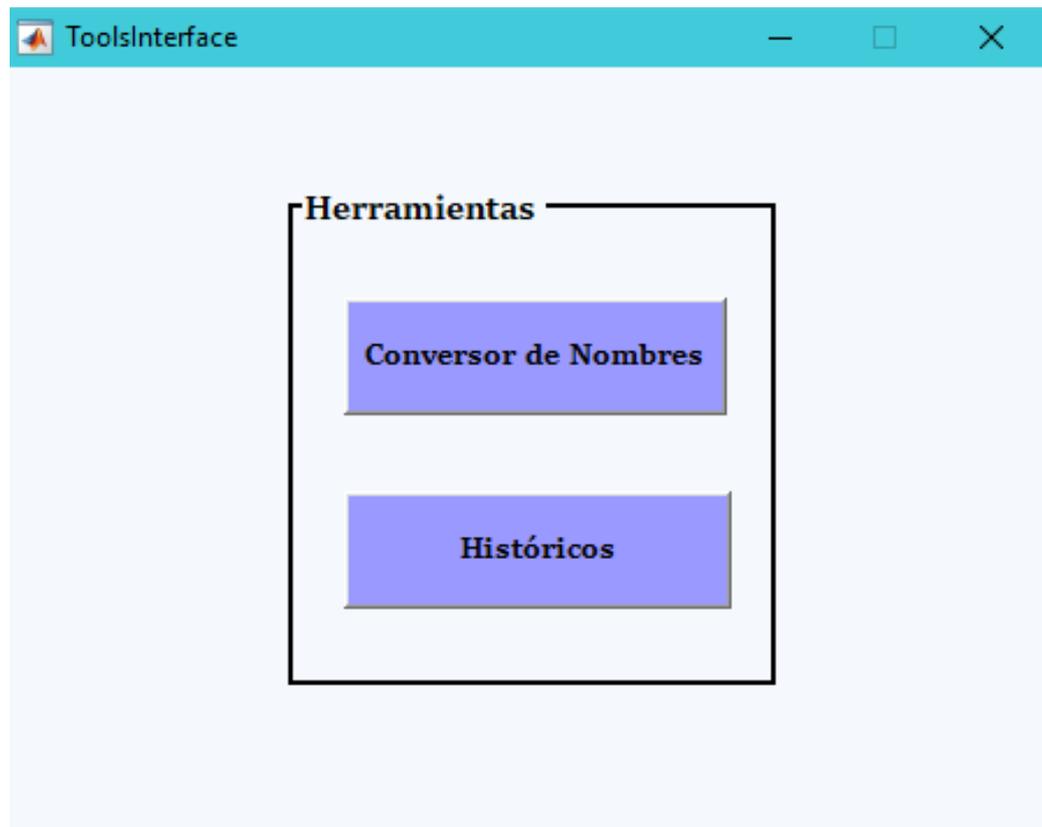


Figura 7.2: Interfaz Herramientas

Como se comentó anteriormente una de las herramientas consiste en un conversor de nombres de imágenes, cuando el usuario pulsa el botón *Conversor de Nombres*, se abre un cuadro de diálogo donde podrá seleccionar la imagen o imágenes a las que desea cambiar el nombre (figura 7.3).

Esta herramienta se desarrolló al encontrarnos con el problema de que las imágenes relativas a cada prueba no siempre tenían el mismo nombre. El criterio que sigue la herramienta para poner nombre es el siguiente:

Nombre de la Prueba+Fecha de Adquisición + Detector-Eje(si corresponde)

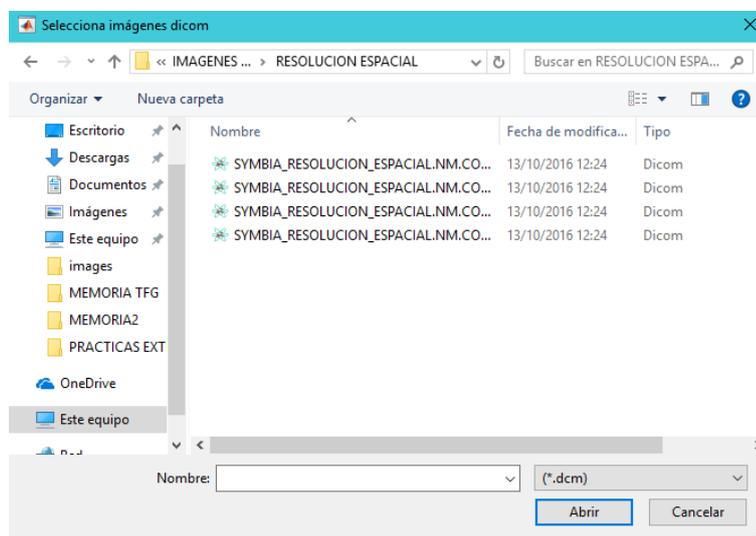


Figura 7.3: Interfaz Cambio Nombre.

La información necesaria la encontramos en los metadatos DICOM, como la prueba a la que pertenece la imagen, la fecha de adquisición de esta, y en el caso de las pruebas *Tamaño de píxel* y *Resolución espacial* obtenemos el detector utilizado y el eje de la imagen.

La otra herramienta desarrollada se trata de los históricos de cada prueba, con el fin de que el usuario tenga una referencias visual a las pruebas realizadas, más intuitiva que los datos almacenados en los ficheros EXCEL.

En esta herramienta por cada histórico tendremos una gráfica, esto se ve reflejado en la interfaz que podemos ver en la figura 7.4, se aprecian cuatro botones, uno para cada prueba implementada.

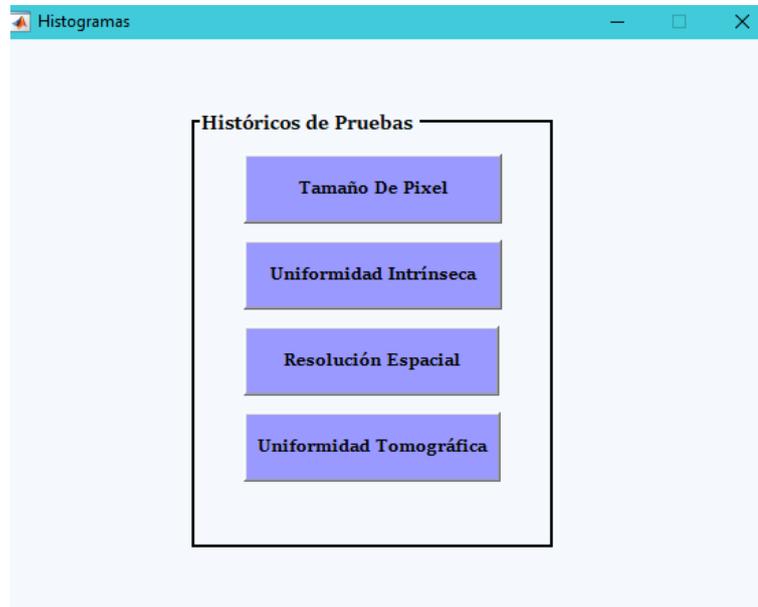


Figura 7.4: Interfaz Históricos

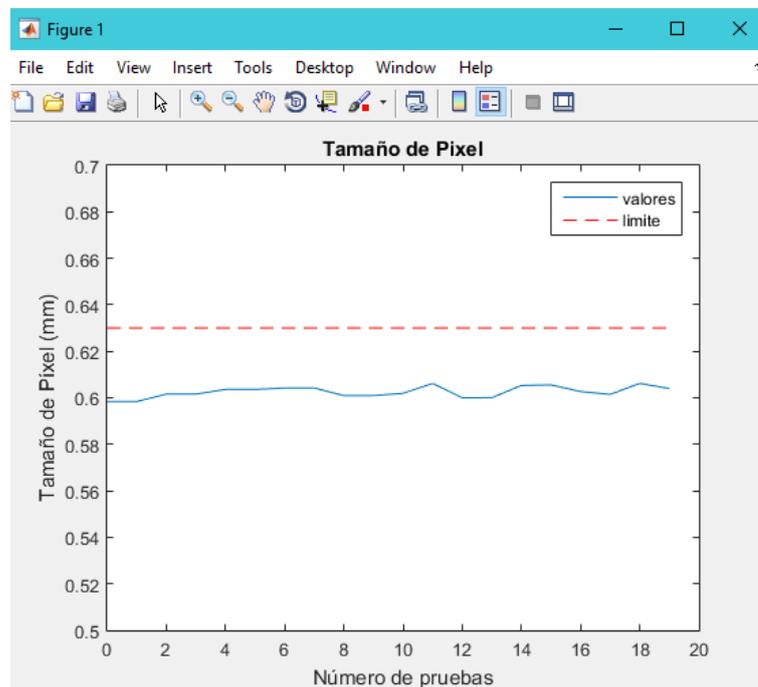


Figura 7.5: Históricos Tamaño Pixel

Las gráficas generadas de las pruebas *Tamaño De Pixel* y *Resolución*

Espacial, serán similares, y consistirán en el ploteado de los datos obtenidos en cada prueba, en rojo tendremos un indicador de la tolerancia límite de la prueba. Como ejemplo tenemos el histórico generado para la prueba *Tamaño de Píxel* en la figura 7.5, donde podemos observar el número de pruebas procesadas en el eje X, mientras que en el eje Y encontramos los resultados del tamaño de píxel en mm calculados en la realización de la prueba.

En el caso de la prueba *Uniformidad Intrínseca*, el sistema de generación de gráficas es el mismo, salvo que en esta caso tendremos cuatro gráficas, correspondientes al par *Parámetro calculado/Área de cálculo*, esto podemos verlo en la figura 7.6.

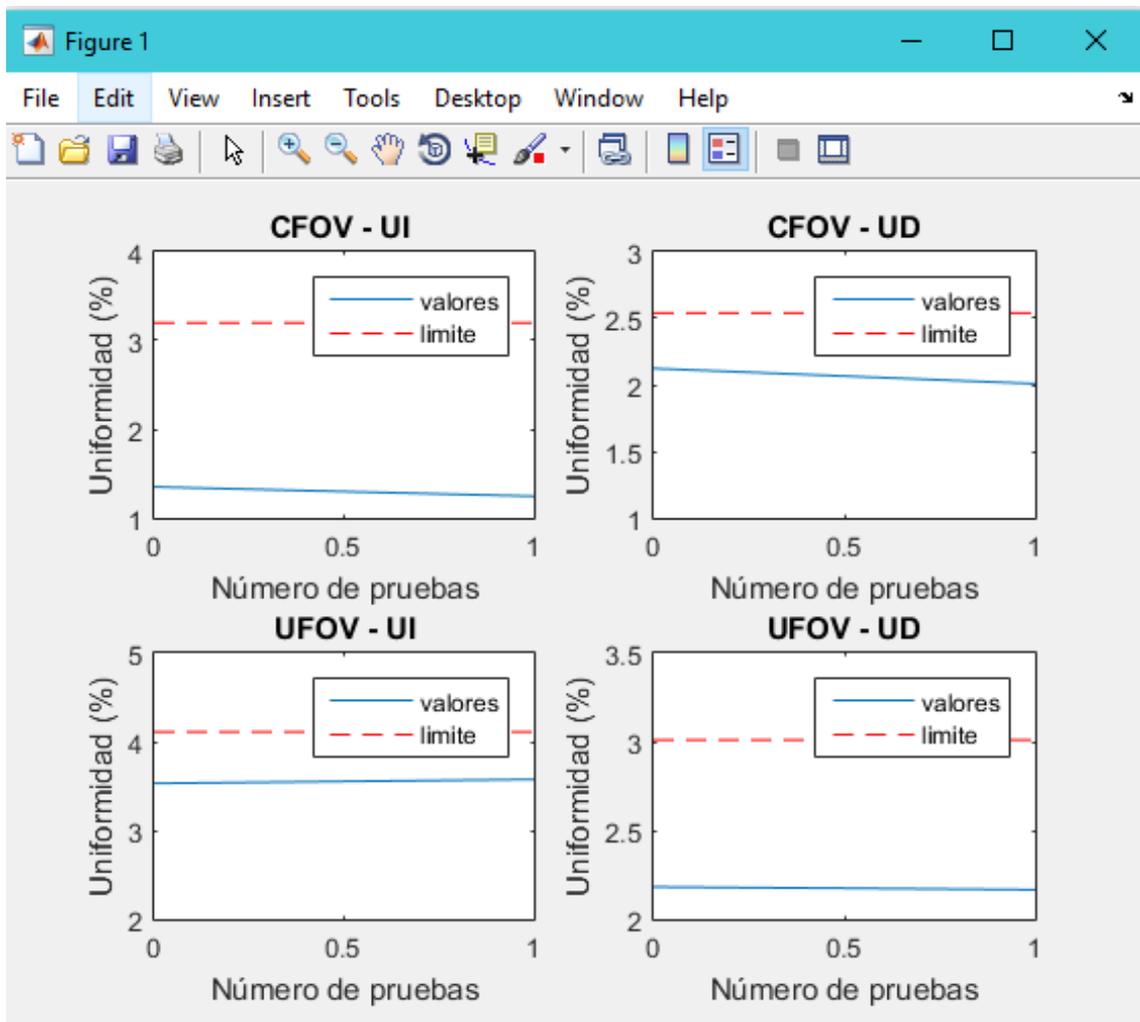


Figura 7.6: Histórico Uniformidad Intrínseca

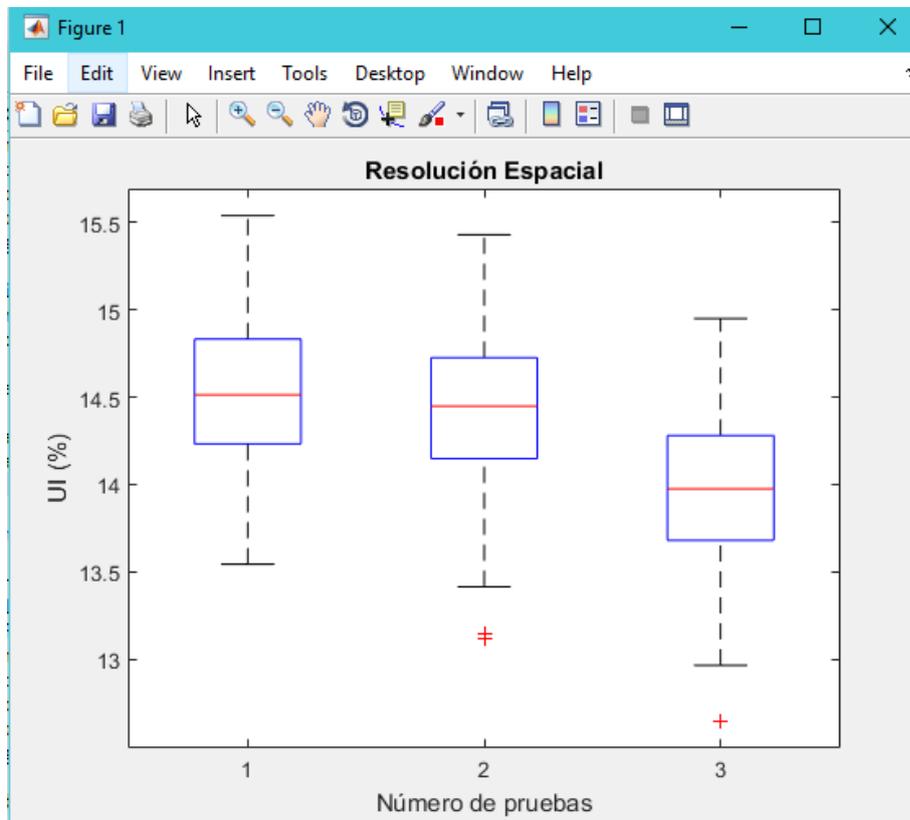


Figura 7.7: Histórico Uniformidad Tomográfica

Por último, para la prueba *Uniformidad Tomográfica*, utilizamos un tipo de gráfica distinto ya que queremos observar la uniformidad de los datos, por lo que utilizaremos en este caso la función *boxplot*, en esta gráfica podemos ver la media de los datos en rojo y su distribución..

7.2. Interfaz Pruebas

La interfaz que recoge las pruebas anteriormente explicadas tiene el siguiente aspecto:

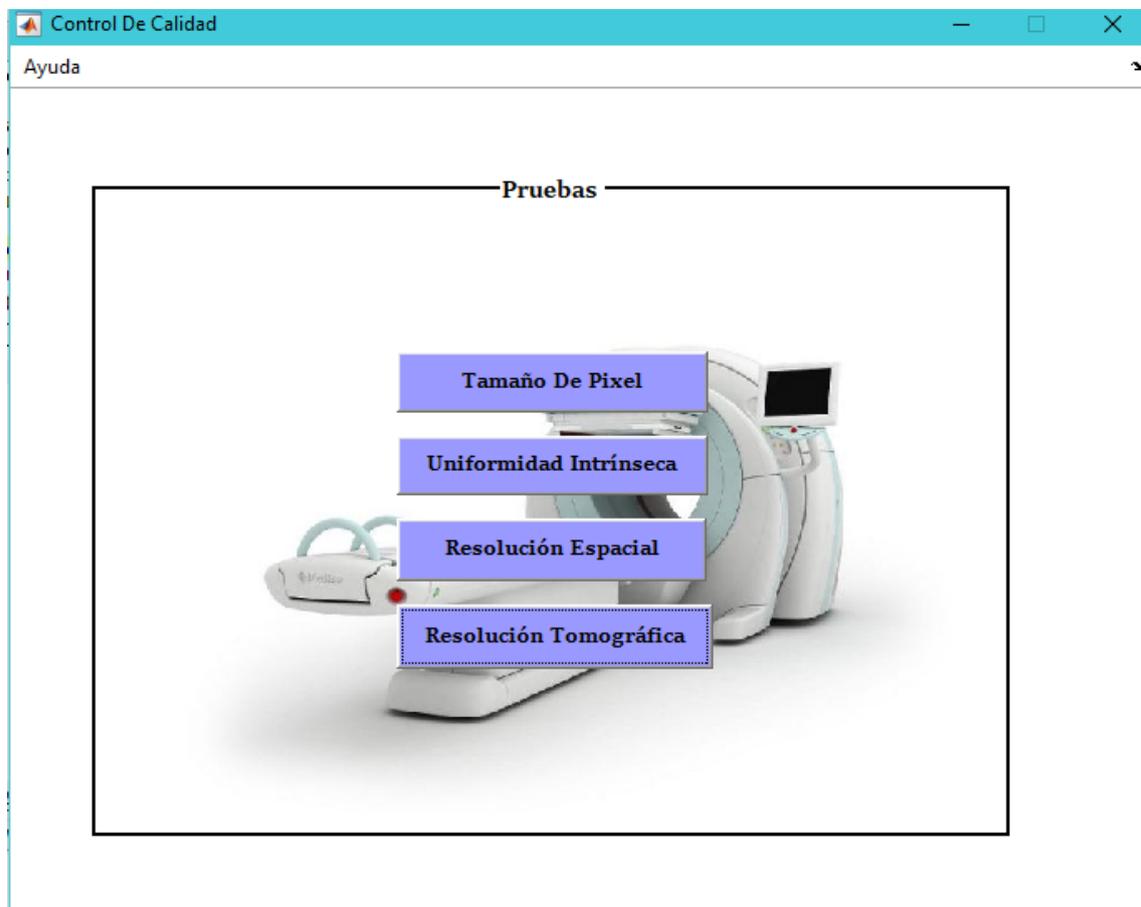


Figura 7.8: Interfaz Pruebas

Como podemos ver se trata de una interfaz sencilla, formada por un menú de ayuda donde el usuario encontrará una breve descripción de la prueba a realizar y unos botones, cuando el usuario pulse sobre uno de los botones se abrirá la interfaz de la prueba deseada.

Capítulo 8

Resultados

En éste capítulo analizaremos los resultados obtenidos en la implementación de cada prueba realizada.

8.1. Tamaño de Píxel

Como se especificó en el apartado correspondiente a esta prueba el tamaño de píxel nominal (el suministrado por la casa) es de 0.6mm. Nuestra aplicación da resultados en torno a ese valor, que no sobrepasan la tolerancia requerida en torno al 5%, en la figura 8.1 tenemos un ejemplo de resultado en una ejecución de nuestra aplicación, se puede distinguir dos valores, el primero es la Distancia calculada entre las dos cargas en número de píxeles, el otro resultado que es el que más interesa al usuario es el tamaño de un píxel en el cristal medido en mm.

Resultados	
Distancia:	330.676
Tamaño Pixel:	0.6048

Figura 8.1: Resultados Tamaño Píxel

Otra forma de comprobar el correcto resultado del tamaño de píxel es teniendo en cuenta la distancia, la distancia entre las cargas en el maniquí utilizado para la obtención de las imágenes es de 200 mm, si convertimos el valor en número de píxeles calculado (330.676 en nuestro ejemplo) a mm utilizando el tamaño de píxel calculado (0.6078 en el ejemplo), obtendremos un valor cercano a 200 mm (199.99 en el ejemplo utilizado en este caso).

8.2. Uniformidad Intrínseca

En esta prueba se nos da una tabla con los resultados que debe dar la prueba dependiendo de la zona de la imagen tratada (tabla 6.1).

En la siguiente figura tenemos un ejemplo de los resultados de nuestra aplicación:

Resultados Imagen 1		
	CFOV	UFOV
UI	1.24491	3.57488
UD	2.0084	2.16625

Figura 8.2: Resultados Uniformidad Intrínseca

Como podemos apreciar comparando ambas tablas, los valores calculados por nuestra aplicación no se acercan tanto a los proporcionados por el fabricante, esto se debe a que la imagen que procesamos tiene mucho ruido y no solo hay que suavizarla por medio de filtros, sino que hay que normalizar el valor de los píxeles en función a la distancia a la que se situó la carga del detector y además hay que redimensionar la imagen de 1024x1024 a 64x64 perdiendo más calidad. Aun así los valores calculados entran dentro de las tolerancias admitidas especificadas en la prueba.

8.3. Resolución Espacial

En esta prueba el fabricante especifica que el valor del AIMA (FWHM) es de 7.8 mm, como vemos en la 8.3 nuestra aplicación da valores similares a este.

Resultado Imagen 1	
MTF 0%:	16.6667
MTF 2%:	13.3333
AIMA:	12.6
AIMA(mm):	7.56

Figura 8.3: Resultados Resolución Espacial

En la figura 8.3, también podemos apreciar la MTF al 0% y al 2%, estos valores fueron pedidos expresamente por el usuario. En la anterior figura también podemos comprobar que el valor dado en mm es correcto al pasar el valor en número de píxeles a mm ($12.6 * 0.6 = 7.56$ mm).

8.4. Uniformidad Tomográfica

La valoración del resultado de esta prueba es más compleja ya que es una valoración que ha de realizar el usuario subjetivamente. En esta prueba se realizan 128 cortes, si hay algún error en uno de los detectores en la imagen se apreciarían objetos circulares, estos objetos variarían notablemente el valor de la uniformidad de un corte con respecto a los demás. La forma de valorar si esta prueba contiene algún error es realizando la media de todos los valores de uniformidad integral calculados para cada corte, y comprobando que estos no sobrepasan a la media un tanto por ciento.

En conclusión los resultados hallados son similares los calculados por el software de la casa comercial suministradora de la gammacámara, excepto en el caso de la Uniformidad Intrínseca que al encontrarnos con una imagen con mucho ruido, estos resultados varían bastante a los obtenidos en la máquina. En aquellas pruebas de valoración subjetiva, la aplicación ayuda a optimizar el proceso de evaluación.

8.5. Resultados académicos

Como resultado académico, cabe destacar que no sólo se realizaron los objetivos inicialmente propuestos, sino que durante la realización del proyecto adquirí nuevas habilidades como el manejo más profundo en la herramienta MATLAB, el conocimiento en el área de procesamiento de imágenes médicas sobre todo aquellas captadas por medio de la medicina nuclear.

Capítulo 9

Conclusiones

En este capítulo se describirán el conjunto de conclusiones de este proyecto y el trabajo futuro.

En primer lugar la aplicación consigue simplificar el trabajo notablemente, ya que puede usarse desde un ordenador cualquiera (que tenga MATLAB), sin la necesidad de utilizar la consola de las gammacámaras, eliminando las dependencias con el software proporcionado. Además como la interfaz es capaz de generar resultados tanto para la máquina Symbia T como para la E.Cam, evita que el usuario tenga que desplazarse entre ambas máquinas para realizar las pruebas, teniendo un sistema más centralizado con el que poder hacer las dos, separando en todo momento los resultados de las dos máquinas en sus correspondientes ficheros de EXCEL.

Otra consecuencia es que al tratarse de una aplicación abierta y modificable, se puede no solo añadir otras pruebas de control de calidad a la interfaz, sino también obtener otros datos de interés en las pruebas como en el caso de la Resolución Espacial donde el usuario además del valor del AIMA para contrastar la prueba pidió que se calculara el valor de la MTF en distintos puntos.

Como este proyecto comenzó en las prácticas de empresa, como consecuencia los objetivos inicialmente propuestos se alcanzaron dentro de la temporalización establecida.

9.1. Aportaciones

Este proyecto aporta a nuestro sector científico, un avance en nuestro sector sanitario aunando las técnicas informáticas con la investigación médica, esta clase de proyectos no solo pueden ayudar a la investigación sino también a hacer crecer el sector sanitario y dotarlo de ayudas que hagan más

sencillo el trabajo de los técnicos e incluso el de los médicos por medio de herramientas de ayuda al diagnóstico.

Otra aportación es la presentación de este proyecto en el V Congreso Conjunto SEFM-SEPR celebrado en Girona, y se encuentra recogido en el libro generado en dicho congreso con ID:01104.[1]

En conclusión, es importante que sectores tan importantes como el sanitario posean un equipo de investigación que sean capaces de generar aplicaciones informáticas útiles para el personal hospitalario, como el estudio de nuevas técnicas que avancen investigaciones en el sector.

9.2. Trabajo Futuro

Como trabajo futuro como se ha comentado anteriormente se pueden integrar otras pruebas de control de calidad a la aplicación. Además estas técnicas de procesamiento de imágenes se pueden reutilizar para otro tipo de aplicaciones enfocadas no solo al control de calidad de otras máquinas sino también a la ayuda al diagnóstico por medio de dichas máquinas.

Apéndice A

Manual de Usuario

A continuación, se describirá como usar nuestra aplicación.

A.1. Iniciar aplicación

Para utilizar esta aplicación es necesario MATLAB R2015a, la ejecutaremos desde su propio directorio. Para abrirla se ejecutará el fichero *MainInterface.c*, que nos abrirá la aplicación como podemos ver en la figura 7.1 en la página 63.

Una vez abierta nuestra aplicación podemos seleccionar si queremos utilizar las herramientas con el botón *Herramientas* o si deseamos empezar a realizar las pruebas de control de calidad pulsando el botón *Pruebas*.

A.2. Herramientas

Si pulsamos el botón *Herramientas* en la interfaz nos dirige a otra que podemos ver en la figura 7.2, en este caso tenemos otras dos opciones, o bien utilizar en conversor de nombres de imágenes pulsando *Conversor De Nombres* o dirigirnos a los históricos de las pruebas con *Históricos*.

A.2.1. Conversor de Nombres

El conversor de nombre de imágenes es una herramienta sencilla de utilizar, basta con pulsar sobre el botón *Seleccionar imágenes*, tras esto se nos abre un diálogo en el que podemos seleccionar las imágenes a las que deseamos cambiar el nombre, les cambia el nombre por uno compuesto por la prueba, la fecha de adquisición y el detector y eje si fuera necesario. Sólo cambia el nombre, no mueve las imágenes a otro directorio.

A.2.2. Históricos

En esta interfaz disponemos de cuatro botones uno para cada prueba, al pulsar sobre uno de ellos se abrirá una gráfica como la que vimos en la figura 7.5, se mostrará la evolución de los valores obtenidos a lo largo del tiempo en la prueba deseada, además se comparará con la tolerancia que vendrá especificada por una línea roja.

A.3. Pruebas

En esta interfaz tenemos un botón por cada prueba que se puede realizar.

A.3.1. Tamaño De Píxel

En esta interfaz como hemos descrito anteriormente se calculará la distancia entre dos cargas puntuales. El uso es simple, se carga las imágenes de la prueba que se desea procesar, y se pulsa sobre el botón *Calcular*, tras esto se mostrarán los datos en el cuadro de resultados correspondiente. Si el usuario desea exportar los resultados, basta con pulsar el icono de exportar a EXCEL (figura A.1).



Figura A.1: Icono EXCEL

A.3.2. Uniformidad Intrínseca

Al igual que la prueba anterior, se cargará la imagen (ya que en este caso se utiliza solo una), para calcular se pulsará el botón *Calcular* y para exportar los resultados el icono correspondiente.

A.3.3. Resolución Espacial

Esta interfaz es igual que la de la prueba Tamaño de Píxel, se cargan las cuatro imágenes que queremos procesar se calcula con el botón *Calcular* y se puede exportar a EXCEL pulsando en su icono.

A.3.4. Uniformidad Tomográfica

Esta interfaz difiere de las demás, al igual que el resto se cargará la imagen a procesar, esta imagen tiene 128 cortes, para representarlos todos tenemos

un scroll, cada vez que avanzamos en el scroll lo hacemos en los cortes de la imagen, además no es necesario pulsar ningún botón para calcular el resultado ya que se calcula automáticamente al movernos por el scroll. Al lado de la imagen tendremos un histograma correspondiente a ella, si quisiéramos verlo en más detalle solo hay que pulsar el botón *Ver en Detalle* que nos muestra una figura con el histograma. Si queremos exportar los datos, pulsaremos sobre el icono correspondiente al igual que en las anteriores interfaces.

Apéndice B

Versiones anteriores

Durante la realización de este proyecto, se han desarrollado varias versiones de la aplicación, a continuación, explicamos alguna de estas versiones.

La prueba que más versiones tuvo es sin duda *Tamaño de Píxel*, al tratarse de la primera interfaz a desarrollar. En primeras versiones de esta prueba, sólo se cargaba una imagen mostrando al usuario los dos cortes (captados por los detectores), como podemos ver en la figura B.1.

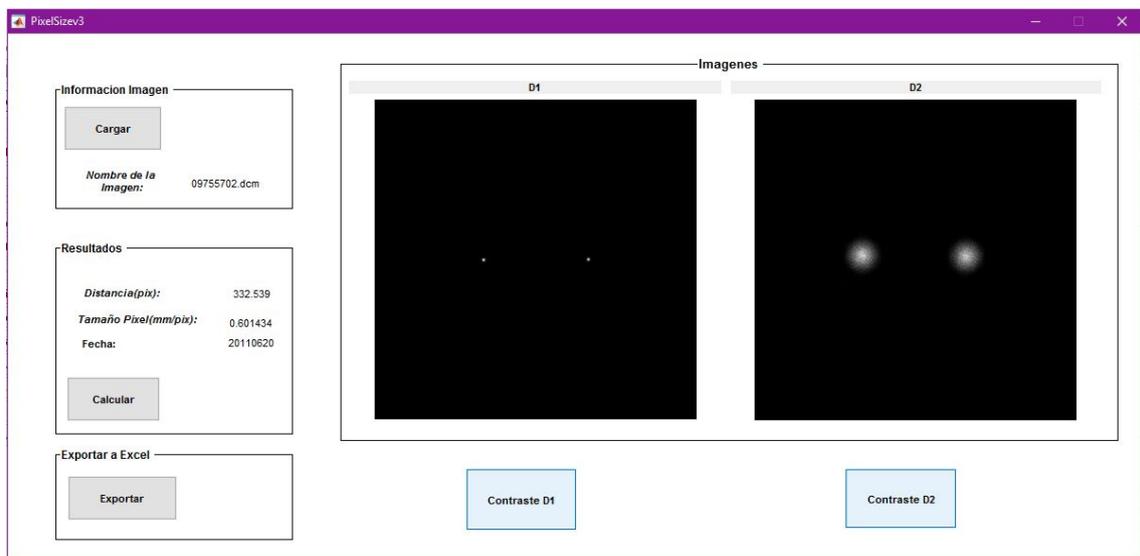


Figura B.1: Versión 1 Tamaño de píxel

La carga de las imágenes de una en una no sólo dificultaba la realización de la prueba al usuario, al tratarse de una prueba que necesita realizar los cálculos a cuatro imágenes, sino que desembocaba en una tarea tediosa para este al tener que ir cargando las distintas imágenes. También se pueden ver

debajo de cada imagen dos botones de *Contraste*, esto se debe a que en esta versión la imagen no realizaba el contraste internamente como en las versiones más recientes, por lo que lo primero que veía el usuario era una imagen completamente en negro. Además de las diferencias con la carga de imágenes, en esta versión se encuentran todas las opciones en la propia interfaz por medio de botones, no posee un menú ni una barra de herramientas que las ordene.

En posteriores versiones, se arregló la carga de imágenes, dando al usuario la opción de cargar cuatro imágenes a la vez, y se limpió la interfaz eliminando los botones innecesarios y creando menús y barra de herramientas que recogieran las posibles tareas que podía realizar el usuario.

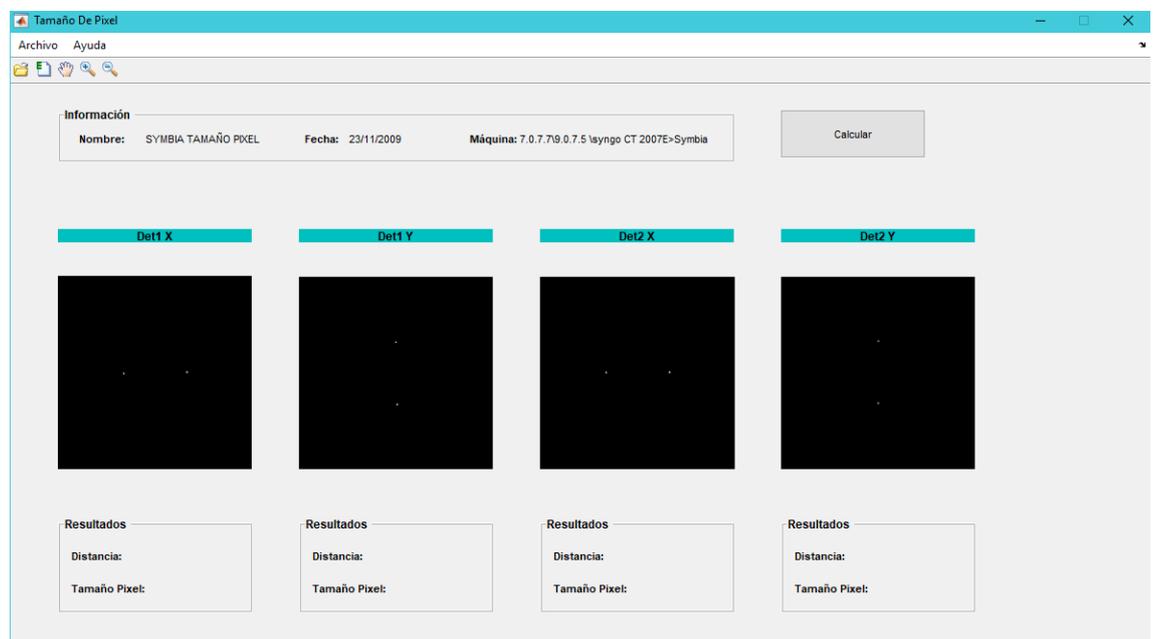


Figura B.2: Versión 2

Aun así, en estas versiones no se cargaba automáticamente la imagen más reciente. Esta mejora se incluyó en versiones finales, donde también se mejoraron los colores y fuentes de la interfaz, como en la figura 6.3.

Otra interfaz que sufrió muchos cambios fue la *Resolución Espacial*, debido a la necesidad del usuario de utilizar gráficas, como en las primeras versiones el botón calcular no solo calculaba los resultados y los mostraba en su cuadro correspondiente sino que también mostraba la gráfica asociada a la imagen, se optó por poner un botón de cálculo para cada imagen evitando que se generaran cuatro figuras al mismo tiempo.

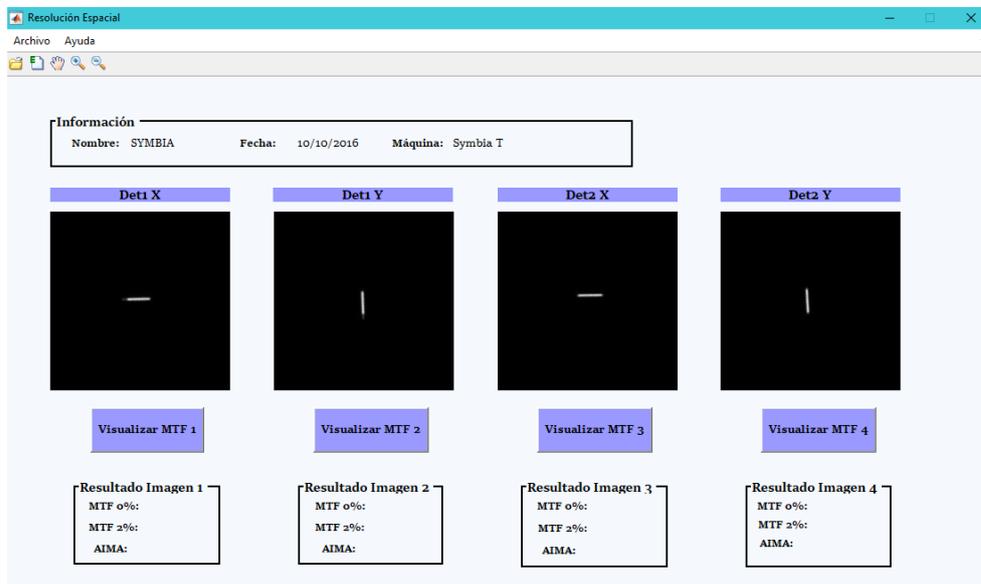


Figura B.3: Versión 2 Resolución Espacial

En las versiones más recientes este problema se soluciona por medio de los checkbox. (figura 6.23).

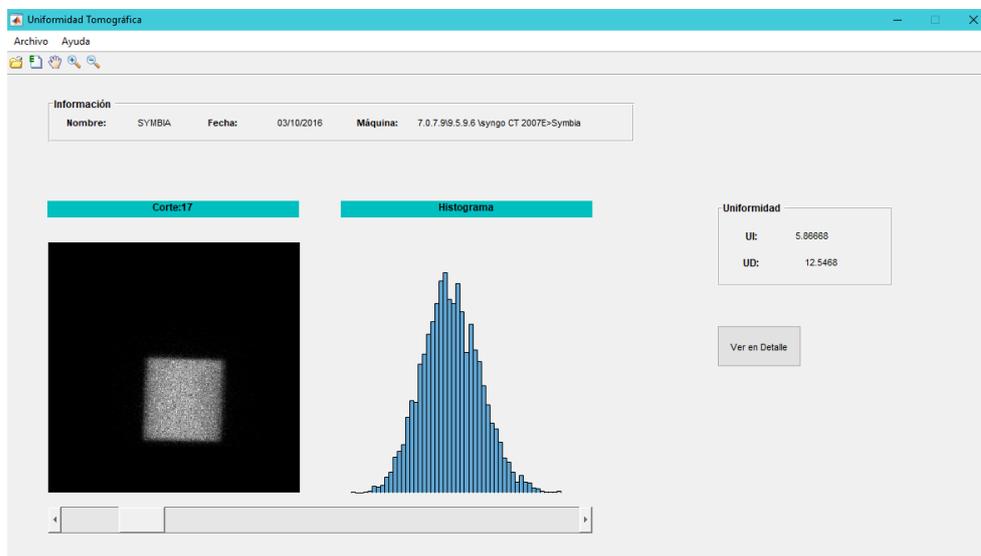


Figura B.4: Versión 1 Uniformidad Tomográfica

La interfaz *Uniformidad Tomográfica* en versiones iniciales no poseía el cuadro de información de errores (figura B.4).

Bibliografía

- [1] *V Congreso Conjunto SEFM-SEPR. La radiación: progreso y salud.*
- [2] *Control de calidad de Instrumentación de Medicina Nuclear, SEFM-SEMnim-SEPR. 2005.*
- [3] Craig S. Lent. *Learning to Program with MATLAB- Building GUI Tools.*
- [4] MATLAB. *Curve Fitting Toolbox.*
- [5] MATLAB. *Data Import and Export.*
- [6] MATLAB. *Image Processing Toolbox.*
- [7] N. Otsu. *A Threshold Selection Method from Gray-Level Histograms.* IEEE Transactions on Systems, Man, and Cybernetics, Vol. 9, No. 1, 1999.
- [8] R.E. Woods R.C. González. *Digital Image Processing.*
- [9] R.E. Woods R.C. González. *Digital Image Processing with MATLAB.*
- [10] Siemens. *e.cam Signature Series.*
- [11] Siemens. *Symbia T Series, System Especifications.*
- [12] P. Suetens. *Fundamentals of Medical Imaging.* Cambridge, 2009.