



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



Escola Tècnica
Superior d'Enginyeria
Informàtica

Escola Tècnica Superior d'Enginyeria Informàtica
Universitat Politècnica de València

Filtrado de artefactos en imágenes médicas de TAC cuando se emplean pocas proyecciones.

TRABAJO FIN DE GRADO

Grado en Ingeniería Informática

Autor: Joan Climent Ferrero

Tutor: Vicente E. Vidal Gimeno

Director Experimental: Mónica Chillarón Pérez

Curso 2017-2018

Agradecimientos

Quiero mostrar mi agradecimiento a Vicent Vidal, mi tutor, por sus enseñanzas y por sus orientaciones, y, como no, por su aliento para abordar este trabajo. Así como a Mónica Chillarón por su paciencia y por su dedicación.

También quiero agradecer a mis padres por su constante apoyo.

Resum

Quan s'adquireixen els sinogrames i a més es redueixen el número de projeccions per a la reconstrucció d'imatges mèdiques de TC, aquestes poden presentar artefactes, sent necessari eliminar la major quantitat de soroll sense alterar la informació que conté la imatge.

El present treball està centrat en l'avaluació de tècniques de filtrat aplicades en l'eliminació d'artefactes presents en imatges mèdiques de tomografies computades, en particular en l'eliminació de soroll impulsiu y gaussià.

Les tècniques de filtrat escollides, s'han analitzat utilitzant quatre funcions dissenyades per a mesurar la qualitat de la imatge filtrada y comparar-la amb una imatge de referència. Es aleshores quan s'extrauen els paràmetres ideals per a realitzar novament el procés de filtrat, previ al procés de reconstrucció de les imatges.

En aquest moment, en base als resultats obtinguts, es pot determinar la viabilitat sobre la utilització dels corresponents mètodes de filtrat.

Paraules clau: Imatge mèdica TC, Tècniques de Filtrat, Soroll Gaussià, Soroll Speckle, MATLAB

Resumen

Cuando se adquieren los sinogramas y además se reduce el número de proyecciones para la reconstrucción de imágenes médicas de TC, éstas pueden presentar artefactos, siendo necesario eliminar la mayor cantidad de ruido sin alterar en exceso la información que contiene la imagen.

El presente trabajo está centrado en la evaluación de técnicas de filtrado aplicadas en la eliminación de artefactos presentes en imágenes médicas de tomografías axiales computarizadas, en particular en la eliminación de ruido impulsivo y gaussiano.

Las técnicas de filtrado elegidas, se han analizado utilizando cuatro funciones diseñadas para medir la calidad de la imagen filtrada y compararla con una imagen de referencia. Es entonces cuando se extraen los parámetros ideales para realizar nuevamente el proceso de filtrado, previo al proceso de reconstrucción de las imágenes.

En ese momento, en base a los resultados obtenidos, se puede determinar la viabilidad sobre la utilización de los correspondientes métodos de filtrado.

Palabras clave: Imagen médica TC, Técnicas de Filtrado, Ruido Gaussiano, Ruido Speckle, MATLAB

Abstract

When the sinograms are acquired and the number of projections used for the reconstruction of medical images is reduced, artifacts will appear on the reconstructions. Thus, it is necessary to eliminate the greater amount of noise possible without altering in excess the information that the image contains.

The present work is focused on the evaluation of several filtering techniques applied in the elimination of artifacts present in computerized tomography medical images, in particular in the elimination of gaussian and impulsive noise.

The chosen filtering techniques have been analyzed using four functions designed to measure the quality of the filtered image and compare it with a reference image. It is then when the ideal parameters are extracted to carry out the filtering process again, prior to the process of reconstruction of the images.

At that time, based on the results obtained, we can determine the feasibility of using the corresponding filtering methods.

Key words: CT medical imaging, Filtering Techniques, Gaussian Noise, Speckle Noise, MATLAB

Índice general

Índice general	VII
Índice de figuras	IX
Índice de tablas	IX
<hr/>	
1 Introducción	1
1.1 Motivación	1
1.2 Estado de la cuestión	2
1.3 Objetivos	3
1.4 Estructura de la memoria	3
2 Materiales y métodos	5
2.1 Reconstrucción de imagen TC	5
2.1.1 Datos de entrada	7
2.1.2 Método iterativo de reconstrucción	8
2.2 Parámetros de calidad	9
2.2.1 SSIM – índice de similitud estructural	9
2.2.2 PSNR – proporción máxima de señal a ruido	9
2.3 Técnicas de filtrado	9
2.3.1 Filtro gaussiano	9
2.3.2 Filtro de mediana	10
2.3.3 Filtro de Wiener	10
2.3.4 Filtro bilateral	11
3 Estudio de los métodos de filtrado	13
3.1 Filtrado del sinograma	13
3.1.1 Función filtro gaussiano	15
3.1.2 Función filtro de mediana	16
3.1.3 Función filtro de Wiener	16
3.1.4 Función filtro bilateral	17
3.2 Análisis de los resultados	18
3.2.1 Resultados de la función filtro_b1()	20
3.2.2 Resultados de la función filtro_b2()	20
3.2.3 Resultados de la función filtro_b3()	21
3.2.4 Resultados de la función filtro_b4()	22
3.3 Reconstrucción mediante sinogramas filtrados	24
3.4 Comparación métodos de filtrado sobre el fantoma	25
4 Conclusiones	31
4.1 Trabajos futuros	32
Bibliografía	33
<hr/>	
Apéndice	
A Resultados estudio de parámetros	35
A.1 Tablas de los sinogramas	35
A.1.1 Ruido gaussiano	35

A.1.2	Ruido speckle	41
A.2	Tablas de los fantasmas	48
A.2.1	Ruido gaussiano	48
A.2.2	Ruido speckle	51

Índice de figuras

2.1	Proceso completo de reconstrucción de una imagen de TC (2D)	5
2.2	Adquisición de datos durante el proceso de escaneo. Fuente: [6]	6
2.3	Corte axial en el punto central. Fuente: [11]	7
2.4	Características del escáner virtual utilizado en las pruebas.	8
2.5	Ejemplo filtro mediana ventana 3x3, píxel central.	10
3.1	Sinograma con ruido gaussiano con varianza 0,00001	19
3.2	Sinograma con ruido gaussiano con varianza 0,0005	19
3.3	Sinograma con ruido gaussiano con varianza 0,005	19
3.4	Sinograma con ruido gaussiano con filtrado gaussiano y valor de varianza 0,7.	20
3.5	Sinograma con ruido speckle con filtrado gaussiano y valor de varianza 0,5.	20
3.6	Sinograma con ruido gaussiano con filtrado de mediana y con tamaño de ventana 3.	21
3.7	Sinograma con ruido speckle con filtrado de Wiener y tamaño de ventana 3.	21
3.8	Sinograma con ruido gaussiano con filtrado de Wiener y tamaño de ventana 3.	22
3.9	Sinograma con ruido speckle con filtrado de Wiener y tamaño de ventana 3.	22
3.10	Sinograma con ruido gaussiano con filtrado bilateral y tamaño de ventana 5.	23
3.11	Sinograma con ruido speckle con filtrado bilateral y tamaño de ventana 3.	23
3.12	Imágenes de los fantasmas reconstruidos sin ruido, con ruido gaussiano y filtrado.	24
3.13	Imágenes de los fantasmas reconstruidos sin ruido, con ruido impulsivo y filtrado.	25
3.14	Comparación fantasmas sin ruido, con ruido gaussiano y filtrado	29
3.15	Comparación fantasmas sin ruido, con ruido impulsivo y filtrado	29

Índice de tablas

1.1	Radiación efectiva en mSv para la obtención de la imágenes.	2
1.2	Radiación absorbida en mGy en la realización de los escáneres.	2
2.1	Descripción de todos los objetos.	7
3.1	Resultados filtro gaussiano para ruido con varianza 0,00001.	19
3.2	Resultados filtro gaussiano para ruido con varianza 0,0005.	19
3.3	Resultados filtro gaussiano para ruido con varianza 0,005.	19
3.4	Resultados calidad del sinograma con ruido gaussiano y filtro gaussiano.	20

3.5	Resultados calidad del sinograma con ruido speckle y filtro gaussiano. . .	20
3.6	Resultados calidad del sinograma con ruido gaussiano y filtro de mediana. .	21
3.7	Resultados calidad del sinograma con ruido speckle y filtro de mediana. .	21
3.8	Resultados calidad del sinograma con ruido gaussiano y filtro de Wiener. .	22
3.9	Resultados calidad del sinograma con ruido speckle y filtro de Wiener. . .	22
3.10	Resultados calidad del sinograma con ruido gaussiano y filtro bilateral. . .	23
3.11	Resultados calidad del sinograma con ruido speckle y filtro bilateral. . . .	23
3.12	Valores de calidad de los fantasmas reconstruidos, mediante sinogramas con ruido gaussiano filtrado, y tiempo de reconstrucción.	24
3.13	Valores de calidad de los fantasmas reconstruidos, mediante sinogramas con ruido speckle filtrado, y tiempo de reconstrucción.	25
3.14	Resultados fantoma ruido gaussiano con filtro gaussiano.	27
3.15	Resultados fantoma ruido gaussiano con filtro de mediana.	27
3.16	Resultados fantoma ruido gaussiano con filtro de Wiener.	27
3.17	Resultados fantoma ruido gaussiano con filtro bilateral.	27
3.18	Resultados fantoma ruido speckle con filtro gaussiano.	28
3.19	Resultados fantoma ruido speckle con filtro de mediana.	28
3.20	Resultados fantoma ruido speckle con filtro de Wiener.	28
3.21	Resultados fantoma ruido speckle con filtro bilateral.	28

CAPÍTULO 1

Introducción

Vivimos en un mundo tecnológicamente en constante evolución en el que la informática ha alcanzado una considerable relevancia y, de hecho, resulta imprescindible para muchos de los aspectos que rodean nuestra vida cotidiana. La informática está presente en la práctica totalidad de los sectores en los que las personas desarrollan su actividad: la industria, la investigación científica y desarrollo tecnológico, las comunicaciones, los transportes, la salud y medicina, la educación, el comercio, las finanzas y negocios, el ocio y entretenimiento, la seguridad, la robótica, la inteligencia artificial, los macrodatos.

En el ámbito de las ciencias de la salud, los avances en la informática y la tecnología han permitido a la medicina mejorar un gran número de aspectos que en otros tiempos habrían sido impensables; como los casos de las resonancias magnéticas o las tomografías computarizadas. Por ello, se puede decir que la calidad de vida de las personas ha mejorado sustancialmente gracias a la tecnología y a la utilización de la informática como técnica de instrumentación científica avanzada para aplicaciones de imagen. Uno de los avances que podemos destacar al respecto, es el referente a la mejora y optimización de las imágenes de utilización médica para la obtención de mejores diagnósticos clínicos.

1.1 Motivación

Esa capacidad de mejora es la que motiva este trabajo. Desde el primer escáner realizado, de la zona cerebral, a un paciente en 1971 en el antiguo Atkinson Morley Hospital de Copse Hill (Londres), Inglaterra [1]; hasta la actualidad, han evolucionado diversas generaciones de escáneres. Diferenciando cuatro generaciones principalmente por un aumento exponencial del número de receptores, de un solo receptor a más de 4000; mejora del haz de rayos, de un haz plano a lo largo de la fuente a una haz en forma de abanico; movimiento de la fuente y de los receptores, de la translación de la fuente y rotación de los receptores cambió a rotación de la fuente con receptores fijos; reducción del tiempo de escaneo, de 30 minutos a unos pocos segundos [2].

El inconveniente que comporta el proceso del escáner TC¹ es la radiación ionizante², utilizada para la generación de las imágenes. Según la densidad de cada materia presente en el cuerpo humano, cada tipo de tejido absorbe mayor o menor cantidad de radiación. Así, aparecen en distintas tonalidades y puede diferenciarse tejido muscular, óseo, elementos del sistema cardiovascular, etc., en las imágenes.

El Sievert, símbolo Sv, es la unidad derivada del sistema internacional que mide la dosis de radiación efectiva, un indicador cuantitativo de la probabilidad de que pueda

¹Tomografía axial computarizada, también TAC.

²Radiación con suficiente energía para extraer los electrones de los átomos de la materia.

ocurrir un efecto estocástico, generalmente cáncer. La unidad derivada del sistema internacional que mide la cantidad de radiación ionizante absorbida por la materia es el Gray, con el símbolo Gy. A diferencia del Sievert, no tiene en cuenta el riesgo de los efectos estocásticos.

El problema reside en la cuantía de radiación necesaria para la obtención de imágenes de una resolución y nitidez aceptables. Tomando como referencia una radiografía, un escáner TC puede llegar a suponer un centenar de veces más de radiación como se puede observar en la tabla 1.1. En el caso particular de una imagen de tórax, la radiación efectiva para su obtención mediante TC, es de aproximadamente 400 veces mayor que la dosis necesaria para un mismo resultado con una radiografía [3].

	radiografía	TAC
Cabeza	0,07	2
Tórax	0,02	8
Abdomen	1	10-20
Pelvis	0,7	10-20

Tabla 1.1: Radiación efectiva en mSv para la obtención de la imágenes.

Según sobre qué zonas del cuerpo se produzca el escaneo, los órganos internos circundantes también absorben un significativo porcentaje de la radiación recibida, ver tabla 1.2; en un escáner de cabeza o en uno de tórax, los ojos y las mamas reciben una gran cantidad de radiación respectivamente [3].

	ojos	tiroides	mamas	útero	ovarios	testículos
Cabeza	50	1,9	0,03	-	-	-
Tórax	0,14	2,3	28	0,06	0,08	-
Abdomen	-	0,05	0,72	8	8	0,7
Pelvis	-	-	0,03	26	23	1,7

Tabla 1.2: Radiación absorbida en mGy en la realización de los escáneres. Los espacios en blanco indican una radiación menor que 0,005 mGy.

Por todo lo cual se hace patente la necesidad de reducir la dosis de radiación efectiva. Una disminución de la radiación empleada para la generación de las imágenes, significa una menor exposición del paciente, pero también una menor definición de la imagen, provocando del mismo modo la aparición de ruido. Por ello es necesario hacer uso de técnicas de filtrado que sean capaces de suprimir los artefactos presentes en su mayoría.

■ Motivación personal

En los últimos tres años, he sido usuario de los servicios de radiología en un hospital, con exploraciones mediante la prescripción de radiografías, resonancia magnética y tomografía axial computarizada, para tratar diferentes lesiones con un diagnóstico más preciso. Ello me ha llevado a interesarme por esta temática para desarrollar el trabajo.

1.2 Estado de la cuestión

Hasta la actualidad, mayoritariamente se utilizaban sistemas de reconstrucción de imágenes basadas en filtered back projection porque puede ejecutarse en equipos más antiguos, ya que presenta un coste computacional bajo y es posible la terminación del

método en un tiempo relativamente aceptable. A su vez la calidad de las imágenes resultantes son más que correctas cuando se utiliza un número elevado de proyecciones, situación en la que el paciente absorbe una alta dosis de radiación. Sin embargo, la calidad de las imágenes empeora cuando se reduce el número de proyecciones.

En estudios sobre ruido gaussiano y/o impulsivo (speckle) presente en imágenes de TC, RM³ o US⁴, las técnicas de filtrado y algoritmos propuestos por ejemplo en [4] y [5], se han centrado en el filtrado de las imágenes una vez procesadas y reconstruidas. En este mismo sentido, en este trabajo se pretende analizar el comportamiento de estas técnicas cuando se aplican sobre el sinograma antes de la reconstrucción de la imagen.

1.3 Objetivos

El fin del presente trabajo es el análisis del comportamiento de unos determinados filtros, sobre un sinograma⁵ de pocas proyecciones y adicionalmente sobre una imagen de fantoma matemático⁶. Así como la posterior evaluación, de la etapa de reconstrucción, con la utilización de los sinogramas procedentes del filtrado.

Para la consecución del objetivo del trabajo, se procede al cumplimiento de unos objetivos más específicos.

- Utilizar el entorno MATLAB para el filtrado de imágenes.
- Diseñar las funciones para las pruebas de filtrado y evaluación de sinogramas y fantomas.
- Someter a prueba los diferentes parámetros de cada filtro.
- Analizar los datos extraídos para la selección de los parámetros más favorables.
- Evaluar los resultados del proceso de reconstrucción de los datos filtrados con los parámetros elegidos.
- Comparar la calidad de la imagen de los fantomas filtrados.
- Estudiar el efecto del ruido en el sinograma y en el fantoma.

1.4 Estructura de la memoria

El presente documento está distribuido en cuatro capítulos, posteriormente se encuentra la bibliografía con todas las referencias empleadas, seguida de un anexo en el cual se localizan todos los resultados logrados y las imágenes obtenidas.

En el actual capítulo se realiza una introducción de la problemática a resolver y un sumario de la evolución en la obtención de imágenes médicas en TC, desde el primer prototipo hasta la actualidad. Se indica la situación actual de los procedimientos de reconstrucción de las imágenes y los procesos de filtrado aplicados a las imágenes médicas. También se expone la motivación para la realización del presente trabajo, al igual que los objetivos a satisfacer.

³Resonancia magnética.

⁴Ultrasonidos.

⁵Representación gráfica de los datos obtenidos en un escáner de TAC antes de cualquier procesado.

⁶Reproducción matemática la cual representa estructuras anatómicas.

En el capítulo 2 se describe el procedimiento de reconstrucción de las imágenes elegido para la elaboración de las pruebas, así como una descripción de las técnicas de filtrado elegidas y de los parámetros de calidad empleados para examinar el comportamiento de los filtros.

En el capítulo 3 se presentan las funciones utilizadas para el análisis de los métodos de filtrado seleccionados, contiene el resultado y análisis de las diferentes técnicas de filtrado y finalmente se encuentra una descripción de los resultados del proceso de reconstrucción utilizando los sinogramas filtrados.

Por último, en el capítulo 4 se presentan las conclusiones del trabajo desarrollado y posibles acciones a desarrollar en el futuro.

CAPÍTULO 2

Materiales y métodos

Así pues, con la rápida mejora de la tecnología y el aumento de la potencia de cómputo de los sistemas, se empieza a hacer uso de métodos algebraicos iterativos para la reconstrucción de la imagen. Estas técnicas tienen un mayor coste computacional pero no necesitan de una colección completa de proyecciones para lograr reconstruir una imagen con una calidad aceptable.

2.1 Reconstrucción de imagen TC

Cuando se adquieren los sinogramas y además se emplean pocas vistas (proyecciones) en la fase de reconstrucción de la imagen aparecen más o menos artefactos del tipo “escalonado” [6] y ruido del tipo gaussiano y/o impulsivo. En la figura 2.1 se muestra el proceso completo de reconstrucción de una imagen de TC en dos dimensiones.

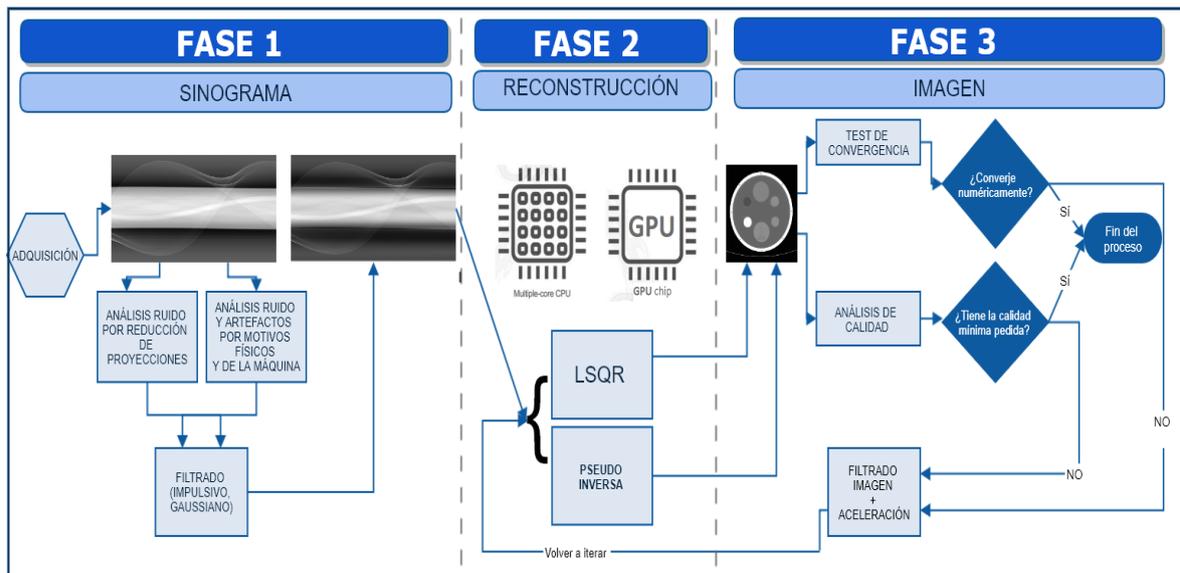


Figura 2.1: Proceso completo de reconstrucción de una imagen de TC (2D)

La fase 1 consiste en la generación y filtrado del sinograma, y es la fase en la que se centra principalmente el TFG, y que se abordará más detalladamente en la sección 2.3.

En la fase 2, denominada fase de reconstrucción algebraica de imagen TC consiste en la resolución del sistema de ecuaciones dado por la ecuación 2.1. Esa fase ha sido resuelta en diversos trabajos [7, 8, 9].

$$Af = g + w \quad (2.1)$$

$$A = (a_{ij}) \in R^{M \times N} \quad (2.2)$$

$$g = [g_1, g_2, \dots, g_M]^T \in R^M \quad (2.3)$$

$$f = [f_1, f_2, \dots, f_N]^T \in R^N \quad (2.4)$$

$$w = [w_1, w_2, \dots, w_M]^T \in R^M \quad (2.5)$$

La matriz del sistema A es de tamaño $M \times N$, donde M es el número de rayos trazados, y N el tamaño en píxeles de la imagen a reconstruir. La matriz A se genera discretizando el espacio de exploración en píxeles y midiendo la influencia de cada haz que se traza en cada píxel, tal como se explica en el trabajo de Joseph[10], donde a_{ij} representa la contribución del píxel j al rayo i . El vector de proyecciones g , es el cálculo de la atenuación experimentada por los rayos que pasan por el objeto de estudio, ver figura 2.2. El tamaño del vector para cada ángulo de proyección es $n \times 1$, siendo n el número de detectores que forman el escáner. Mediante la concatenación de todos ellos se obtiene un vector unidimensional que representa el objeto entero. El vector f es la imagen que se reconstruye como resultado de aproximar este sistema de ecuaciones. La imagen viene representada por un vector de tamaño N que se corresponde al número de píxeles. Por lo tanto, para una imagen de resolución 128×128 píxeles, el vector solución f será de tamaño $128^2 \times 1$. Por último w , es el vector ruido desconocido o perturbación, que en el trabajo es simulado debido a que se está trabajando con proyecciones de un fantoma matemático, en la sección 3.1 se explica el proceso empleado en su generación.

En la fase 3 se analiza la calidad de la imagen obtenida en la fase 2 y se decide como proceder, ver figura 2.1. Tanto en la fase 1 de generación del sinograma como en la fase 3 de mejora de la imagen se deben utilizar filtros para disminuir el ruido, que es la tarea principal en la que se basa el presente trabajo.

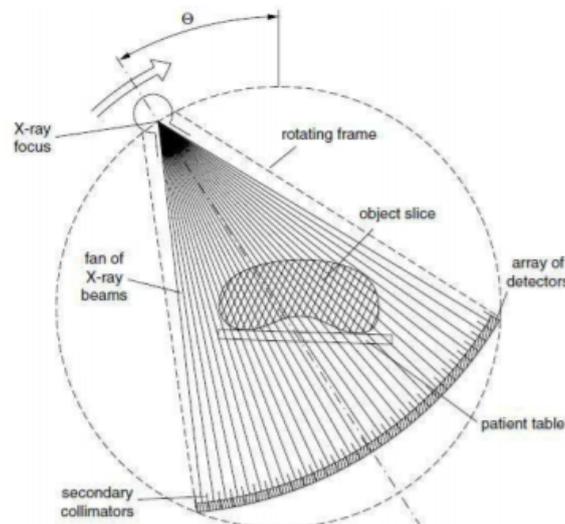


Figura 2.2: Adquisición de datos durante el proceso de escaneo. Fuente: [6]

2.1.1. Datos de entrada

Para realizar los estudios propuestos, es necesario generar la matriz del sistema A , y el vector de proyecciones g . Ambos conjuntos de datos tendrán que ser generados una vez para cada resolución de imagen deseada, lo cual se ha realizado mediante el método Forward Projection de Joseph [10], el cual calcula el peso de cada rayo sobre cada píxel (formando la matriz A) ponderado por los correspondientes coeficientes de atenuación lineal de los rayos. Para esta simulación se ha hecho uso del fantoma Forbild Head Phantom [11] que representa matemáticamente una aproximación de las estructuras existentes en una cabeza humana, ver la figura 2.3 y tabla 2.1.

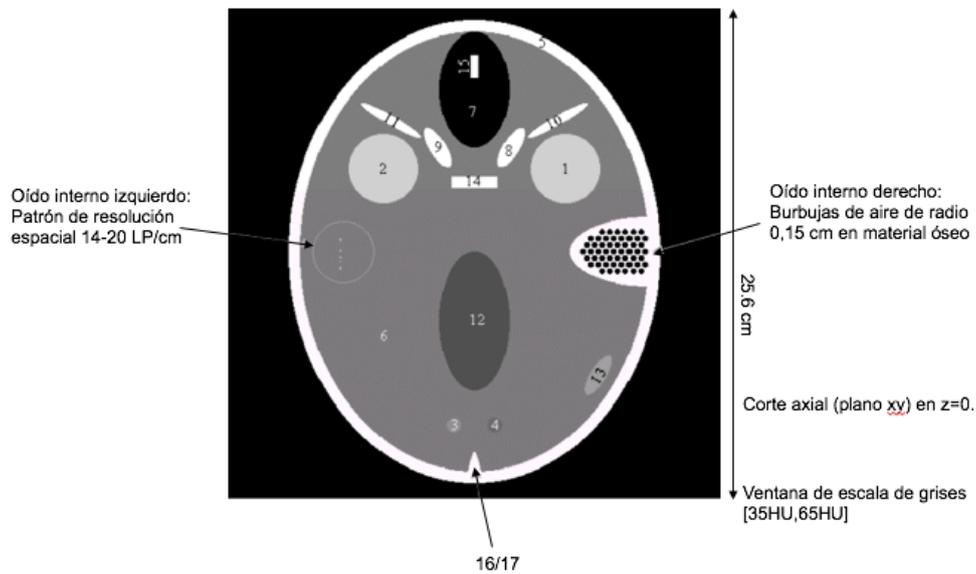


Figura 2.3: Corte axial en el punto central. Fuente: [11]

Etiqueta	Geometría	Número-CT	Relación anatómica
1	Esfera	60	Ojo
2	Esfera	60	Ojo
3	Esfera	52.5	
4	Esfera	47.5	
5	Elipse	800	Cúpula
6	Elipse	50	Materia cerebral homogénea
7	Elipse	-1000	Senos frontales
8	Elipse	800	Hueso alrededor del seno frontal
9	Elipse	800	Hueso alrededor del seno frontal
10	Cilindro elíptico	800	Hueso alrededor del seno frontal
11	Cilindro elíptico	800	Hueso alrededor del seno frontal
12	Elipsoide	45	Ventrículo
13	Elipsoide	55	Hematoma subdural
14	Cilindro elíptico	800	Hueso alrededor del seno frontal
15	Cilindro elíptico	800	Hueso alrededor del seno frontal
16	Cono	800	Protuberancia occipital interna
17	Cono	800	Protuberancia occipital interna

Tabla 2.1: Descripción de todos los objetos.

El sinograma g de tamaño ‘vistas’ se genera a partir de las proyecciones que se seleccionen del conjunto de datos completo generado por el esquema descrito por la ecuación 2.6, dependiendo del número de vistas utilizadas habrá más o menos ruido.

$$\Theta_i = \begin{cases} (360/vistas) * (i - 1) & \text{si } 1 \leq i \leq (vistas/2) \\ 1,5 + (360/vistas) * (i - 1) & \text{si } (vistas/2) < i < vistas \\ (360 - 1) & \text{si } i = vistas \end{cases} \quad (2.6)$$

En la pruebas el objeto es fijado, la fuente de rayos X y el detector giran alrededor del eje de rotación de forma sincronizada. Se toma una distancia de 75 cm de la fuente de rayos X al isocentro. El tamaño de píxel de la imagen reconstruida es de $0,0512 \times 0,0512 \text{ cm}^2$. El vector de detectores virtuales es equidistante y contiene 1025 elementos, cada uno de los cuales tiene una abertura de 0,0784 cm. En la figura 2.4 se dispone de más información.

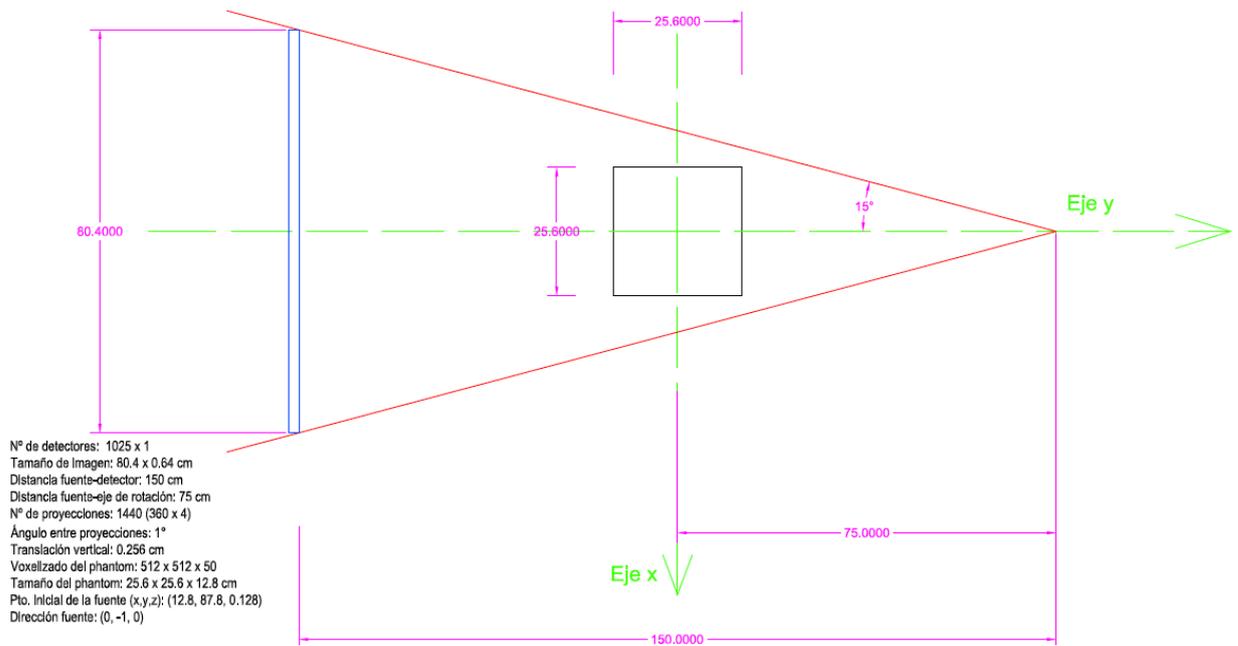


Figura 2.4: Características del escáner virtual utilizado en las pruebas.

2.1.2. Método iterativo de reconstrucción

El método de reconstrucción (fase 2 y 3) que se utiliza consta de tres procesos que se repiten hasta lograr la convergencia: un primer proceso de resolución del sistema de ecuaciones, tras la cual se obtiene una imagen solución aproximada. Posteriormente, un proceso de filtrado de dicha imagen, seguido por un proceso de aceleración que prepara el método para volver a iterar si no se ha logrado la convergencia.

En la resolución del sistema de ecuaciones se emplea el método QR de mínimos cuadrados (LSQR) [12] ya que es uno de los métodos más estables. El método resuelve el sistema 2.1 para $w = 0$ minimizando $\min \|Af - g\|_2$.

Para el filtrado del ruido de tipo gaussiano y/o impulsivo de la imagen obtenida en la resolución del sistema se aplica el filtro no lineal *Soft Thresholding Filter* [13] [14], conservando el gradiente no solo vertical-horizontal sino también el gradiente diagonal,

provocando que la imagen sea más nítida sin perder su estructura en los bordes, lo cual es de vital importancia en pruebas médicas de imagen de TC. Por último, a la imagen filtrada se aplica la técnica de aceleración *Fast Iterative Shrinkage Thresholding Algorithm* (FISTA) definido en [15].

2.2 Parámetros de calidad

Con el objetivo de analizar los resultados obtenidos mediante la utilización de los filtros, es necesario medir la calidad de las imágenes, comparando cada una, y así seleccionar los parámetros idóneos para obtener el máximo rendimiento posible de las técnicas de filtrado.

2.2.1. SSIM – índice de similitud estructural

Se trata de un método que se utiliza para medir la similitud entre dos imágenes, una imagen de referencia y otra elegida, considerando la degradación como el cambio de la imagen, a nivel estructural, entre ambas [16]. Los píxeles mantienen una relación de interdependencia con sus adyacentes. Con lo cual, estas dependencias aportan información sobre la estructura de los objetos de forma visual, cómo se percibe por el cerebro humano.

$$SSIM = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{x,y} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

2.2.2. PSNR – proporción máxima de señal a ruido

Esta medida representa el ratio de valor máximo que puede alcanzar un píxel y la cantidad de ruido que afecta a la imagen [16]. Se evalúa el valor de cada píxel y el nivel de ruido presente.

$$PSNR = 10 * \log_{10} \frac{MAX_{I_0}^2}{MSE}$$

Donde el MSE es la media de los errores de la imagen al cuadrado.

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} ||I_0(i, j) - u(i, j)||^2$$

2.3 Técnicas de filtrado

En vista de la variedad de técnicas encontradas en los estudios consultados, es necesario acotar el número de métodos para filtrar las imágenes. Tomando como referencia el trabajo [5], se ha procedido a la elección de las siguientes técnicas de filtrado.

2.3.1. Filtro gaussiano

Es el resultado de suavizado de la imagen mediante una función gaussiana [17]. A costa de la reducción o eliminación del ruido, se corre el riesgo de perder una gran cantidad de detalle por el hecho de no preservar los bordes. Obteniéndose una imagen borrosa y poco nítida.

El filtro gaussiano se aplica sobre una imagen 2D según se define en (2.7), donde G es la máscara gaussiana con coordenadas x e y , σ es el parámetro que define la desviación típica. Si el valor de σ es grande, el efecto de suavizado de imagen será mayor. El suavizado puede efectuarse mediante la convolución de una ventana de la imagen original $I(x,y)$ de tamaño $w \times h$ con una máscara gaussiana G como se ilustra en la ecuación (2.8). La imagen filtrada se obtiene al calcular la suma de productos entre todos los píxeles de la ventana de la imagen de entrada y la matriz gaussiana.

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \quad (2.7)$$

$$f(x,y) = \sum_{i=0}^{w-1} \sum_{j=0}^{h-1} G(i,j)I(x-i,y-j) \quad (2.8)$$

2.3.2. Filtro de mediana

El filtro mediana es un método no lineal que se utiliza para eliminar el ruido de las imágenes [18]. Es ampliamente utilizado ya que es muy efectivo para eliminar el ruido y preservar los bordes. Es particularmente efectivo para eliminar el ruido tipo 'sal y pimienta'.

El filtro mediana funciona moviéndose a través de la imagen píxel por píxel, reemplazando cada valor con el valor mediana de los píxeles vecinos. El patrón de vecinos se denomina "ventana", que se desliza, píxel por píxel sobre toda la imagen.

En los límites de la imagen no existen valores anteriores o posteriores, se utiliza el valor del propio píxel repetido en las zonas que no existan valores. También es posible rellenar los espacios vacíos con ceros o unos.

En la Figura 2.5 se muestra un ejemplo del cálculo del valor del filtro correspondiente al píxel central de una ventana de tamaño 3×3 .

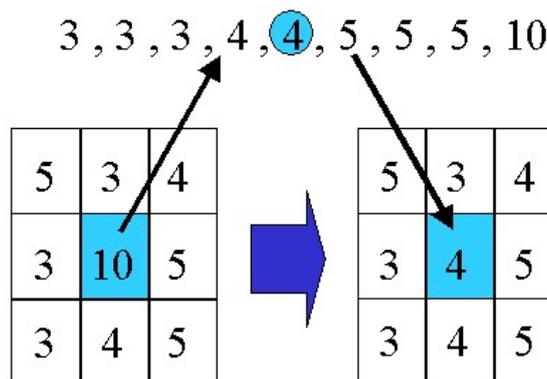


Figura 2.5: Ejemplo filtro mediana ventana 3×3 , píxel central.

2.3.3. Filtro de Wiener

Recibe el nombre por el matemático estadounidense Norbert Wiener, quien propuso el filtro. Mediante filtrado inverso elimina el ruido aditivo e invierte la difuminación de la imagen simultáneamente [19].

Considera las imágenes y el ruido como variables aleatorias. El objetivo es encontrar una estimación de la imagen original tal que el error medio cuadrático entre ellas sea minimizado.

El filtro estima la media local (2.9) y la varianza (2.10) alrededor de cada píxel, donde η es la vecindad local N por M de cada píxel en la imagen A . Luego hace un filtrado (2.11) píxel a píxel usando estas estimaciones, donde v^2 es la varianza del ruido. Si no se proporciona la varianza del ruido, se usa el promedio de todas las varianzas locales estimadas.

$$\mu = \frac{1}{NM} \sum_{n_1, n_2 \in \eta} a(n_1, n_2) \quad (2.9)$$

$$\sigma^2 = \frac{1}{NM} \sum_{n_1, n_2 \in \eta} a^2(n_1, n_2) - \mu^2 \quad (2.10)$$

$$b(n_1, n_2) = \mu + \frac{\sigma^2 - v^2}{\sigma^2} (a(n_1, n_2) - \mu) \quad (2.11)$$

2.3.4. Filtro bilateral

Se trata de un filtro no lineal y preservador de bordes [20]. Reemplaza el valor de cada píxel por una media ponderada de los píxeles próximos a su localización. La ponderación generalmente está basada en una distribución normal según los valores de los píxeles.

El filtro bilateral se define como

$$I^{\text{filtrada}}(x) = \frac{1}{W_p} \sum_{x_i \in \Omega} I(x_i) f_r(\|I(x_i) - I(x)\|) g_s(\|x_i - x\|), \quad (2.12)$$

donde el término normalización

$$W_p = \sum_{x_i \in \Omega} f_r(\|I(x_i) - I(x)\|) g_s(\|x_i - x\|) \quad (2.13)$$

asegura que el filtro conserva la energía de la imagen y:

I^{filtrada} es la imagen filtrada; I es la imagen de entrada original para ser filtrada; x son las coordenadas del píxel a filtrar; Ω es la ventana centrada en x ; f_r es el rango del kernel para suavizar las diferencias en intensidades; g_s es el rango espacial para suavizar las diferencias de coordenadas. Las dos últimas funciones pueden ser funciones gaussianas.

El peso W_p se asigna mediante la cercanía espacial y la diferencia de intensidad. Consideremos un píxel situado en (i, j) que necesita ser filtrado en imagen con sus píxeles vecinos y uno de sus píxeles vecinos se encuentra en (k, l) . El peso asignado por píxel (k, l) para eliminar el ruido del *pxel* (i, j) viene dado por:

$$w(i, j, k, l) = e^{\left(-\frac{(i-k)^2 + (j-l)^2}{2\sigma_d^2} - \frac{\|I(i, j) - I(k, l)\|^2}{2\sigma_r^2} \right)} \quad (2.14)$$

donde σ_d y σ_r son parámetros de suavizado y $I(i, j)$ y $I(k, l)$ son la intensidad de los píxeles (i, j) y (k, l) respectivamente. Después de calcular los pesos, hay que normalizarlos:

$$I_D(i, j) = \frac{\sum_{k, l} I(k, l) * w(i, j, k, l)}{\sum_{k, l} w(i, j, k, l)} \quad (2.15)$$

donde I_D es la intensidad sin ruido del píxel (i, j) .

El método de reconstrucción LSQR correspondiente a las fases 2 y 3 de la figura 2.1 está desarrollado en MATLAB¹. Dicho software contiene un *toolbox*² de procesamiento de imágenes, con una sección dedicada al filtrado y a la mejora de imágenes. El *toolbox* posee un conjunto de algoritmos estándar de referencia, predefinidos y en mayor o menor medida personalizables. Por todo lo anterior, se toma la decisión de utilizar el propio módulo de filtrado de imágenes, ya que contiene los filtros anteriormente descritos, para estudiar su viabilidad en el presente trabajo.

¹Herramienta matemática con entorno de desarrollo y lenguaje propio.
Disponible en www.mathworks.com/products/matlab.html

²Módulo de contenido con funciones de una misma temática.

Estudio de los métodos de filtrado

Para hacer uso de las técnicas propuestas en la sección 2.3, hemos decidido utilizar las funciones implementadas en el entorno MATLAB. Quedando relacionadas de la siguiente manera:

- Filtro gaussiano <-> `imgaussfilt`
- Filtro de mediana <-> `medfilt2`
- Filtro de Wiener <-> `wiener2`
- Filtro bilateral <-> `imbilatfilt`

En este caso se utilizarán las funciones `imgaussfilt()`, `medfilt2()`, `wiener2()` y `imbilatfilt()`. Las funciones con el número 2 en el nombre indican que se trata de las versiones de dos dimensiones.

Por otra parte, ya que el sinograma utilizado proviene de un fantoma matemático y no contiene ruido, se empleará la función `imnoise()` para la adición de ruido.

La función del filtro gaussiano tiene la forma `imgaussfilt(im, sigma, nombre, valor)`, siendo los argumentos de entrada: `im` la imagen a filtrar y `sigma` la desviación estándar de la función. En la última parte los parámetros opcionales, usando los pares nombre-valor, en este caso el dominio en el que se realizara la operación de convolución.

El método `medfilt2(im, [m n])`, efectúa un filtrado de mediana a la imagen `im` donde el valor de cada píxel se obtiene por el valor de la mediana de los píxeles de alrededor en un área delimitada por el tamaño que se indique en el vector `[m n]`.

El método `wiener2(im, [m n])` somete a la imagen `im` al filtro de Wiener usando los píxeles vecinos, el vector `[m n]`, para estimar la desviación estándar y la media de la imagen.

El método `imbilatfilt(im, nombre, valor)` aplica sobre la imagen `im` un filtro bilateral preservador de bordes, además haciendo uso de los pares nombre-valor para especificar el área de acción de suavizado entorno al píxel.

3.1 Filtrado del sinograma

El fantoma matemático `Forbild` elegido, descrito en la sección 2.1.1, produce una imagen sin ningún tipo de degradación, se trata de una simulación sin ruido. Para reproducir un sinograma más cercano a la realidad que contenga ruido, se utiliza la función

`imnoise()` del mismo módulo de procesamiento de imágenes. Esta función permite introducir ruido de tipo gaussiano o de tipo speckle en las imágenes. Por ejemplo, `imnoise(im, 'gaussian', m, var)` añade a la imagen `im` ruido blanco gaussiano con media `m` y varianza `var`. En cuanto al ruido speckle, `imnoise(im, 'speckle', var)` añade a la imagen `im` ruido multiplicativo con varianza `var`.

El ruido es añadido durante la generación del vector de proyecciones, aunque son necesarios unos pasos previos ya que la imagen está en una escala de color diferente a la escala en la que trabajan las funciones de MATLAB (256 niveles de gris). La imagen del fantoma digital es normalizada antes de ser utilizada para la generación del sinograma como se muestra en el siguiente código.

```
1 im = generate_discrete_phantom(xs, dx, nx, ys, dy, ny, phantom);
2 im = 255*mat2gray(im);
3 save(strcat('./IMref',int2str(dim),'head.mat'),'im','-v7.3');
```

Aunque la función `imnoise()` acepte como parámetro de entrada datos de tipo doble precisión, en imágenes con valores de píxeles con decimales, devuelve la imagen con un rango de valores entre 0 y 1. Por ello, una vez generado, el sinograma también es normalizado. Además se realiza una copia para la inserción de ruido, en la cual se cambia el tipo de doble a entero sin signo de 8 bits como se muestra en el código siguiente.

```
1 b = 255*mat2gray(b);
2 bn = uint8(b);
3 bn = imnoise(bn,'gaussian',0,0.0005); % tambien bn=imnoise(bn,'speckle',0.0005);
;
```

Finalmente, utilizando la función `reshape()`, se almacenan los sinogramas en forma de vector de proyecciones (unidimensional).

```
1 bn = reshape(bn, ngamma*nlambda,1);
2
3 b = reshape(b, ngamma*nlambda,1);
```

El hecho de normalizar tanto la imagen como el sinograma se debe a que el rango de valores se sitúa entre 0 y 1,8, con valores tan pequeños la más mínima adición de ruido elimina cualquier información relevante.

Este proceso se realizará varias veces con el fin de evaluar los dos tipos de ruido, gaussiano y speckle, y los diferentes valores de los parámetros de cada uno. Específicamente se va modificando la varianza de ambos con el fin de añadir más o menos ruido y encontrar una cantidad de ruido lo suficientemente significativa sin eliminar gran parte de información.

Para la realización de las pruebas del nivel de ruido, se ha escrito un fragmento de código que permite ejecutar las diferentes funciones de filtrado de una sola vez como se muestra en el bloque de código siguiente.

En primer lugar se redimensionan ambos vectores de proyecciones, sin ruido y con ruido, en función de las vistas, líneas 2 y 3. Mediante `num2str()` el valor de `i` se convierte en un carácter y se concatena a la cadena `'filtro_b'`. La cadena es entonces asignada como valor del *handler*¹ `filtro_b` con la función `str2func()`, línea 10. Así, en cada iteración del bucle `for` se hace una llamada a un método de filtrado distinto. El resultado que devuelve cada función se almacena en una matriz tridimensional, `Aso1B`, donde en cada página se encuentran los resultados de la resolución de cada función, línea 12.

¹Un *function handler* o identificador de función, es un tipo de datos que almacena una asociación a una función.

```

1 %Sinogramas segun las vistas elegidas , en este caso 60
2 b = reshape(b, 1025, vistas);
3 bn = reshape(bn, 1025, vistas);
4
5 %Ciclo for 1 a 4, numero de filtros
6 for i=1:4
7
8     %Asignacion de valor al handler
9     %i=1 – filtro_b=filtro_b1 , i=2 – filtro_b=filtro_b2 ...
10    filtro_b = str2func(['filtro_b' num2str(i)]);
11    % Matriz solucion
12    AsolB(:, :, i) = filtro_b(b,bn);
13
14 end

```

Se ha decidido crear unas funciones, una para cada filtro descrito en la sección 2.3, que apliquen cada técnica de filtrado al sinograma y calculen los parámetros de calidad de la imagen.

3.1.1. Función filtro gaussiano

La función implementada para aplicar el filtro gaussiano tiene como parámetros de entrada el sinograma sin ruido b y el sinograma con ruido $bNoise$. Se realiza un bucle for para la prueba de distintos valores de sigma y conocer su comportamiento con los diferentes niveles de ruido. Otro parámetro configurado es el dominio espacial en el que se realiza el filtrado, a diferencia del dominio de frecuencia aplicado al filtrado de señales, línea 15. También se mide el tiempo antes y después de la invocación a `imgaussfilt()` para saber cuánto tarda la ejecución del filtro, líneas 13 y 16 de la función `filtro_b1`. Antes de proseguir se realiza un cambio de tipo de entero a doble para la medición de la calidad de la imagen, líneas 19 y 20. Después, se calculan los parámetros SSIM y PSNR, entre el sinograma filtrado y el original, y el sinograma filtrado con el de ruido añadido, líneas 23-26. Los resultados entonces son almacenados en la matriz solución `Asol` junto con el tiempo necesario para ejecutar el filtro. Por último, la función devuelve la matriz solución y el sinograma filtrado.

■ Código filtro_b1

```

1 function [Asol, bFilt] = filtro_b1(b,bNoise)
2 %FILTRO_B1 Filtro gaussiano
3 % Filtrado de bNoise y calculo de parametros de calidad SSIM y PSNR
4 % entre b – bNoise y b – bFilt
5
6 km = 1;
7 kn = 1;
8
9 %Ciclo para la evaluacion de parametros sigma
10 for i=0.3:0.2:0.9
11
12     %Calculo del tiempo de cada ejecucion con tic + toc
13     tic;
14     %Ejecucion de la funcion de filtrado
15     bFilt = imgaussfilt(bNoise,i,'FilterDomain','spatial');
16     Asol(km,kn+4) = toc;
17
18     %Cambio de tipo para realizar las mediciones
19     bN = double(bNoise);
20     bFilt = double(bFilt);
21

```

```

22     % Calculo de parametros SSIM y PSNR y almacenamiento en Asol
23     Asol(km, kn) = ssim(b, bN);
24     Asol(km, kn+1) = PSNRImages(b, bN);
25     Asol(km, kn+2) = ssim(b, bFilt);
26     Asol(km, kn+3) = PSNRImages(b, bFilt);
27
28     km = km+1;
29
30     end
31
32 end

```

3.1.2. Función filtro de mediana

En esta función implementada para aplicar el filtro mediana se usa el mismo esquema que en el caso anterior, pero en lugar de los valores de sigma, el índice del ciclo se corresponde con el tamaño del *kernel* del filtro, es decir la ventana de píxeles a los que se aplica el proceso de filtrado, línea 15.

■ Código filtro_b2

```

1 function [Asol, bFilt] = filtro_b2(b, bNoise)
2 % FILTRO_B2 Filtro de mediana
3 % Filtrado de bNoise y medicion de parametros de calidad SSIM y PSNR
4 % entre b = bNoise y b = bFilt
5
6 km = 1;
7 kn = 1;
8
9 % Ciclo para la evaluacion de parametros de ventana
10 for i=3:2:9
11
12     % Calculo del tiempo de cada ejecucion con tic + toc
13     tic;
14     % Ejecucion de la funcion de filtrado
15     bFilt = medfilt2(bNoise, [i i]);
16     Asol(km, kn+4) = toc;
17
18     % Cambio de tipo para realizar las mediciones
19     bN = double(bNoise);
20     bFilt = double(bFilt);
21
22     % Calculo de parametros SSIM y PSNR y almacenamiento en Asol
23     Asol(km, kn) = ssim(b, bN);
24     Asol(km, kn+1) = PSNRImages(b, bN);
25     Asol(km, kn+2) = ssim(b, bFilt);
26     Asol(km, kn+3) = PSNRImages(b, bFilt);
27
28     km = km+1;
29
30     end
31
32 end

```

3.1.3. Función filtro de Wiener

La aplicación de este filtro se realiza mediante una función que es similar a la anterior, se sustituye el nombre del filtro y se mantiene la nomenclatura que define la ventana de acción del mismo, usando la forma de matriz cuadrada.

■ Código filtro_b3

```

1 function [Asol, bFilt] = filtro_b3(b, bNoise)
2 % FILTRO_B3 Filtro de Wiener
3 % Filtrado de bNoise y medicion de parametros de calidad SSIM y PSNR
4 % entre b - bNoise y b - bFilt
5
6 km = 1;
7 kn = 1;
8
9 % Ciclo para la evaluacion de parametros de ventana
10 for i=3:2:9
11
12     % Calculo del tiempo de cada ejecucion con tic + toc
13     tic;
14     % Ejecucion de la funcion de filtrado
15     bFilt = wiener2(bNoise,[i i]);
16     Asol(km, kn+4) = toc;
17
18     % Cambio de tipo para realizar las mediciones
19     bN = double(bNoise);
20     bFilt = double(bFilt);
21
22     % Calculo de parametros SSIM y PSNR y almacenamiento en Asol
23     Asol(km, kn) = ssim(b, bN);
24     Asol(km, kn+1) = PSNRImages(b, bN);
25     Asol(km, kn+2) = ssim(b, bFilt);
26     Asol(km, kn+3) = PSNRImages(b, bFilt);
27
28     km = km+1;
29
30 end
31
32 end

```

3.1.4. Función filtro bilateral

De igual modo que en las funciones previas, se modifica la notación donde queda definida la ventana del filtrado, utilizando la formula nombre-valor. Siendo el nombre del parámetro 'NeighborhoodSize' e 'i' el valor que obtiene el mismo.

■ Código filtro_b4

```

1 function [Asol, bFilt] = filtro_b4(b, bNoise)
2 % FILTRO_B4 Filtro bilateral
3 % Filtrado de bNoise y medicion de parametros de calidad SSIM y PSNR
4 % entre b - bNoise y b - bFilt
5
6 km = 1;
7 kn = 1;
8
9 % Ciclo para la evaluacion de parametros de ventana
10 for i=3:2:9
11
12     % Calculo del tiempo de cada ejecucion con tic + toc
13     tic;
14     % Ejecucion de la funcion de filtrado
15     bFilt = imbilatfilt(bNoise, 'NeighborhoodSize', i);
16     Asol(km, kn+4) = toc;
17

```

```

18      %Cambio de tipo para realizar las mediciones
19      bN = double(bNoise);
20      bFilt = double(bFilt);
21
22      %Calculo de parametros SSIM y PSNR y almacenamiento en Asol
23      Asol(km, kn) = ssim(b, bN);
24      Asol(km, kn+1) = PSNRImages(b, bN);
25      Asol(km, kn+2) = ssim(b, bFilt);
26      Asol(km, kn+3) = PSNRImages(b, bFilt);
27
28      km = km+1;
29
30      end
31
32      end

```

■ Parámetros de filtrado

Los tamaños del *kernel* o las ventanas de los filtros han de ser valores enteros impares mayores que uno. Existe algún parámetro más en todos los filtros como el *padding*, la forma en la que se rellena el espacio vacío fuera de los límites de la imagen, para la aplicación de las técnicas de filtrado sobre los bordes. Los valores disponibles son: simétrico, usando un efecto espejo sobre el borde; replicación del contenido del borde, se rellena con el mismo valor que el borde; o un valor numérico escalar, como 0, 1, etc. Una vez examinado cada parámetro de cada filtro con todos los valores posibles, se han optado por utilizar los valores por defecto a posteriori, ya que la diferencia de resultados era despreciable.

Las distintas funciones de los filtros devuelven el resultado de las mediciones de calidad entre los sinogramas original, filtrado y con ruido. Estos resultados son almacenados en la matriz de tres dimensiones *AsolB* en cada iteración, una vez terminada la ejecución de la función correspondiente.

3.2 Análisis de los resultados

Antes de analizar el comportamiento de los filtros debemos determinar el valor del parámetro de la varianza que se debe utilizar en la función *imnoise()*. Para ello, se van a realizar pruebas añadiendo ruido con diferentes varianzas y comprobando los resultados de calidad tras filtrar.

A continuación, de manera comparativa, en las tablas 3.1, 3.3 y 3.3 se muestran los resultados de los valores elegidos junto con los que presentan un nivel inferior de ruido y un nivel superior. Se puede observar los parámetros evaluados, los resultados de las mediciones de calidad y el tiempo en segundos dedicado únicamente al proceso de filtrado.

Una vez se han realizado todas las ejecuciones necesarias, se ha llegado a la conclusión de que en el factor varianza del ruido gaussiano y speckle el valor más adecuado es 0,0005. En ambos tipos son los valores más convenientes porque aportan suficiente ruido como para apreciar el funcionamiento de los métodos de filtrado. Si se añade poco ruido los filtros empeoran la calidad de la imagen, no trabajan con valores adecuados. Tampoco es posible añadir ruido en exceso, a pesar de que ahora sí funcionarían los filtros, no quedaría información primordial en la imagen. En las figuras 3.1, 3.2 y 3.3 se muestran los sinogramas resultado de añadir ruido con diferentes varianzas.

Todos los valores examinados y los resultados de todas las ejecuciones se encuentran en el anexo A.1. Los valores seleccionados de los parámetros de los filtros son utilizados posteriormente con la finalidad de observar el comportamiento de los sinogramas filtrados en la reconstrucción de la imagen.



sigma	SSIM	PSNR	tiempo
Con ruido	0,5701	50,1945	-
0,3	0,5701	50,1945	0,0012
0,5	0,5657	45,5727	0,0011
0,7	0,5463	40,2661	0,0012
0,9	0,5362	37,8728	0,0011

Figura 3.1: Sinograma con ruido gaussiano con varianza 0,00001 **Tabla 3.1:** Resultados filtro gaussiano para ruido con varianza 0,00001.



sigma	SSIM	PSNR	tiempo
Con ruido	0,3844	34,0654	-
0,3	0,3844	34,0655	0,0014
0,5	0,4409	36,7953	0,0011
0,7	0,4693	37,1464	0,0017
0,9	0,4668	36,1990	0,0011

Figura 3.2: Sinograma con ruido gaussiano con varianza 0,0005 **Tabla 3.2:** Resultados filtro gaussiano para ruido con varianza 0,0005.



sigma	SSIM	PSNR	tiempo
Con ruido	0,1791	24,2016	-
0,3	0,1805	24,2762	0,0011
0,5	0,2458	27,4077	0,0011
0,7	0,3111	29,7862	0,0011
0,9	0,3390	30,5065	0,0011

Figura 3.3: Sinograma con ruido gaussiano con varianza 0,005 **Tabla 3.3:** Resultados filtro gaussiano para ruido con varianza 0,005.

3.2.1. Resultados de la función filtro_b1()

A. Filtro gaussiano

En primer lugar, en el tipo de ruido gaussiano, figura 3.4, en la tabla 3.4 se aprecia una mejora en los valores de sigma 0,5 , 0,7 y 0,9. Destacando el valor 0,7 entre los tres anteriores con SSIM = 0,4693 y PSNR = 37,1464; además el tiempo de ejecución es similar.



sigma	SSIM	PSNR	tiempo
Con ruido	0,3844	34,0654	-
0,3	0,3844	34,0655	0,0014
0,5	0,4409	36,7953	0,0011
0,7	0,4693	37,1464	0,0017
0,9	0,4668	36,1990	0,0011

Figura 3.4: Sinograma con ruido gaussiano con filtrado gaussiano y valor de varianza 0,7. **Tabla 3.4:** Resultados calidad del sinograma con ruido gaussiano y filtro gaussiano.

En cuanto al ruido speckle, el filtro gaussiano, figura 3.5, a pesar de que su nombre pueda indicar que no sería de utilidad para este tipo de ruido, produce un buen resultado, ver tabla 3.5, con un factor de sigma igual a 0,5; situando el SSIM = 0,9074 y el PSNR = 41,0708.



sigma	SSIM	PSNR	tiempo
Con ruido	0,8703	38,7487	-
0,3	0,8703	38,7487	0,0011
0,5	0,9074	41,0708	0,0010
0,7	0,9118	39,4115	0,0011
0,9	0,8943	37,5705	0,0011

Figura 3.5: Sinograma con ruido speckle con filtrado gaussiano y valor de varianza 0,5. **Tabla 3.5:** Resultados calidad del sinograma con ruido speckle y filtro gaussiano.

3.2.2. Resultados de la función filtro_b2()

B. Filtro de mediana

En segundo lugar, este filtro con ruido gaussiano, figura 3.6, incrementa los valores de calidad del sinograma, tabla 3.6, 0,4413 para SSIM y 37,1286 para PSNR, mediante una ventana de tamaño 3.



ventana	SSIM	PSNR	tiempo
Con ruido	0,3844	34,0654	-
3	0,4413	37,1286	0,0009
5	0,4263	35,7394	0,0011
7	0,4095	33,2523	0,0019
9	0,3824	30,1743	0,0021

Figura 3.6: Sinograma con ruido gaussiano con filtrado de mediana y con tamaño de ventana 3. **Tabla 3.6:** Resultados calidad del sinograma con ruido gaussiano y filtro de mediana.

Siguiendo con el filtro de mediana en relación al sinograma de ruido speckle, figura 3.7, se observa en la tabla 3.7 una ligera mejora usando el valor de ventana 3 con el SSIM = 0,8811 y el PSNR = 39,0205.



ventana	SSIM	PSNR	tiempo
Con ruido	0,8703	38,7487	-
3	0,8811	39,0205	0,0006
5	0,8549	36,4268	0,0007
7	0,8549	33,6202	0,0021
9	0,7413	30,3401	0,0023

Figura 3.7: Sinograma con ruido speckle con filtrado de Wiener y tamaño de ventana 3. **Tabla 3.7:** Resultados calidad del sinograma con ruido speckle y filtro de mediana.

3.2.3. Resultados de la función filtro_b3()

C. Filtro de Wiener

A continuación, el filtro de Wiener para ruido gaussiano, figura 3.8, también aumenta la calidad de la imagen, ver tabla 3.8, SSIM = 0,4475 y PSNR = 36,7330 para un tamaño de ventana 3.



ventana	SSIM	PSNR	tiempo
Con ruido	0,3844	34,0654	-
3	0,4475	36,7330	0,0023
5	0,4061	34,5812	0,0020
7	0,3522	32,2436	0,0021
9	0,2847	30,1459	0,0021

Figura 3.8: Sinograma con ruido gaussiano con filtrado de Wiener y tamaño de ventana 3.

Tabla 3.8: Resultados calidad del sinograma con ruido gaussiano y filtro de Wiener.

Continuando con el presente método de filtrado sobre el sinograma con ruido speckle en la figura 3.9, del mismo modo que con el ruido gaussiano, el mejor resultado se alcanza con una ventana de tamaño 3 que obtiene un SSIM de 0,8825 y un PSNR de 39,1307 en la tabla 3.9.



ventana	SSIM	PSNR	tiempo
Con ruido	0,8703	38,7487	-
3	0,8825	39,1307	0,0012
5	0,8183	35,6420	0,0020
7	0,7525	32,8745	0,0026
9	0,6786	30,5499	0,0021

Figura 3.9: Sinograma con ruido speckle con filtrado de Wiener y tamaño de ventana 3.

Tabla 3.9: Resultados calidad del sinograma con ruido speckle y filtro de Wiener.

3.2.4. Resultados de la función filtro_b4()

D. Filtro bilateral

Finalmente, el filtro bilateral en todas las ejecuciones sobre el sinograma con ruido gaussiano, figura 3.10, acrecienta los valores SSIM y PSNR, hasta 0,4836 y 38,2506 respectivamente como se aprecia en la tabla 3.10. Teniendo en cuenta el aumento de tiempo necesario de filtrado y la prácticamente nula progresión de la calidad el valor 5 para el tamaño de ventana es el preferido.



ventana	SSIM	PSNR	tiempo
Con ruido	0,3844	34,0654	-
3	0,4770	38,1393	0,0046
5	0,4836	38,2506	0,0056
7	0,4839	38,2503	0,0070
9	0,4839	38,2504	0,0080

Figura 3.10: Sinograma con ruido gaussiano filtrado bilateral y tamaño de ventana 5.

Tabla 3.10: Resultados calidad del sinograma con ruido gaussiano y filtro bilateral.

Prosiguiendo con el filtro bilateral aplicado al sinograma con ruido speckle, figura 3.11, todas las ejecuciones que se observan en la tabla 3.11 muestran mayor calidad, la mejora de la ventana 3 con $SSIM = 0,9201$ y $PSNR = 41,1654$ es la más destacada.



ventana	SSIM	PSNR	tiempo
Con ruido	0,8703	38,7487	-
3	0,9201	41,1654	0,0043
5	0,9169	41,0604	0,0055
7	0,9169	40,7640	0,0105
9	0,9169	40,7640	0,0081

Figura 3.11: Sinograma con ruido speckle filtrado bilateral y tamaño de ventana 3.

Tabla 3.11: Resultados calidad del sinograma con ruido speckle y filtro bilateral.

Para el proceso sobre el sinograma con ruido gaussiano; el filtro bilateral es el método de filtrado que mejor resultados consigue tanto a nivel de SSIM como de PSNR, en particular la aplicación del método con una ventana de tamaño cinco, tabla 3.10. Seguidamente se sitúa el filtro gaussiano, utilizando un valor de sigma de 0,7 se obtiene una diferencia de un punto en base al PSNR, tabla 3.4. A continuación el filtro de mediana haciendo uso de una tamaño de ventana tres, consigue un valor similar pero ligeramente inferior al filtro gaussiano, tabla 3.6. Por último el filtro de Wiener, con un tamaño de ventana igual a tres, es el que menos consigue mejorar el valor del PSNR, tabla 3.8.

Respecto a los filtrados del sinograma con ruido speckle, el filtro bilateral vuelve a ser el mejor método de los probados en cuanto a los resultados de los parámetros de calidad, esta vez con un tamaño de ventana tres, tabla 3.11. A continuación, el filtro gaussiano con un valor de sigma de 0,5 se queda muy cerca de los resultados del filtro bilateral, tabla 3.5. Posteriormente se encuentra el filtro de Wiener haciendo uso de un tamaño tres de ventana, con un valor dos unidades inferior sobre el PSNR conseguido por el filtro bilateral, tabla 3.9. Finalmente el filtro de mediana se sitúa ligeramente por debajo del filtro de Wiener con la utilización también de un valor de ventana de tres, tabla 3.7.

En resumen, los filtros gaussiano y bilateral aportan unos resultados superiores, este último presenta un aumento del tiempo de ejecución cuanto mayor es la ventana de filtrado, aunque en relación a los valores observados es insignificante. Las otras dos técnicas

de filtrado, el filtro de mediana y el de Wiener, se encuentran mínimamente por detrás en cuanto a los resultados obtenidos.

3.3 Reconstrucción mediante sinogramas filtrados

Una vez realizadas todas las pruebas de filtrado, se han determinado los valores óptimos de cada uno de los filtros. Así pues, se dispone a realizar la fase de reconstrucción de la imagen utilizando los sinogramas anteriormente filtrados.

Para la reconstrucción de la imagen, no se han utilizado ni STF ni FISTA y se ha seleccionado realizar 200 iteraciones del método de reconstrucción utilizado 2.1.2. No son excesivas iteraciones, de hecho son escasas para obtener una imagen de mínima calidad, pero son suficientes para observar la acción de los métodos de filtrado o en su defecto la falta de acción. Es una manera de comprobar eficientemente el funcionamiento de los filtros dado que la reconstrucción de la imagen demanda una potencia computacional elevada.

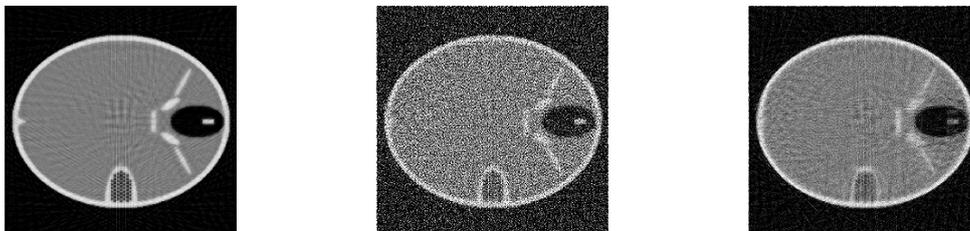
■ Ruido gaussiano

De acuerdo con los resultados obtenidos, se observan en la tabla 3.12 unas medidas de calidad muy bajas, como era de esperar, tanto del SSIM como del PSNR. Los valores son muy similares para todos los sinogramas filtrados, también para los sinogramas completamente sin ruido y con ruido sin filtrar.

	SSIM	PSNR	tiempo
Sinograma sin ruido	0,0024	6,9108	9,0128
Sinograma con ruido	0,0023	6,8854	9,0453
Filtro gaussiano	0,0025	6,8848	8,9839
Filtro de mediana	0,0020	6,8913	8,9880
Filtro de Wiener	0,0023	6,8857	8,9691
Filtro bilateral	0,0024	6,8912	8,9410

Tabla 3.12: Valores de calidad de los fantasmas reconstruidos, mediante sinogramas con ruido gaussiano filtrado, y tiempo de reconstrucción.

No obstante en la comparación visual de las imágenes resultantes mostradas en la figura 3.12, sí que son apreciables diferencias entre ellas. Se observa claramente la cantidad de ruido añadido, así como el efecto de las técnicas de filtrado.



(a) Reconstrucción sin ruido (b) Reconstrucción con ruido gaussiano (c) Reconstrucción con filtrado bilateral

Figura 3.12: Imágenes de los fantasmas reconstruidos sin ruido, con ruido gaussiano y filtrado.

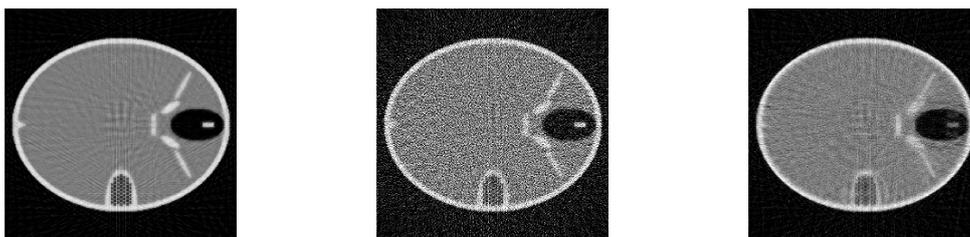
■ Ruido speckle

De igual manera que con el ruido gaussiano añadido, en la tabla 3.13 se observan también valores muy similares para el SSIM y para el PSNR, e incluso se advierten mejores valores del SSIM en algunos sinogramas filtrados respecto al sinograma sin ningún tipo de ruido.

	SSIM	PSNR	tiempo
Sinograma sin ruido	0,0024	6,9108	9,0128
Sinograma con ruido	0,0024	6,9029	9,1064
Filtro gaussiano	0,0027	6,8996	8,9180
Filtro de mediana	0,0020	6,8972	8,9299
Filtro de Wiener	0,0024	6,8934	8,9006
Filtro bilateral	0,0025	6,8995	8,8926

Tabla 3.13: Valores de calidad de los fantasmas reconstruidos, mediante sinogramas con ruido speckle filtrado, y tiempo de reconstrucción.

Sin embargo, como sucede anteriormente, sí es posible apreciar diferencias entre las imágenes reconstruidas de la figura 3.13.



(a) Reconstrucción sin ruido añadido (b) Reconstrucción con ruido speckle (c) Reconstrucción con filtrado bilateral

Figura 3.13: Imágenes de los fantasmas reconstruidos sin ruido, con ruido impulsivo y filtrado.

Esto se debe al reducido número de iteraciones del proceso de reconstrucción, pues se compara las imágenes parcialmente reconstruidas, solo 200 iteraciones, con la imagen final esperada de máxima calidad para la resolución seleccionada. Del mismo modo que anteriormente mentado, a nivel visual es factible advertir diferencias entre las imágenes, con la connotación de lograr evaluar el comportamiento de las técnicas de filtrado sobre los sinogramas.

3.4 Comparación métodos de filtrado sobre el fantoma

Como complemento al trabajo desarrollado, se han realizado pruebas de las técnicas de filtrado utilizando el fantoma de referencia, con el fin de conocer el comportamiento de los métodos de filtrado sobre la imagen completa. En primer lugar se carga el fantoma de referencia, se normaliza y se cambia el tipo de la imagen a entero. En segundo lugar se añaden el ruido, gaussiano o speckle. Seguidamente se ejecutan las mismas funciones de filtrado que los utilizados anteriormente para el sinograma, adaptadas al fantoma, todo lo dicho se puede ver en el tramo de código siguiente:

```
1 % Cargar imagen fantoma
2 load('variable_IMref')
3
4 % Normalizado y cambio de tipo
5 IMr = 255*mat2gray(IMref);
6 IMn = uint8(IMr);
7
8 % Adición de ruido
9 IMn = imnoise(IMn, 'gaussian', 0, 0.00001); %y tambien IMn = imnoise(IMn, 'speckle
   ', 0.00001);
10
11 % Ciclo for 1 a 4, numero de filtros
12 for i=1:4
13
14     % Asignacion de valor al handler
15     % i=1 - filtro_b=filtro_b1, i=2 - filtro_b=filtro_b2 ...
16     filtro_IM = str2func(['filtro_IM' num2str(i)]);
17     % Matriz solucion
18     AsolIM(:, :, i) = filtro_IM(IMr, IMn);
19
20 end
```

Los resultados de las ejecuciones con varianza 0,00001 para ambos tipos de ruido, el termino medio empleado en la ejecución de las pruebas, se muestran a continuación.

■ **Ruido gaussiano**

sigma	SSIM	PSNR	tiempo
Con ruido	0,2625	50,6557	-
0,3	0,2624	49,7310	0,0017
0,5	0,2536	30,4372	0,0011
0,7	0,2405	25,2957	0,0018
0,9	0,2785	23,4913	0,0012

Tabla 3.14: Resultados fantoma ruido gaussiano con filtro gaussiano.

ventana	SSIM	PSNR	tiempo
Con ruido	0,2625	50,6557	-
3	0,3665	23,9428	0,0014
5	0,7835	22,0250	0,0012
7	0,9088	22,0298	0,0010
9	0,8988	21,7320	0,0011

Tabla 3.15: Resultados fantoma ruido gaussiano con filtro de mediana.

ventana	SSIM	PSNR	tiempo
Con ruido	0,2625	50,6557	-
3	0,3331	42,0871	0,0047
5	0,6551	36,0427	0,0021
7	0,7613	32,5885	0,0021
9	0,7659	30,3284	0,0021

Tabla 3.16: Resultados fantoma ruido gaussiano con filtro de Wiener.

ventana	SSIM	PSNR	tiempo
Con ruido	0,2625	50,6557	-
3	0,3141	56,0841	0,0047
5	0,4236	57,5388	0,0056
7	0,4301	57,6377	0,0075
9	0,4308	57,6428	0,0089

Tabla 3.17: Resultados fantoma ruido gaussiano con filtro bilateral.

■ **Ruido speckle**

sigma	SSIM	PSNR	tiempo
Con ruido	0,6767	54,7095	-
0,3	0.6749	52.8108	0,0012
0,5	0.6491	30.4711	0,0011
0,7	0.6638	25.3020	0,0012
0,9	0.7781	23.4947	0,0012

Tabla 3.18: Resultados fantoma ruido speckle con filtro gaussiano.

ventana	SSIM	PSNR	tiempo
Con ruido	0,6767	54,7095	-
3	0.7663	23.9251	0,0006
5	0.9377	22.0670	0,0007
7	0.9296	22.0111	0,0010
9	0.9127	21.7134	0,0012

Tabla 3.19: Resultados fantoma ruido speckle con filtro de mediana.

ventana	SSIM	PSNR	tiempo
Con ruido	0,6767	54,7095	-
3	0.8342	42.2364	0,0020
5	0.8803	36.0812	0,0021
7	0.8226	32.6130	0,0020
9	0.7772	30.3452	0,0021

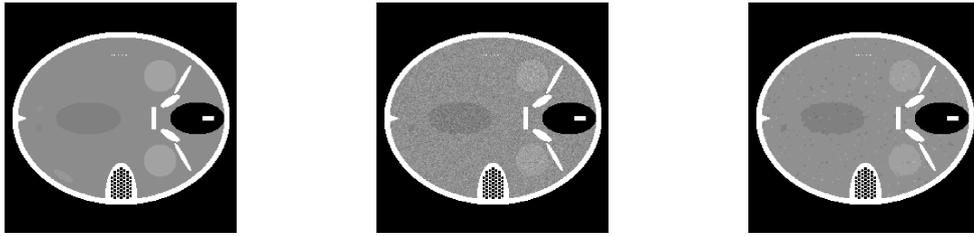
Tabla 3.20: Resultados fantoma ruido speckle con filtro de Wiener.

ventana	SSIM	PSNR	tiempo
Con ruido	0.6767	54,7095	-
3	0.8159	60.9030	0,0049
5	0.9359	62.0096	0,0059
7	0.9436	62.0605	0,0070
9	0.9436	62.0605	0,0084

Tabla 3.21: Resultados fantoma ruido speckle con filtro bilateral.

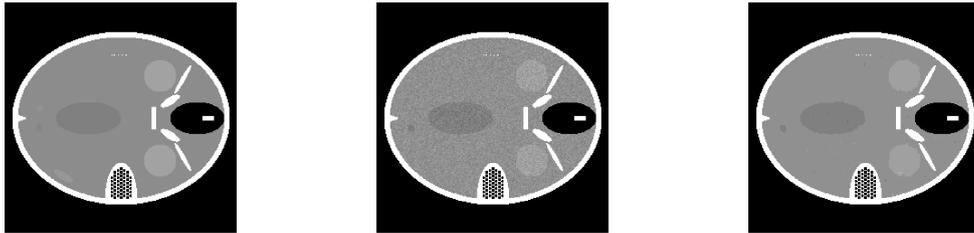
Como es posible observar, ninguna ejecución mejora ambos parámetros salvo el filtro bilateral donde se produce un incremento del SSIM hasta 0,4308 y del PSNR hasta 57,6428 en el fantoma con ruido gaussiano, tabla 3.17. También un aumento en el SSIM hasta 0,9436 y el PSNR hasta 62,0605 para el ruido speckle, tabla 3.21. Así pues, produciendo una mejora en los parámetros de calidad, en la figura 3.14 se muestra el resultado en las imágenes como comparativa con el ruido gaussiano y en la figura 3.15 el resultado en las imágenes en comparación con el ruido speckle.

Se aprecia de manera evidente el trabajo del filtro bilateral en la reducción de ruido. Elimina una gran cantidad de ruido y realiza una buena conservación de los contornos de las figuras, manteniendo los bordes definidos.



(a) Fantoma de referencia sin ruido (b) Fantoma con ruido gaussiano añadido (c) Fantoma con ruido gaussiano filtrado

Figura 3.14: Comparación fantomas sin ruido, con ruido gaussiano y filtrado



(a) Fantoma de referencia sin ruido (b) Fantoma con ruido speckle añadido (c) Fantoma con ruido speckle filtrado

Figura 3.15: Comparación fantomas sin ruido, con ruido impulsivo y filtrado

El resto de los valores obtenidos se encuentran en el anexo [A.2](#). Aun con la más mínima cantidad de ruido, se eliminan dos componentes del fantoma, esto se debe en conjunto al reducido rango de valores de los píxeles, los cambios de tipo de la imagen, en adición a los métodos no dedicados a las imágenes tan especializadas como es este fantoma.

CAPÍTULO 4

Conclusiones

En este trabajo se ha realizado un estudio de diferentes técnicas de filtrado sobre un sinograma de 60 vistas al que se le ha añadido ruido, de forma que se simule la aparición de artefactos en un fantoma matemático, que podría presentar una imagen de un sinograma obtenido de un escáner real. El estudio de los parámetros de los filtros así como la evaluación de los parámetros de calidad se ha realizado mediante la herramienta matemática MATLAB.

En el estudio del comportamiento de los filtros ha sido necesario seleccionar un nivel de ruido que refleje notablemente la acción de las técnicas de filtrado. Por ello se ha elegido un valor de varianza 0,0005 para añadir ambos tipos de ruido al sinograma. La cantidad de ruido aplicado a los sinogramas se ha elegido teniendo en cuenta que los métodos utilizados para el filtrado no han sido diseñados para ser usados específicamente sobre el tipo de imagen que es un sinograma, que es la representación de un conjunto de señales.

El diseño de las funciones para las pruebas de los parámetros de filtrado ha resultado de mucha utilidad, pues ha permitido examinar múltiples combinaciones de valores de los parámetros en cada ejecución. Además el modo de almacenar los valores adquiridos en matrices ha facilitado la lectura y el posterior análisis de los resultados.

De los datos examinados se desprende que el filtro bilateral es el que ofrece mejores resultados para el tratamiento del sinograma con ruido gaussiano así como para el sinograma con ruido multiplicativo. Esto se ha observado durante la comparación entre las diversas técnicas de filtrado elegidas.

En la parte de la reconstrucción de la imagen no se han mejorado los parámetros de calidad, incluso algún método los ha empeorado ligeramente. Esto se debe al reducido número de iteraciones que se han elegido para realizar el método de reconstrucción. Por ello, a pesar de los resultados desfavorables de las pruebas de calidad, de forma visual se ha podido apreciar claramente el efecto del filtrado en la imagen del fantoma obtenida realizando 200 iteraciones del método de reconstrucción, siendo nuevamente el filtro bilateral el que consigue una calidad de imagen superior.

También se ha realizado el proceso de filtrado sobre la imagen de referencia del fantoma matemático con una resolución de 256x256. De nuevo, el filtro bilateral ha sido el que más ha aumentado los valores de calidad de las imágenes filtradas con respecto a la imagen de referencia para los dos tipos de ruido añadido, gaussiano y multiplicativo. Produciendo una mejora de hasta el 64 % en el PSNR sobre el fantoma con ruido speckle añadido.

En cuanto a las mejoras producidas en los sinogramas filtrados, en el mejor caso han sido de alrededor de un 25 %, nada mal si se tiene en cuenta que un sinograma no es una

imagen con una estructura común y ello puede verse reflejado en valores bajos de SSIM y PSNR. Sobre las imágenes del fantoma se han producido unas mejoras del 40 % sobre el PSNR en comparación con el fantoma que presenta ruido gaussiano y de hasta el 64 % en el PSNR sobre el fantoma con ruido speckle añadido.

Como se ha podido comprobar, el filtro bilateral destaca en la capacidad que tiene para la conservación de los contornos de todo tipo de formas presentes en las imágenes, al mismo tiempo que elimina gran parte del ruido.

4.1 Trabajos futuros

Una vez concluido el trabajo quedarían abiertas diversas líneas futuras para poder continuar con el desarrollo y mejoras de los procesos de filtrado analizados:

- Generalizar las implementaciones de los programas desarrollados en MATLAB de forma que sean independientes de ciertas variables, como puede ser el tamaño de la ventana de los filtros. Esto se puede conseguir modificando las cabeceras de las funciones pasando como argumentos de entrada ciertos valores.
- Analizar el comportamiento de los filtros para diferentes resoluciones de la imagen al variar el número de proyecciones empleadas.
- En el caso del estudio del filtrado de la imagen, sería conveniente procesar los programas hasta llegar al criterio de convergencia. De esa forma se podría determinar el mejor proceso con respecto a calidad y/o tiempo de ejecución.
- En este trabajo se ha realizado un estudio de los cuatro filtros suponiendo que la imagen o el sinograma solamente contenían un tipo de ruido, gaussiano o speckle. Normalmente el ruido en una imagen no es de un solo tipo, por ello, sería interesante estudiar el comportamiento de los filtros para imágenes con ambos ruidos. Además, los filtros podrían aplicarse por separado o bien combinarse.

Bibliografía

- [1] Beckman, EC. CT scanning the early days *The British Journal of Radiology*, 79(937),5-8, (2006).
- [2] Evolución de los escáneres de TAC. Principales generaciones hasta la actualidad. Disponible: <https://radiopaedia.org/articles/ct-scanner-evolution>. [Accedido: 7 de sept. 2018]
- [3] Rehani, Madan M., et al. Managing Patient Dose in Computed Tomography. *Ann ICRP*, 30 (4), 7-45, (2000).
- [4] Kumar I., Bhadauria H.S., Virmani J., Rawat J. Reduction of Speckle noise from Medical Images using Principal Component Analysis Image Fusion. *9th International Conference on Industrial and Information Systems* pp. 1-6. doi: 10.1109, (2014).
- [5] Senthilraja S., Suresh P., Suganthi M. Noise Reduction in Computed Tomography Image Using WB - Filter. *International Journal of Scientific & Engineering Research*, 5(3), (2014).
- [6] Alonso García, D. Corrección del artefacto de anillo para un tomógrafo de rayos x de pequeños animales, Tesis de Máster. 2013
- [7] Chillarón, M. CT Medical Imaging Reconstruction Using Direct Algebraic Methods with Few Projections. *International Conference on Computational Science*. Springer, Cham, 334-346, (2018)
- [8] Flores, L., et al. Parallel CT image reconstruction based on GPUs. *Radiation Physics and Chemistry*. 95: 247-250. (2014)
- [9] Parceró, E., et al. Impact of view reduction in CT on radiation dose for patients. *Radiation Physics and Chemistry*. 137: 173-175. (2017)
- [10] Joseph P.M. An Improved Algorithm for Reprojecting Rays through Pixel Images. *IEEE transactions on medical imaging*, 1(3), 192-196, (1982).
- [11] Lauritsch G., Bruder H. Forbild Phantom Group, *Head Phantom*. [En línea]. Disponible: <http://www.imp.uni-erlangen.de/phantoms/head/head.html>. [Accedido: 7 de sept. 2018]
- [12] Paige C., Saunders M. A. LSQR: An algorithm for sparse linear equations and sparse least squares. *ACM transactions on mathematical software*, 8, 43-71, (1982).
- [13] Yu H., Wang G. A soft-threshold filtering approach for reconstruction from a limited number of projections. *Physics in Medicine and Biology*, 55(13), 3905-3916, (2010).
- [14] Yu W., Zeng L. A novel weighted total difference based image reconstruction algorithm for few-view computed tomography. *PLoS ONE*, 9(10), 1-10, (2014).

-
- [15] Beck A., Teboulle M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1), 183-202, (2009).
- [16] Hore, A., Ziou, D. Image quality metrics: PSNR vs. SSIM. *Pattern recognition (icpr), 20th international conference on*. IEEE. p. 2366-2369. (2010)
- [17] Jain, A. K. Fundamentals of digital image processing. *Englewood Cliffs, NJ: Prentice Hall*,. (1989).
- [18] Lim, Jae S. Two-Dimensional Signal and Image Processing, *Englewood Cliffs, NJ, Prentice Hall*, 469-476, (1990).
- [19] Lim, Jae S. Two-Dimensional Signal and Image Processing, *Englewood Cliffs, NJ, Prentice Hall*, 548, (1990).
- [20] Tomasi, C., and R. Manduchi. Bilateral Filtering for Gray and Color Images *Proceedings of the 1998 IEEE International Conference on Computer Vision*. Bombay, India. 836-846, (1998).

APÉNDICE A

Resultados estudio de parámetros

En este anexo se encuentran los resultados de todas la pruebas de calidad de las imágenes, efectuadas haciendo uso de los diferentes parámetros de cada filtro.

A.1 Tablas de los sinogramas

Resultados de los filtros aplicados a los sinogramas con ruido.

A.1.1. Ruido gaussiano

varianza

Filtro gaussiano

sigma	SSIM	PSNR	tiempo
0,3	0,5701	50,1945	0,0012
0,5	0,5657	45,5727	0,0011
0,7	0,5463	40,2661	0,0012
0,9	0,5362	37,8728	0,0011

Filtro de mediana

ventana	SSIM	PSNR	tiempo
3	0,5871	40,1154	0,0009
5	0,8196	37,2237	0,0007
7	0,8333	34,1833	0,0019
9	0,7486	30,2825	0,0021

Filtro de Wiener

ventana	SSIM	PSNR	tiempo
3	0,5295	39,2552	0,0020
5	0,6516	35,8207	0,0021
7	0,7190	32,9422	0,0022
9	0,6832	30,5988	0,0025

Filtro bilateral

ventana	SSIM	PSNR	tiempo
3	0,5539	42,0688	0,0045
5	0,5725	41,4207	0,0057
7	0,5750	41,3745	0,0071
9	0,5755	41,3721	0,0083

varianza

0,0001

Filtro gaussiano

sigma	SSIM	PSNR	tiempo
0,3	0,4988	41,0404	0,0130
0,5	0,5251	42,1288	0,0036
0,7	0,5208	39,4454	0,0047
0,9	0,5016	37,4732	0,0135

Filtro de mediana

ventana	SSIM	PSNR	tiempo
3	0,4938	39,3390	0,0055
5	0,4851	36,8555	0,0036
7	0,5262	33,9653	0,0076
9	0,5373	30,2951	0,0070

Filtro de Wiener

ventana	SSIM	PSNR	tiempo
3	0,4862	38,6262	0,0466
5	0,4286	35,5348	0,0059
7	0,3698	32,7781	0,0070
9	0,3023	30,4931	0,0078

Filtro bilateral

ventana	SSIM	PSNR	tiempo
3	0,5217	41,0087	0,0072
5	0,5178	40,6047	0,0080
7	0,5175	40,5754	0,0085
9	0,5175	40,5747	0,0203

varianza

0,0005

Filtro gaussiano

sigma	SSIM	PSNR	tiempo
0,3	0,3844	34,0655	0,0014
0,5	0,4409	36,7953	0,0011
0,7	0,4693	37,1464	0,0017
0,9	0,4668	36,1990	0,0011

Filtro de mediana

ventana	SSIM	PSNR	tiempo
3	0,4413	37,1286	0,0009
5	0,4263	35,7394	0,0011
7	0,4095	33,2523	0,0019
9	0,3824	30,1743	0,0021

Filtro de Wiener

ventana	SSIM	PSNR	tiempo
3	0,4475	36,7330	0,0023
5	0,4061	34,5812	0,0020
7	0,3522	32,2436	0,0021
9	0,2847	30,1459	0,0021

Filtro bilateral

ventana	SSIM	PSNR	tiempo
3	0,4770	38,1393	0,0046
5	0,4836	38,2506	0,0056
7	0,4839	38,2503	0,0070
9	0,4839	38,2504	0,0080

varianza

0,001

Filtro gaussiano

sigma	SSIM	PSNR	tiempo
0,3	0,3244	31,1420	0,0011
0,5	0,3896	34,1129	0,0011
0,7	0,4325	35,4091	0,0011
0,9	0,4392	35,0623	0,0011

Filtro de mediana

ventana	SSIM	PSNR	tiempo
3	0,4062	35,6507	0,0006
5	0,4030	34,8605	0,0007
7	0,3850	32,7479	0,0020
9	0,3552	30,0257	0,0022

Filtro de Wiener

ventana	SSIM	PSNR	tiempo
3	0,4190	35,2001	0,0019
5	0,3921	33,7595	0,0021
7	0,3405	31,7276	0,0021
9	0,2746	29,8246	0,0021

Filtro bilateral

ventana	SSIM	PSNR	tiempo
3	0,4373	35,9577	0,0045
5	0,4485	36,2722	0,0057
7	0,4491	36,2825	0,0065
9	0,4491	36,2828	0,0079

varianza

0,005

Filtro gaussiano

sigma	SSIM	PSNR	tiempo
0,3	0,1805	24,2762	0,0011
0,5	0,2458	27,4077	0,0011
0,7	0,3111	29,7862	0,0011
0,9	0,3390	30,5065	0,0011

Filtro de mediana

ventana	SSIM	PSNR	tiempo
3	0,2913	30,6234	0,0006
5	0,3211	31,6336	0,0007
7	0,3088	30,3668	0,0021
9	0,2778	28,6760	0,0022

Filtro de Wiener

ventana	SSIM	PSNR	tiempo
3	0,3030	29,5891	0,0019
5	0,3287	29,9549	0,0020
7	0,2978	29,0337	0,0020
9	0,2384	27,8581	0,0021

Filtro bilateral

ventana	SSIM	PSNR	tiempo
3	0,2564	28,0995	0,0047
5	0,2650	28,5568	0,0056
7	0,2654	28,5859	0,0066
9	0,2654	28,5866	0,0079

varianza

0,01

Filtro gaussiano

sigma	SSIM	PSNR	tiempo
0,3	0,1294	21,3744	0,0011
0,5	0,1872	24,4973	0,0011
0,7	0,2530	27,0354	0,0010
0,9	0,2865	27,9708	0,0011

Filtro de mediana

ventana	SSIM	PSNR	tiempo
3	0,2386	27,9594	0,0060
5	0,2770	29,6321	0,0070
7	0,2655	28,8795	0,0021
9	0,2363	27,6196	0,0024

Filtro de Wiener

ventana	SSIM	PSNR	tiempo
3	0,2362	26,5343	0,0019
5	0,2773	27,4993	0,0020
7	0,2640	27,0698	0,0029
9	0,2147	26,2700	0,0021

Filtro bilateral

ventana	SSIM	PSNR	tiempo
3	0,1701	23,8845	0,0044
5	0,1749	24,2377	0,0057
7	0,1751	24,2628	0,0104
9	0,1751	24,2634	0,0079

A.1.2. Ruido speckle

varianza

0,0001

Filtro gaussiano

sigma	SSIM	PSNR	tiempo
0,3	0,9524	45,4984	0,0012
0,5	0,9597	44,6206	0,0012
0,7	0,9384	40,1563	0,0011
0,9	0,9101	37,8534	0,0010

Filtro de mediana

ventana	SSIM	PSNR	tiempo
3	0,9170	39,8341	0,0060
5	0,8783	37,0981	0,0070
7	0,8352	34,0952	0,0020
9	0,7536	30,3153	0,0023

Filtro de Wiener

ventana	SSIM	PSNR	tiempo
3	0,9017	39,1886	0,0019
5	0,8284	35,8031	0,0019
7	0,7614	32,9422	0,0020
9	0,6892	30,6017	0,0020

Filtro bilateral

ventana	SSIM	PSNR	tiempo
3	0,9404	41,9428	0,0045
5	0,9304	41,3546	0,0056
7	0,9298	41,3115	0,0065
9	0,9298	41,3098	0,0081

varianza

0,0005

Filtro gaussiano

sigma	SSIM	PSNR	tiempo
0,3	0,8703	38,7487	0,0011
0,5	0,9074	41,0708	0,0010
0,7	0,9118	39,4115	0,0011
0,9	0,8943	37,5705	0,0011

Filtro de mediana

ventana	SSIM	PSNR	tiempo
3	0,8811	39,0205	0,0006
5	0,8549	36,4268	0,0007
7	0,8142	33,6202	0,0021
9	0,7413	30,3401	0,0023

Filtro de Wiener

ventana	SSIM	PSNR	tiempo
3	0,8825	39,1307	0,0012
5	0,8183	35,6420	0,0020
7	0,7525	32,8745	0,0026
9	0,6786	30,5499	0,0021

Filtro bilateral

ventana	SSIM	PSNR	tiempo
3	0,9201	41,1654	0,0043
5	0,9169	41,0604	0,0055
7	0,9169	40,7640	0,0105
9	0,9169	40,7640	0,0081

varianza

0,001

Filtro gaussiano

sigma	SSIM	PSNR	tiempo
0,3	0,8177	35,8114	0,0011
0,5	0,8683	38,7508	0,0010
0,7	0,8882	38,5776	0,0010
0,9	0,8784	37,1954	0,0010

Filtro de mediana

ventana	SSIM	PSNR	tiempo
3	0,8539	37,4341	0,0060
5	0,8365	35,8132	0,0060
7	0,7970	33,1734	0,0020
9	0,7308	30,3200	0,0023

Filtro de Wiener

ventana	SSIM	PSNR	tiempo
3	0,8656	37,9767	0,0019
5	0,8086	35,3357	0,0019
7	0,7456	32,7198	0,0020
9	0,6724	30,4637	0,0021

Filtro bilateral

ventana	SSIM	PSNR	tiempo
3	0,8994	39,8122	0,0042
5	0,9008	39,7949	0,0055
7	0,9011	39,7872	0,0071
9	0,9011	39,7868	0,0076

varianza

0,005

Filtro gaussiano

sigma	SSIM	PSNR	tiempo
0,3	0,6718	28,8874	0,0011
0,5	0,7361	32,4577	0,0010
0,7	0,7900	34,8961	0,0010
0,9	0,8036	35,1278	0,0010

Filtro de mediana

ventana	SSIM	PSNR	tiempo
3	0,7436	32,8725	0,0005
5	0,7530	32,9418	0,0007
7	0,7257	31,3765	0,0021
9	0,6768	29,5288	0,0024

Filtro de Wiener

ventana	SSIM	PSNR	tiempo
3	0,7857	34,5932	0,0019
5	0,7657	33,8005	0,0020
7	0,7146	31,8804	0,0021
9	0,6467	29,9524	0,0021

Filtro bilateral

ventana	SSIM	PSNR	tiempo
3	0,7741	33,9281	0,0043
5	0,7821	34,3985	0,0058
7	0,7827	34,4255	0,0071
9	0,7827	34,4262	0,0079

varianza

0,01

Filtro gaussiano

sigma	SSIM	PSNR	tiempo
0,3	0,6138	25,9920	0,0011
0,5	0,6727	29,6021	0,0010
0,7	0,7321	32,6279	0,0010
0,9	0,7545	33,5245	0,0010

Filtro de mediana

ventana	SSIM	PSNR	tiempo
3	0,6859	30,4299	0,0006
5	0,7127	31,4429	0,0006
7	0,6957	30,3437	0,0023
9	0,6534	28,8253	0,0025

Filtro de Wiener

ventana	SSIM	PSNR	tiempo
3	0,7162	31,7079	0,0019
5	0,7321	32,3285	0,0019
7	0,6988	31,0607	0,0021
9	0,6361	29,3875	0,0020

Filtro bilateral

ventana	SSIM	PSNR	tiempo
3	0,6824	29,7709	0,0044
5	0,6864	30,1793	0,0055
7	0,6867	30,2033	0,0066
9	0,6867	30,2037	0,0079

varianza

0,05

Filtro gaussiano

sigma	SSIM	PSNR	tiempo
0,3	0,5160	19,5456	0,0011
0,5	0,5479	23,0306	0,0010
0,7	0,5939	26,3840	0,0011
0,9	0,6202	27,9859	0,0011

Filtro de mediana

ventana	SSIM	PSNR	tiempo
3	0,5691	24,0928	0,0006
5	0,6041	26,1873	0,0006
7	0,6070	26,3643	0,0024
9	0,5874	25,6949	0,0026

Filtro de Wiener

ventana	SSIM	PSNR	tiempo
3	0,5628	24,0202	0,0018
5	0,5657	25,2226	0,0019
7	0,5563	25,4919	0,0020
9	0,5414	25,3693	0,0020

Filtro bilateral

ventana	SSIM	PSNR	tiempo
3	0,5215	20,4630	0,0043
5	0,5186	20,6066	0,0057
7	0,8185	20,6160	0,0064
9	0,5185	20,6164	0,0080

varianza

0,1

Filtro gaussiano

sigma	SSIM	PSNR	tiempo
0,3	0,4909	16,9046	0,0011
0,5	0,5122	20,3445	0,0010
0,7	0,5462	23,6092	0,0011
0,9	0,5678	25,1812	0,0010

Filtro de mediana

ventana	SSIM	PSNR	tiempo
3	0,5337	21,3920	0,0006
5	0,5652	23,6243	0,0009
7	0,5723	24,0073	0,0023
9	0,5559	23,5532	0,0026

Filtro de Wiener

ventana	SSIM	PSNR	tiempo
3	0,5185	21,0871	0,0018
5	0,5141	22,0889	0,0019
7	0,5005	22,3809	0,0020
9	0,4853	22,3903	0,0021

Filtro bilateral

ventana	SSIM	PSNR	tiempo
3	0,4888	17,2887	0,0046
5	0,4854	17,3688	0,0060
7	0,4851	17,3743	0,0064
9	0,4851	17,3745	0,0076

A.2 Tablas de los fantomas

Resultados de los filtros aplicados a los fantomas con ruido.

A.2.1. Ruido gaussiano

varianza

0,000001

Filtro gaussiano

sigma	SSIM	PSNR	tiempo
0,3	0,4180	56,0380	0,0133
0,5	0,3981	30,4949	0,0033
0,7	0,8948	25,3109	0,0047
0,9	0,8653	23,5052	0,0129

Filtro de mediana

ventana	SSIM	PSNR	tiempo
3	0,9687	23,9147	0,0053
5	0,9545	22,0564	0,0024
7	0,9423	21,9984	0,0070
9	0,9244	21,6972	0,0061

Filtro de Wiener

ventana	SSIM	PSNR	tiempo
3	0,9643	42,4898	0,0480
5	0,8923	36,2045	0,0060
7	0,8322	32,6829	0,0072
9	0,7868	30,4008	0,0079

Filtro bilateral

ventana	SSIM	PSNR	tiempo
3	0,9860	64,1136	0,0070
5	0,9866	64,1149	0,0080
7	0,9867	64,1184	0,0093
9	0,9867	64,1184	0,0202

varianza

0,00001

Filtro gaussiano

sigma	SSIM	PSNR	tiempo
0,3	0,2624	49,7310	0,0017
0,5	0,2536	30,4372	0,0011
0,7	0,2405	25,2957	0,0018
0,9	0,2785	23,4913	0,0012

Filtro de mediana

ventana	SSIM	PSNR	tiempo
3	0,3665	23,9428	0,0014
5	0,7835	22,0250	0,0012
7	0,9088	22,0298	0,0010
9	0,8988	21,7320	0,0011

Filtro de Wiener

ventana	SSIM	PSNR	tiempo
3	0,3331	42,0871	0,0047
5	0,6551	36,0427	0,0021
7	0,7613	32,5885	0,0021
9	0,7659	30,3284	0,0021

Filtro bilateral

ventana	SSIM	PSNR	tiempo
3	0,3141	56,0841	0,0047
5	0,4236	57,5388	0,0056
7	0,4301	57,6377	0,0075
9	0,4308	57,6428	0,0089

varianza

0,0001

Filtro gaussiano

sigma	SSIM	PSNR	tiempo
0,3	0,2382	41,1942	0,0011
0,5	0,2329	30,1997	0,0011
0,7	0,2130	25,2254	0,0011
0,9	0,1955	23,4464	0,0011

Filtro de mediana

ventana	SSIM	PSNR	tiempo
3	0,2404	23,9912	0,0006
5	0,2681	22,1369	0,0006
7	0,3396	22,0310	0,0010
9	0,4485	21,8061	0,0011

Filtro de Wiener

ventana	SSIM	PSNR	tiempo
3	0,2497	40,6514	0,0020
5	0,2375	35,5072	0,0020
7	0,2309	32,2703	0,0021
9	0,2789	30,1103	0,0021

Filtro bilateral

ventana	SSIM	PSNR	tiempo
3	0,2530	47,1918	0,0047
5	0,2563	47,8505	0,0057
7	0,2565	47,8998	0,0067
9	0,2565	47,9010	0,0085

A.2.2. Ruido speckle

varianza

0,000001

Filtro gaussiano

sigma	SSIM	PSNR	tiempo
0,3	0,9866	57,1515	0,0011
0,5	0,9506	30,4984	0,0012
0,7	0,9026	25,3115	0,0012
0,9	0,8676	23,5055	0,0012

Filtro de mediana

ventana	SSIM	PSNR	tiempo
3	0,9709	23,9131	0,0006
5	0,9567	22,0550	0,0007
7	0,9447	21,9970	0,0011
9	0,9236	21,6947	0,0012

Filtro de Wiener

ventana	SSIM	PSNR	tiempo
3	0,9679	42,5372	0,0021
5	0,8969	36,2223	0,0021
7	0,8357	32,6922	0,0022
9	0,7864	30,4100	0,0021

Filtro bilateral

ventana	SSIM	PSNR	tiempo
3	0,9887	64,1888	0,0047
5	0,9887	64,1888	0,0059
7	0,9887	64,1888	0,0068
9	0,9887	64,1888	0,0085

varianza

0,00001

Filtro gaussiano

sigma	SSIM	PSNR	tiempo
0,3	0,6749	52,8108	0,0012
0,5	0,6491	30,4711	0,0011
0,7	0,6638	25,3020	0,0012
0,9	0,7781	23,4947	0,0012

Filtro de mediana

ventana	SSIM	PSNR	tiempo
3	0,7663	23,9251	0,0006
5	0,9377	22,0670	0,0007
7	0,9296	22,0111	0,0010
9	0,9127	21,7134	0,0012

Filtro de Wiener

ventana	SSIM	PSNR	tiempo
3	0,8342	42,2364	0,0020
5	0,8803	36,0812	0,0021
7	0,8226	32,6130	0,0020
9	0,7772	30,3452	0,0021

Filtro bilateral

ventana	SSIM	PSNR	tiempo
3	0,8159	60,9030	0,0049
5	0,9359	62,0096	0,0059
7	0,9436	62,0605	0,0070
9	0,9436	62,0605	0,0084

varianza

0,0001

Filtro gaussiano

sigma	SSIM	PSNR	tiempo
0,3	0,6584	46,5540	0,0011
0,5	0,6387	30,3753	0,0011
0,7	0,6040	25,2678	0,0012
0,9	0,5789	23,4728	0,0011

Filtro de mediana

ventana	SSIM	PSNR	tiempo
3	0,6526	23,9446	0,0007
5	0,6489	22,0886	0,0006
7	0,6570	22,0390	0,0011
9	0,7110	21,7485	0,0012

Filtro de Wiener

ventana	SSIM	PSNR	tiempo
3	0,6666	41,5332	0,0019
5	0,6564	35,7464	0,0021
7	0,7326	32,3994	0,0020
9	0,7540	30,2028	0,0021

Filtro bilateral

ventana	SSIM	PSNR	tiempo
3	0,6766	53,0232	0,0045
5	0,6811	53,8454	0,0058
7	0,6815	53,9177	0,0074
9	0,6816	53,9207	0,0084