

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

TRABAJO FINAL DE MÁSTER

Estudio e Implementación en GPU de un Algoritmo de Eliminación de Ruido en Imagen Médica

Autora:

Estíbaliz PARCERO IGLESIAS

Directores:

Dr. Vicente VIDAL GIMENO

Dr. Gumersindo VERDÚ MARTÍN

Trabajo Final del Máster Universitario en Computación Paralela y Distribuida

del

Departamento de Sistemas Informáticos y Computación

16 de febrero de 2015

“Divide each of the difficulties under examination into as many parts as possible, and as might be necessary for its adequate solution”

René Descartes

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Resumen

Departamento de Sistemas Informáticos y Computación

Máster Universitario en Computación Paralela y Distribuida

Estudio e Implementación en GPU de un Algoritmo de Eliminación de Ruido en Imagen Médica

por Estíbaliz PARCERO IGLESIAS

Una de las tareas cruciales de la informática dentro de la imagen médica es la de la aplicación de algoritmos para la reducción de ruido ya que el ruido en una imagen médica puede dificultar un diagnóstico preciso.

En este trabajo se implementa el filtro *Fuzzy Peer Group Averaging* (Fuzzy PGA), que ha mostrado buenos resultados en imágenes a color, y se adapta para su utilización en mamogramas.

Para configurar correctamente el filtro se ha realizado un experimento multiparamétrico utilizando infraestructura grid. Tras ello se ha comprobado que la eficacia de la técnica implementada es adecuada mediante un estudio comparativo. Mediante este estudio Fuzzy PGA ha demostrado ser más eficaz que otras técnicas al obtener mayores puntuaciones de PSNR (*Peak Signal to Noise Ratio*). Por último se ha realizado un estudio de estabilidad de la técnica demostrando que se comporta correctamente con independencia a las características del mamograma.

Otro aspecto importante de los filtros es la eficiencia temporal, para lograrla se ha implementado una segunda versión de la técnica utilizando programación paralela en GPU. Logrando un *speedup* de hasta 12 unidades.

El filtro Fuzzy PGA ha obtenido buenos resultados en eficacia, estabilidad y eficiencia, demostrando ser un buen candidato para ser empleado como filtro en imagen médica.

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Abstract

Departamento de Sistemas Informáticos y Computación

Máster Universitario en Computación Paralela y Distribuida

Estudio e Implementación en GPU de un Algoritmo de Eliminación de Ruido en Imagen Médica

por Estíbaliz PARCERO IGLESIAS

One of the main tasks of computer science in medical imaging is the design and application of denoising algorithms, as noise in medical images can lead to an imprecise diagnostic.

In this work, the filter *Fuzzy Peer Group Averaging* (Fuzzy PGA), which has shown good results in conventional colour images, has been implemented and adapted to its use with mammograms.

First of all, a multiparametric experiment has been conducted in order to configure the implemented filter. Secondly, the performance has been evaluated in terms of efficacy, through a comparative study. This study shows that Fuzzy PGA performs better than other techniques, since it obtains better scores of PSNR (*Peak Signal to Noise Ratio*). Lastly, a stability study has been performed to demonstrate this technique shows an adequate behaviour regardless the properties in the mammogram.

Another critical point in the implementation of a new filter is the efficiency, to achieve it, another version of the filter has been implemented using parallel computation in GPU. Reaching a speed up of 12 units.

The Fuzzy PGA filter has obtained good results in efficacy, stability, and efficiency, proving to be an excellent candidate to be employed in medical image.

Agradecimientos

Primero me gustaría expresar mi más sincero agradecimiento a mis directores de trabajo final de máster, a Vicent Vidal por su esfuerzo, paciencia y dedicación y a Gumersindo Verdú por darme la magnífica oportunidad de trabajar en el Instituto de Seguridad Industrial, Radiofísica y Medioambiental.

Hago extensivo este agradecimiento a todos los profesores del Máster Universitario en Computación Paralela y Distribuida, en especial a Damián Segrelles y Nacho Blanquer por la confianza depositada y a Pedro Alonso por sus valiosas sugerencias.

Me gustaría también dar las gracias al profesor Josep Arnal por su inestimable ayuda.

Y por supuesto a mi familia, mi pareja y amigos por estar ahí.

Índice general

Resumen	v
Abstract	vii
Agradecimientos	ix
Índice general	x
Lista de figuras	xiii
Lista de tablas	xv
1. Introducción	1
1.1. Motivación	1
1.2. Objetivos	3
1.3. Organización	4
2. Fundamentos y Antecedentes	5
2.1. Filtrado de Imágenes	5
2.1.1. Medidas de Calidad de Imagen	5
2.1.2. Filtrado de Ruido Gaussiano	6
2.1.3. Filtrado de Ruido Impulsivo	8
2.2. Filtrado de Imágenes Médicas	9
2.3. Paralelización de Filtrado de Imagen mediante GPU	10
3. Materiales	11
3.1. Filtro Fuzzy Peer Group Averaging (Fuzzy PGA)	11
3.2. Base de Datos Mini-MIAS	13
4. Obtención de los Parámetros Óptimos de Entrada para Fuzzy PGA	15
4.1. Entorno de Pruebas	15
4.2. Descripción del Experimento Multiparamétrico	16
4.3. Resultados	18
4.4. Conclusiones	19
5. Estudio Comparativo de Técnicas de Filtrado	21
5.1. Técnicas de Filtrado	21

5.1.1. Filtro Peer Group Fuzzy Metric (PGFM)	21
5.1.2. Filtro de Difusión no Lineal (Non-linear Diffusive Filter, NDF)	22
5.1.3. Filtro Fuzzy Non-linear Diffusion (FNLDF)	23
5.2. Descripción del Estudio Comparativo	23
5.3. Resultados	24
5.4. Conclusiones	26
6. Estudio de Estabilidad del Filtro Fuzzy PGA en la Base de Datos mini-MIAS	27
6.1. Descripción del Estudio de Estabilidad	27
6.2. Resultados	28
6.3. Conclusiones	30
7. Implementación Paralela en GPU del Filtro Fuzzy PGA	33
7.1. Introducción a la computación en GPU	33
7.2. Fuzzy PGA para GPU	36
7.3. Resultados	37
7.4. Conclusiones	37
8. Conclusiones	39
9. Trabajo Futuro	43
Bibliografía	45

Índice de figuras

4.1.	Flujo de la ejecución secuencial.	17
4.2.	Flujo de la ejecución grid.	18
4.3.	Valor PSNR del filtrado mediante Fuzzy PGA en función del F_t con un $F_\sigma = 480$ (imagen 6 de mini-MIAS con ruido impulsivo).	19
4.4.	Valor PSNR del filtrado mediante Fuzzy PGA en función del F_σ con un $F_t = 0,85$ (imagen 6 de mini-MIAS con ruido impulsivo).	19
5.1.	(a) Imagen con ruido impulsivo ($D=0,10$), (b) Filtrada (PGFM), (c) Filtrada (FNLDF), (d) Filtrada (NDF), (e) Filtrada (Fuzzy PGA)	24
5.2.	(a) Imagen con ruido gaussiano ($\sigma = 0,01$), (b) Filtrada (PGFM), (c) Filtrada (FNLDF), (d) Filtrada (NDF), (e) Filtrada (Fuzzy PGA)	25
5.3.	(a) Imagen con ruido impulsivo ($D = 0,10$) y gaussiano ($\sigma = 0,01$), (b) Filtrada (PGFM), (c) Filtrada (FNLDF), (d) Filtrada (NDF), (e) Filtrada (Fuzzy PGA)	26
6.1.	(a) Imagen original, (b) Imagen ruidosa (impulsivo, $D = 0,10$ y gaussiano, $\sigma = 0,01$), (c) Filtrada con Fuzzy PGA	28
6.2.	Histograma de frecuencia en función de los valores de PSNR entre imagen original y filtrada por clase de tejido: F (graso), G (graso-glandular), D (denso-glandular).	29
6.3.	Histograma de frecuencia en función de los valores de PSNR entre imagen original y filtrada por clase de anomalía NORM (normal), CALC (calcificación), CIRC (masa bien definida/circunscrita), SPIC (masa espigada), MISC (otras, masas no definidas), ARCH (distorsión estructural), ASYM (asimetría).	30
6.4.	Histograma de frecuencia en función de los valores de PSNR entre imagen original y filtrada por tipo de diagnóstico: B (benigno), M (maligno).	31
7.1.	Modelo de programación de CUDA.	34
7.2.	Modelo de memoria para la programación en CUDA.	35
7.3.	Comparación de tiempo de ejecución entre la implementación secuencial ejecutada en CPU y la implementación paralela para GPU del filtro Fuzzy PGA: (a) aplicando los dos pasos de filtrado, (b) aplicando el paso de filtrado de ruido impulsivo y (c) aplicando el paso de filtrado de ruido gaussiano.	38

Lista de Tablas

4.1. Valores óptimos para F_t y F_σ	19
5.1. Valores óptimos para m y d	24
5.2. Resultados de calidad para una imagen con ruido impulsivo ($D = 0,10$) . .	24
5.3. Resultados de calidad para una imagen con ruido gaussiano ($\sigma = 0,01$) . .	25
5.4. Resultados de calidad para una imagen con ruido impulsivo ($D = 0,10$) y gaussiano ($\sigma = 0,01$)	25
6.1. Ejemplo de datos cargados de la base de datos mini-MIAS	29

Capítulo 1

Introducción

1.1. Motivación

La informática de imagen médica es un campo multidisciplinar dentro de la ciencia y de la ingeniería que se encarga de diversas tareas tales como: recuperación, transferencia, almacenamiento, procesado, muestra, comprensión y uso de la información en medicina. Otras disciplinas relacionadas con la informática de imagen médica son por ejemplo, la ingeniería eléctrica, la física médica, la fisiología y la psicología.

Uno de los aspectos que trata la informática dentro de la imagen médica es el procesado del ruido de la imagen. Y es que son diversos los factores que pueden afectar a una imagen médica, ya sea analógica o digital. Estos factores provocan variaciones aleatorias en el brillo o color de la imagen, y a este fenómeno se le conoce como ruido [1]. El ruido dota a la imagen de una apariencia sucia, sin embargo, el mayor problema radica en que el ruido puede ocultar o reducir la visibilidad de ciertas características críticas de la imagen.

En el caso de la imagen médica, el ruido puede dificultar el diagnóstico. Por ejemplo, en una mamografía (la radiografía de una mama) puede pasar inadvertida una microcalcificación. Para acentuar el contraste en una mamografía se utiliza una energía de rayos X muy inferior a la utilizada en una radiografía de propósito general, por lo tanto los sistemas de rayos y detectores se diseñan específicamente para esto.

Es cierto que es posible reducir el ruido cambiando ciertos parámetros relacionados con la toma de la imagen, en el caso de una radiografía sería posible aumentar la dosis, sin embargo debemos encontrar una solución de compromiso entre esto y la exposición del paciente a la radiación. En el caso de la mamografía, además de emplearse en el apoyo al diagnóstico de mujeres con síntomas como, por ejemplo, la presencia de masas

(mamografía diagnóstica), ésta también se emplea en la prevención de cáncer de mama, es decir, en mujeres asintomáticas (mamografía de detección), en el que el uso de una dosis excesiva para la mejora del ruido es injustificable.

La necesidad de reducir la dosis radiactiva es especialmente importante en imágenes de tomografía computarizada (CT) [2]. Una imagen CT se toma mediante la irradiación de rayos X desde distintos ángulos para después ser reconstruida por ordenador, haciendo posible obtener una imagen tridimensional. El auge de esta técnica ha provocado que aumente la media de la carga radiactiva de la población.

Un modo de reducir el ruido en una imagen médica es mediante el uso de técnicas de limpieza o filtrado de imágenes. Este tipo de técnicas pueden mejorar sustancialmente la imagen tomada. Se trata de un campo dentro del procesamiento de la imagen digital, y por lo tanto significa utilizar algoritmos para manipular imágenes. Mediante el filtrado se obtiene una imagen más limpia que facilita la interpretación clínica o la recuperación automática de información.

Existe una multitud de algoritmos diseñados especialmente para cada tipo de ruido. Los tipos más comunes de ruido son el ruido gaussiano y el ruido impulsivo. El ruido gaussiano aparece durante la adquisición de la imagen debido a un dispositivo en mala condiciones (o falta de iluminación, altas temperaturas...), como podría ser el sensor en una cámara o un escáner. Como consecuencia se obtiene un emborronamiento de todos los píxeles de la imagen, un efecto que se conoce como efecto bruma (*blur*). Por otro lado, está el ruido impulsivo que aparece durante la transmisión de la imagen por un canal contaminado, afectando sólo a ciertos píxeles de la imagen [3].

En este trabajo se estudia la técnica de filtrado *Fuzzy Peer Group Averaging* (Fuzzy PGA). Esta técnica combina dos métodos para la limpieza de ruido impulsivo y gaussiano. Ha demostrado un buen comportamiento en imágenes convencionales a color [4], y se aplica a imágenes mamográficas, buscando que se mantenga la eficacia de la técnica y no elimine información relevante.

Los métodos de filtrado son algoritmos más o menos sofisticados que consumen muchos recursos computacionales. Por una parte, para un resultado satisfactorio, es necesario configurar adecuadamente los parámetros de entrada que dependerán de la naturaleza de la imagen. A fin de determinar estos parámetros, podría ser necesario un estudio detallado, para lo cual resulta de especial interés técnicas como la computación *grid*, que encaja especialmente bien en estudios multiparamétricos. En este trabajo, el primer paso para configurar nuestra implementación de Fuzzy PGA será encontrar los parámetros óptimos de entrada, ese estudio se realiza con la ayuda de la computación *grid* mediante un experimento multiparamétrico.

Por otra parte, la propia ejecución del filtro es costosa, y para casos en los que la velocidad de ejecución sea importante (por ejemplo en tratamiento en tiempo real, o para transmisión de vídeo) se podría optar por la paralelización del método.

La computación paralela ha sido un campo de estudio de especial interés en estas últimas décadas, en especial con el descenso del coste de las tarjetas gráficas, cuyas GPUs (*Graphic Processing Unit*) pueden utilizarse incluso para la computación de algoritmos no relacionados con gráficos. Este nuevo campo de estudio se denomina computación de propósito general en GPUs (GPGPU). La plataforma de programación más utilizada en el ámbito de la GPGPU es CUDA (*Compute Unified Device Architecture*)

El rápido incremento en el rendimiento del *hardware* de gráficos, junto con recientes mejoras en su facilidad de programación, han hecho que el *hardware* de gráficos sea una plataforma atractiva bajo el punto de vista computacional para una gran cantidad de aplicaciones de distintos dominios. Hoy en día, con el abaratamiento de las GPUs, muchos investigadores y desarrolladores se han interesado por el gran poder de computación para la resolución de problemas de de propósito general.

En este trabajo se ha optado por la implementación del Filtro Fuzzy PGA en GPU. La mejora en rendimiento temporal del filtro puede facilitar su uso en otros ámbitos como en postproceso de imagen de CT o en filtrado de vídeo en tiempo real (por ejemplo para transmisión en *streaming*).

1.2. Objetivos

El objetivo principal de este trabajo es la correcta implementación y adecuación de un filtro para imagen de tipo radiográfico. En concreto para la eliminación de ruido impulsivo y gaussiano. Dicho filtro debe ser eficaz, i.e. debe conseguir una calidad de filtrado adecuada, estable, i.e. debe comportarse bien sea cual sea la morfología de lo radiografiado, por último, y debe ser temporalmente eficiente. A continuación se detallan los objetivos concretos:

- Implementar una técnica de filtrado que haya dado buenos resultados en estudios previos.
- Adaptar la técnica implementada para su uso con imágenes mamográficas mediante la obtención de los parámetros óptimos.
- Analizar la eficacia de la técnica comparándola con otras técnicas de la bibliografía empleando medidas de calidad relevantes.

- Asegurar la estabilidad del filtro ante diferencias morfológicas de los mamogramas.
- Paralelizar la técnica de filtrado para su ejecución en GPU de manera que mejore su rendimiento temporal.

1.3. Organización

Este trabajo se estructura según la consecución de los objetivos planteados. En el capítulo 2 se definen los fundamentos del ruido en imagen digital, filtrado y métricas para la medida de calidad del proceso de filtrado. También en este capítulo se ven los antecedentes (*estado del arte*) relacionados con este trabajo.

En el capítulo 3 se presentan los materiales utilizados en este trabajo que son, el filtro implementado, Fuzzy PGA y la base de datos de imágenes mamográficas, mini-MIAS.

El capítulo 4 trata el experimento multiparamétrico realizado para la obtención de los parámetros óptimos de entrada. La correcta configuración de estos parámetros asegura que el filtro Fuzzy PGA se adapte al tipo de imagen deseado, en nuestro caso el mamograma.

El análisis del filtro Fuzzy PGA en términos de eficacia, utilizando las métricas de calidad descritas, y la comparación con otros métodos se exponen en el capítulo 5.

El capítulo 6 aborda el estudio de estabilidad del filtro, lo que significa examinar si el filtro altera su comportamiento según las características morfológicas encontradas en el mamograma. Este estudio se realiza mediante el análisis de la distribución de los resultados.

Una vez comprobado que el filtro es adecuado para su uso con imágenes mamográficas se mejora su rendimiento temporal mediante la paralelización del método para su ejecución en GPU, esto se muestra en el capítulo 7.

El capítulo 8 muestra las conclusiones finales del trabajo y, por último, el capítulo 9 las perspectivas de trabajo futuro.

Capítulo 2

Fundamentos y Antecedentes

En esta sección se presentan los fundamentos y los trabajos más relevantes en cuanto a filtrado de imágenes en general según tipo de ruido, filtrado de imágenes en el campo de la medicina para la mejora del diagnóstico y para la reducción de la dosis radiactiva absorbida, y la implementación de estos filtros de forma eficiente mediante paralelización en GPUs.

2.1. Filtrado de Imágenes

El problema clásico de filtrado de imagen, o reducción de ruido, consiste en eliminar el ruido de una imagen tomada, de manera que obtengamos la imagen ideal como solución. Es un problema básico de inversión, el cual muchos autores han abordado desde múltiples puntos de vista diferentes. Los factores que pueden afectar a una imagen son múltiples, y se presentan en forma de tipos de ruido distintos. En este estudio se abordan el ruido gaussiano y el ruido impulsivo.

Para cuantificar la actuación de los filtros se utilizan varias medidas de calidad que se extraen de la comparación entre dos imágenes. Las que se utilizarán en este trabajo son: el error medio absoluto (*mean absolute error*, MAE), error cuadrático medio (*mean square error*, MSE), y el *Peak Signal-to-Noise Ratio*(PSNR).

2.1.1. Medidas de Calidad de Imagen

Las medidas de calidad de imagen son medidas objetivas de la variación entre dos imágenes, se utilizan para, por ejemplo, distinguir matemáticamente la calidad de una reconstrucción. Las medidas MAE y MSE indican la variación entre dos imágenes, por tanto

cuanto menor sean mejor es el filtrado. La métrica MSE proporciona una ponderación especialmente alta en errores muy dispares, por tanto es especialmente útil en el caso de que se desee evitar errores puntuales muy grandes. La métrica MAE sin embargo, pondera todos los errores por igual, de manera que es más utilizado para ver el comportamiento en valores continuos. La métrica PSNR se calcula a partir de MSE, y es indirectamente proporcional a ésta, de manera que cuanto mayor sea el valor de mejor calidad es el filtrado.

En resumen, PSNR se utiliza como medida de la reducción de ruido y MAE se utiliza para medir la preservación de la señal. Para definir el PSNR, es necesario calcular el MSE, que para dos imágenes en escala de grises, u , la imagen filtrada y I_0 , la imagen original se define como:

$$MSE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} \|I_0(i, j) - u(i, j)\|^2 \quad (2.1)$$

donde $M \times N$ es el tamaño de la imagen.

La medida PSNR se define:

$$PSNR = 10 \log_{10} \left(\frac{MAX_{I_0}^2}{MSE} \right) \quad (2.2)$$

donde MAX_{I_0} es el máximo valor posible de un píxel.

Y por último, MAE viene dado por:

$$MAE = \frac{1}{MN} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |I_0(i, j) - u(i, j)| \quad (2.3)$$

2.1.2. Filtrado de Ruido Gaussiano

El ruido gaussiano es el también llamado ruido normal, ya que se distribuye por toda la imagen siguiendo una distribución normal. La función de probabilidad de densidad p para una variable aleatoria z está dada por:

$$p_G(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(z - \mu)^2}{2\sigma^2}} \quad (2.4)$$

donde z representa el nivel de gris (intensidad de luz/color), μ el valor medio y σ la desviación estándar.

El principal momento de introducción de ruido gaussiano es en el proceso de adquisición de la imagen. Existen diversos factores que pueden provocar este ruido, se puede deber

a fallas en el sistema de adquisición, ya sea en el sensor de una cámara, en un escáner..., puede ser debido a las condiciones del entorno, por ejemplo una mala iluminación o altas temperaturas, o incluso a ruido provocado por los propios circuitos electrónicos. Como consecuencia la imagen puede mostrar un efecto bruma que difumina muchos píxeles.

Para eliminar el ruido gaussiano se emplean métodos de filtrado en el dominio del espacio, trabajando directamente con los píxeles de la imagen. Uno de los filtros de imágenes clásicos es el suavizado gaussiano lineal, basado en las funciones de Gabor tal y como indican en [5]. El ruido gaussiano se suaviza mediante un mediado de los valores vecinos, y depende de que este conjunto de vecinos sea lo suficientemente grande. También se puede emplear el filtrado en el dominio de la frecuencia, donde las operaciones se llevan a cabo en la transformada de Fourier de la imagen [6], métodos basados en la resolución de problemas de mínimos cuadrados [7], y métodos basados en el uso de ecuaciones de difusión no lineal [8–16].

Sin embargo la línea entre el filtrado y destrucción de la imagen es muy fina. El modelo anisotrópico de filtrado [17, 18] reduce también el ruido gaussiano, al igual que en el caso anterior, pero éste preserva mejor los detalles y los bordes.

La técnica de variación total para la reducción de ruido en imágenes [9, 19] persigue minimizar el coste de la restauración mediante una función que establece los gradientes para cualquier punto de su dominio. La aplicación de esta técnica en imágenes supone evitar oscilaciones provocadas por el ruido y preservar la información de los bordes de los objetos de la imagen, sin embargo es posible que algunos detalles de la imagen se pierdan si el multiplicador de Lagrange (que controla la cantidad de filtrado) es demasiado pequeño. Distintos estudios se han realizado para tratar de corregir este problema [20, 21].

Las anteriores técnicas se basan en el uso de vecinos que están próximos en el espacio, la técnica de filtrado de Yaroslavsky [22–24], sin embargo, basa la relación de vecindad en los valores de niveles de gris que poseen los píxeles. De este modo, el valor filtrado de un píxel i será una media de los valores de gris parecidos al que posee el píxel original. Se trata de un método no local, ya que utiliza píxeles de toda la imagen.

El filtro empírico local Wiener [23, 25] es un filtro que actúa en el dominio de la frecuencia. En este caso se trata de estimar el valor de la señal en el píxel central de una ventana que se mueve a lo largo de la imagen. Este filtro puede incurrir en un problema, la aparición del patrón *chessboard* (tablero de ajedrez).

El uso de filtros lineales no es eficiente para funciones con discontinuidades, para lo cual se estudian métodos basados en lo que se denomina en inglés *wavelet thresholding* [26]. Este procedimiento se basa en la idea de que la imagen viene representada mediante

coeficientes de *wavelets* grandes, los que serán mantenidos, y por coeficientes pequeños que pertenecen al ruido, que se cancelarán. Sin embargo, en los bordes existirán muchos coeficientes muy pequeños, que obtendrán como resultado una imagen con oscilaciones cercanas a los bordes. Una mejora de este método se presenta en [27].

Una aproximación diferente sería el uso de métodos estadísticos para el filtrado de imágenes [28]. Este método se basa en la comparación de subventanas para estimar el valor de restauración.

Las técnicas más recientes resaltan el buen funcionamiento de las ecuaciones de difusión no lineal [29–31]. Son métodos que eliminan el ruido mientras preservan los bordes de la imagen.

2.1.3. Filtrado de Ruido Impulsivo

El ruido impulsivo, a diferencia del ruido gaussiano, no está asociado al proceso de adquisición de la imagen. En este tipo de ruido, los píxeles que son alterados son píxeles aleatorios de la imagen, este ruido no sigue una distribución normal. Generalmente toman valores muy diferentes a los valores de los píxeles vecino, toman los valores extremos (puntos muy luminosos, o muy oscuros). Este ruido es debido a errores de transmisión, o corrupción al guardado o conversión de la imagen.

En presencia de ruido impulsivo, los filtros lineales, expresados como convoluciones a partir de la matriz de la imagen, pueden producir desenfoque en la imagen, pérdida de características locales y un filtrado pobre [32]. Para superar estas limitaciones se proponen los filtros no lineales.

Los filtros no lineales [33] pueden ser muy efectivos a la hora de eliminar ruido impulsivo. Un ejemplo de este tipo de filtros son los filtros de orden. Los filtros de orden están basados en los estadísticos de orden [34], una de las herramientas fundamentales de la estadística no paramétrica. Estos filtros utilizan una ventana o *peer group* de un ancho definido de tamaño pequeño para reemplazar el píxel central por un valor acorde con una regla predefinida para esa ventana. Por ejemplo, el filtro de la mediana [35] selecciona el valor que se sitúa en el medio de la lista ordenada de valores de la ventana. Esta técnica tiene ciertas limitaciones ya que no tiene en cuenta las características locales de la imagen, actúa igual en todos los lugares de la imagen.

Son muchas las mejoras que buscan mejorar la técnica de la mediana mediante una parametrización del ruido mediante un paso previo de detección del ruido, por ejemplo, el filtro de la mediana ponderado (*weighted median filter* [36]), el filtro adaptativo de media truncada [37], el filtro de pesado central de la mediana [38], el filtro basado en

cambio de la mediana [39], el filtro máscara de la mediana [40] y el método de mínimo-máximo [41].

Basados también en el concepto de *peer group* han surgido nuevas aproximaciones en el que éste consiste en un conjunto difuso de entre todos los vecinos, seleccionando aquellos que se consideran más relevantes, llamado ahora *fuzzy peer group*, estos métodos han obtenido buenos resultados en imágenes a color [42–44]. Ejemplos de dichas técnicas son: Peer Group Fuzzy Metric (PGFM) y el método de Difusión No Lineal Fuzzy (FNLDF), analizados en [45–48].

Las técnicas más recientes, como la última citada, buscan eliminar tanto ruido impulsivo como gaussiano. Además, las técnicas más novedosas utilizan detectores de tipo *fuzzy*, basados en diferencias de rangos ordenados (ROD) y en las diferencias estadísticas [49]. De nuevo basándose en los valores que se encuentran en una ventana, también denominada *peer group*, podemos refinar este grupo mediante métricas *fuzzy* a fin de crear el *fuzzy peer group* [4] que no contendrá necesariamente todos los valores vecinos, si no sólo los interesantes para el píxel analizado como en el caso de la técnica Fuzzy PGA que implementamos en este trabajo. Otro estudio basado en técnicas *fuzzy* se dedicó al filtrado de ruido impulsivo en vídeo [50].

2.2. Filtrado de Imágenes Médicas

La necesidad del uso de filtros se hace aún más patente en el área de la imagen médica, ya que en el caso de la imagen médica, el ruido puede dificultar el diagnóstico. Generalmente se utilizarán filtros que ayuden a reducir tanto ruido gaussiano como ruido impulsivo. El filtro de difusión no lineal se ha aplicado con éxito en imágenes de rayos X [51]. Es especialmente importante que los filtros utilizados eliminen el ruido y a la vez preserven los bordes, de manera que se mantengan todas las características locales de la imagen, ya que pueden ser relevantes para el diagnóstico. También se han aplicado las últimas tendencias sobre el uso de métricas *fuzzy* [45, 52]. En un estudio sobre imagen CT se utiliza el filtro de difusión híbrida anisotrópica [53]. En otro estudio sobre imagen MRI (imagen de resonancia magnética) se utiliza un filtro basado en *wavelet thresholding* [54].

Un área de estudio dentro del filtrado de imágenes médicas es el filtrado para la reducción de la dosis radiactiva absorbida. El objetivo es reducir la dosis de radiación a la cual el paciente se ve sometido. La idea de utilizar filtros se basa en que es posible mejorar la calidad de la imagen sin necesidad de incrementar la dosis radiactiva [55, 56]. En imágenes CT es especialmente importante reducir la dosis debido al gran número de planos

radiados. En un estudio se presenta la posibilidad de combinar la reconstrucción iterativa de las imágenes CT con un algoritmo de filtrado basado en la adaptación estadística [57], la misma técnica se utiliza para estimar la radiación de la angiografía coronaria CT [58, 59]. Un análisis sobre imágenes de radiografía computarizada pediátrica valida estadísticamente los modelos de adición de ruido para el estudio de la reducción de dosis radiactiva [60].

2.3. Paralelización de Filtrado de Imagen mediante GPU

Las unidades de procesamiento gráfico (GPU) fueron diseñadas originalmente para computación paralela y eficiente de gráficos, como en el caso de rasterización de imágenes, o efectos de luz. Sin embargo, a medida que las GPUs han ganado en potencia, gracias a la unificación de cientos de procesadores, las GPUs se han comenzado a utilizar para resolver problemas complejos de propósito general. Los primeros intentos de uso de GPUs para tareas no relacionadas con los gráficos se realizaron convirtiendo el problema a primitivas gráficas e interpretando los resultados gráficos. Actualmente existen *toolkits* tanto para nVidia [61] como para ATI [62] que permiten a los desarrolladores escribir sus funciones, llamadas *kernels*, en lenguaje C (incluyendo unas extensiones del propio *toolkit*), y éstas se ejecutarán en paralelo en las distintas unidades de proceso. Esto unido al abaratamiento del *hardware* ha supuesto un incremento muy importante de interés de la comunidad científica y un desarrollo considerable de la programación GPGPU.

Así mismo, el filtrado de imágenes también puede hacer uso de esta tecnología para conseguir un filtrado más eficiente. A menudo el filtrado de imagen se realiza sobre secuencias de vídeo, por tanto necesita que se ejecute en poco tiempo [63], por ejemplo para reducir el ruido impulsivo en vídeo de imagen multicanal utilizando *frames* consecutivos para afinar el filtrado [64], que además utiliza métricas *fuzzy*. Otro trabajo presenta el filtrado como un paso de pre-procesamiento en la codificación de vídeo, y de postprocesamiento en la decodificación [65]. Por otra parte, es posible incrementar la sofisticación de algunos algoritmos y que se ejecuten en un tiempo razonable [66].

Algunos trabajos se han centrado en el filtrado de una imagen, intentando maximizar el uso de la GPU [44, 67] con el objetivo de encontrar una solución de compromiso entre calidad y tiempo computacional.

Capítulo 3

Materiales

Esta sección describe la técnica de filtrado Fuzzy PGA que es el método que queremos examinar en detalle y comprobar que es eficaz, estable y eficiente. También se detalla la base de datos de imágenes mamográficas utilizadas en el trabajo, la base de datos mini-MIAS.

3.1. Filtro Fuzzy Peer Group Averaging (Fuzzy PGA)

Este filtro actúa en dos pasos, el primer paso consiste en la detección y reducción del ruido impulsivo, el segundo paso corresponderá al suavizado del ruido gaussiano. Ambos pasos se basan en la técnica *fuzzy peer group*, esto es un grupo de pares o píxeles vecinos lo suficientemente relevantes al píxel central x_i (el analizado) extraídos de una ventana W de $n \times n$ mediante una métrica *fuzzy*.

La definición del grupo *fuzzy* se basa en la ordenación de los vecinos del píxel central en base a su similitud con el píxel central x_0 .

Sea ρ una medida de similitud apropiada entre dos vectores. Los vectores $x_i \in W$ se ordenan en orden descendiente con respecto a la similitud con el píxel central x_0 , como resultado se obtiene el conjunto $W' = \{x_{(0)}, x_{(1)}, \dots, x_{(n^2-1)}\}$ de modo que $\rho(x_0, x_{(0)}) \geq \rho(x_0, x_{(1)}) \dots \geq \rho(x_0, x_{(n^2-1)})$, donde $x_0 = x_{(0)}$. Entonces el grupo $\rho_m^{x_0}$ de $m+1$ miembros asociado con el píxel x_0 es el conjunto

$$\rho_m^{x_0} = \{x_{(0)}, x_{(1)}, \dots, x_{(m)}\} \quad (3.1)$$

En [4], se propone un método basado en lógica difusa (*fuzzy*) para determinar el mejor número de miembros de un grupo *fuzzy* \hat{m} . El grupo *fuzzy* del píxel central x_0 en una ventana W se define como el conjunto *fuzzy* $F\rho_m^{x_0}$ extraído del conjunto $\{x_{(0)}, x_{(1)}, \dots, x_{(\hat{m})}\}$ dada la función de pertenencia $F\rho_m^{x_0} = \rho(x_0, x_{(i)})$. Entonces, el mejor número \hat{m} de miembros de $\rho_m^{x_0}$ se define como el valor $m \in N_W = 1, 2, \dots, n^2 - 1$ que maximiza la certeza de la siguiente regla:

Regla Fuzzy 1: Determinación de la certeza de que m es el mejor número de miembros para $\rho_m^{x_0}$

IF « x_m es similar a x_0 » y la similitud acumulada para $x_{(m)}$ es grande *THEN* «la certeza de que m sea el mejor número de miembros posible es alta».

$C_{FR1}(m)$ denota la certeza de la Regla Fuzzy 1 para m . $C_{FR1}(m)$ se calcula para cada $m \in N_W$ y el valor que maximiza la certeza se selecciona como el mejor número \hat{m} de miembros de $\rho_m^{x_0}$, esto es, $\hat{m} = \operatorname{argmax}_{m \in N_W} C_{FR1}(m)$.

La certeza de que « x_m es similar a x_0 » está dada por la función de pertenencia C^{x_0} determinada por la medida de similitud:

$$C^{x_0}(x_{(i)}) = \rho(x_0, x_{(i)}), i = 0, 1, \dots, n^2 - 1 \quad (3.2)$$

La similitud acumulada para x_m denotada $A^{x_0}(x_{(m)})$ se define como:

$$A^{x_0}(x_{(i)}) = \sum_{k=0}^{k=i} \rho(x_0, x_{(k)}), i = 0, 1, \dots, n^2 - 1 \quad (3.3)$$

Por lo tanto, la certeza de la expresión « $A^{x_0}(x_{(m)})$ es grande» está dada por la función de pertenencia L^{x_0} definida por:

$$L^{x_0}(x_{(i)}) = \frac{(A^{x_0}(x_{(i)}) - 1)(A^{x_0}(x_{(i)}) - 2n^2 + 1)}{(n^2 - 1)^2}, i = 0, 1, \dots, n^2 - 1 \quad (3.4)$$

La t-norma se utiliza como operador de conjunción, de manera que no es necesaria la *defuzzificación*. Entonces, $C_{FR1}(m) = C^{x_0}(x_{(m)})L^{x_0}(x_{(m)})$.

La función *fuzzy* de similitud, ρ , utilizada se define:

$$\rho(x_i, x_j) = e^{-\frac{\|x_i - x_j\|}{F\sigma}}, i, j = 0, 1, \dots, n^2 - 1 \quad (3.5)$$

donde $\|\cdot\|$ denota la norma euclídea y F_σ es uno de los parámetros de entrada del filtro. Esta función ρ toma valores en el intervalo $[0, 1]$ y satisface la ecuación $\rho(x_0, x_i) = 1$ si y solo si $x_0 = x_i$.

Para detectar el ruido impulsivo se utiliza la siguiente regla *fuzzy*:

Regla Fuzzy 2: Determinación de la certeza de que el píxel x_0 no posee ruido impulsivo *IF* «la similitud acumulada $A^{x_0}(x_{(\hat{m})})$ es grande» y « $x_{(\hat{m})}$ es similar a x_0 » *THEN* « x_0 no posee ruido impulsivo»

Para calcular la certeza de que la Regla Fuzzy 2, denotada por C_{FR2} , la certeza de « $A^{x_0}(x_{(\hat{m})})$ es grande» está dada por L^{x_0} , (definida en (3.4)) y la certeza de la expresión « $x_{(\hat{m})}$ es similar a x_0 » está dada por C^{x_0} definida en la fórmula (3.2). La t-norma se utiliza como el operador de conjunción y entonces $C_{FR2}(x_0) = C^{x_0}(x_{(\hat{m})})L^{x_0}(x_{(\hat{m})})$. Esta certeza ya está, en realidad, calculada, ya que $C_{FR2}(x_0) = C_{FR1}(\hat{m})$, por lo que no es necesario recalcularlo. *IF* la certeza de la Regla Fuzzy 2, C_{FR2} satisface

$$C_{FR2}(x_0) \geq F_t \quad (3.6)$$

THEN x_0 no posee ruido impulsivo *ELSE* x_0 es un píxel ruidoso y se reemplaza mediante la técnica VMF_{out} [35]. F_t es un umbral, indicado como parámetro de entrada del filtro, con valores entre $[0,1]$.

El segundo paso es la corrección del ruido gaussiano, que consiste en la aplicación de un difuminado sobre todos los píxeles de la imagen, con la diferencia de que en dicho difuminado sólo intervendrán los miembros del *fuzzy peer group*.

3.2. Base de Datos Mini-MIAS

En este trabajo se comparan los distintos métodos de filtrado anteriormente descritos sobre una base de datos de imágenes mamográficas, la base de datos mini-MIAS [68]. Esta base de datos es una colección normalizada (imágenes en blanco y negro con bandas negras a los lados de modo que complete el tamaño de la imagen hasta los 1024 píxeles \times 1024 píxeles) de 322 mamogramas, 161 mamogramas de mama izquierda y 161 de mama derecha.

Cada una de las imágenes incluye información acerca de las características del tejido dominante (graso, graso-glandular o glandular denso) y de las características de la anomalía presente: a) calcificación, b) masa bien definida o circunscrita, c) masa espigada,

d) otras, masas mal definidas, e) distorsión estructural, f) asimetría, o g) normal. En caso de presencia de una masa también se indica el diagnóstico de ésta (benigna o maligna) y su localización en forma de coordenadas (x, y) .

A fin de acomodar los nuevo experimentos al anteriormente realizado en [45], el mamograma considerado se recorta a 512×960 píxeles, eliminando las bandas negras laterales (que no contienen información valiosa). Esta imagen será nuestra imagen ideal, y con ella se obtiene una colección de imágenes contaminadas con ruido. A la primera se le añade ruido impulsivo con una densidad de $D = 0,10$. A la segunda imagen se le añade ruido gaussiano ($\sigma = 0,01$). Y la tercera imagen contendrá la mezcla de ruido impulsivo y gaussiano. Así pues, por cada imagen ideal se obtienen 3 contaminadas con ruido.

Capítulo 4

Obtención de los Parámetros Óptimos de Entrada para Fuzzy PGA

Este capítulo presenta el primero de los experimentos realizados en este estudio. Se trata de un experimento multiparamétrico para obtener los parámetros óptimos de entrada para el filtro Fuzzy PGA cuando es utilizado en imágenes mamográficas. Las imágenes se obtienen de la base de datos mini-MIAS descrita anteriormente. Después de la descripción del experimento se presentan los resultados y las conclusiones obtenidas.

4.1. Entorno de Pruebas

El sistema utilizado para realizar la experimentación es un PC con un sistema operativo Linux de 64 bits, con 16GB de RAM y un procesador i5-3570 a 3,40 GHz. Los métodos de filtrado se implementaron utilizando MATLAB (licencia de servidor proporcionada por la Universitat Politècnica de València) y los experimentos fueron programados mediante *bash scripts*. Además, el experimento multiparamétrico hace uso del entorno grid, en concreto de la Organización Virtual (VO) Biomed.

Organización Virtual Biomed

La VO *biomed* es una VO multidisciplinar a gran escala internacional, apoya las distintas comunidades científicas dentro del sector de las ciencias de la vida. Esta VO está gestionada por la *European Grid Infrastructure* [69] (EGI).

EGI proporciona acceso a recursos de computación de altas prestaciones mediante tecnología grid [70]. La EGI conecta diversos centros de distintos países europeos y apoya un gran número de proyectos de investigación de diferentes disciplinas, como el análisis de imagen médica, proporcionando a científicos acceso a más de 370.000 CPUs lógicas y 170PB de capacidad de almacenamiento.

El *middleware* empleado para acceder a la infraestructura EGI es gLite [71]. gLite, que formaba parte de la *European Middleware Initiative*, es ahora un proyecto independiente y puede ser utilizado en cualquier infraestructura grid, como por ejemplo EGI.

En gLite, la gestión de trabajo se implementa mediante el Workload Management Server (WMS), que gestiona la carga del trabajo e interacciona con el servidor de programación de tareas. El WMS determina el estado de los recursos computacionales y del almacenamiento y selecciona un *Computer Element* donde ejecutar el trabajo.

4.2. Descripción del Experimento Multiparamétrico

El objetivo de este estudio es obtener los valores óptimos para F_t y F_σ , parámetros de entrada de la técnica Fuzzy PGA, de modo que se adecue a su uso en mamogramas. Estos parámetros son los encargados de definir cómo se comporta el filtro, y están relacionados con la cantidad de ruido presente en la imagen a filtrar, por tanto es importante establecerlos correctamente.

El experimento consiste en aplicar el filtro Fuzzy PGA sobre toda la colección de mamogramas contaminados (descritos anteriormente). Se prueban diversas combinaciones de los parámetros de entrada. La salida obtenida, es decir, la imagen filtrada se compara con la imagen original y se obtienen unas medidas de calidad. Estas medidas son: el error medio absoluto (*mean absolute error*, MAE), error cuadrático medio (*mean square error*, MSE), y el *Peak Signal-to-Noise Ratio*(PSNR).

Los valores para el parámetro F_t van de 0,05 a 0,90 en incrementos de 0,05 unidades, y los valores de F_σ van de 50 a 600 en incrementos de 10 unidades, esto significa ejecutar la tarea 1008 veces por imagen de la colección. Como podemos observar, el experimento es un problema multiparámetro típico, con un alto número de repeticiones, lo que lleva a plantearse la paralelización. En este caso una buena solución es el empleo del paradigma grid. La computación grid facilita la ejecución de procesos concurrentes y, ya que nuestro experimento no necesita sincronización de hilos, intercambio de mensajes, o particionado de datos, la ejecución de esta tarea en el grid es más que factible.

Existen sutiles diferencias entre la ejecución secuencial (figura 4.1) y la ejecución grid (figura 4.2). El flujo de ejecución de este experimento se describe en el siguiente esquema básico:

- El el paso de **preproceso** se preparan, en el recurso local, los datos de entrada, lo que significa: la creación de un fichero de trabajo desde una plantilla (*template*), y el envío de los ficheros necesarios (imagen a filtrar y ejecutable del filtro Fuzzy PGA).
- Durante la **ejecución remota**, se emplean los recursos del grid para ejecutar concurrentemente cuando sea posible un conjunto de tareas, cada una de ellas genera unos datos de salida.
- Finalmente, en el paso de **postproceso** se recogen los datos de salida para analizarlos posteriormente en el recurso local.

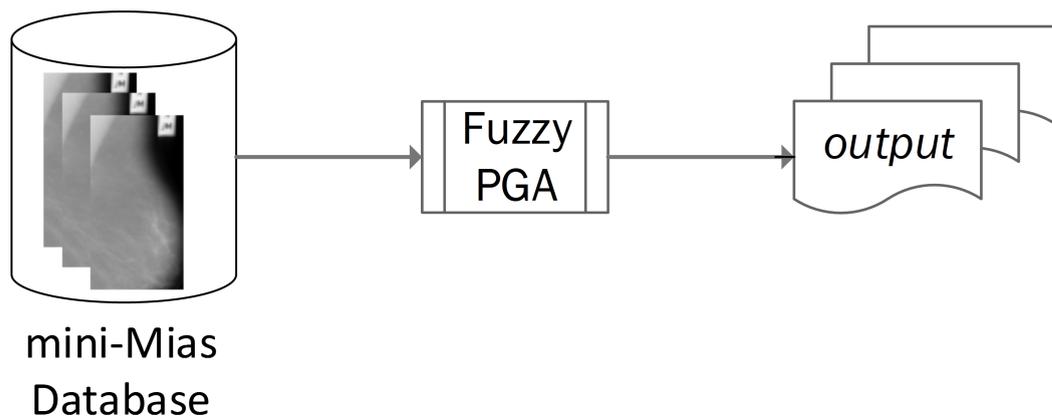


FIGURA 4.1: Flujo de la ejecución secuencial.

Entonces, según indica el esquema básico anterior, la fase de pre-proceso implica la preparación de los datos de entrada. En nuestro caso estos datos se componen de la imagen contaminada con ruido, la imagen original con la que se comparará una vez ejecutado el filtro, el ejecutable de nuestra implementación del filtro Fuzzy PGA y un *script* que genera una tarea interpretable por gLite. Estos ficheros de tareas se definen utilizando un lenguaje específico interpretable por gLite, el JDL (*Job Description Language*).

Una vez creada la colección de JDLs, podemos lanzar los trabajos mediante el siguiente comando gLite:

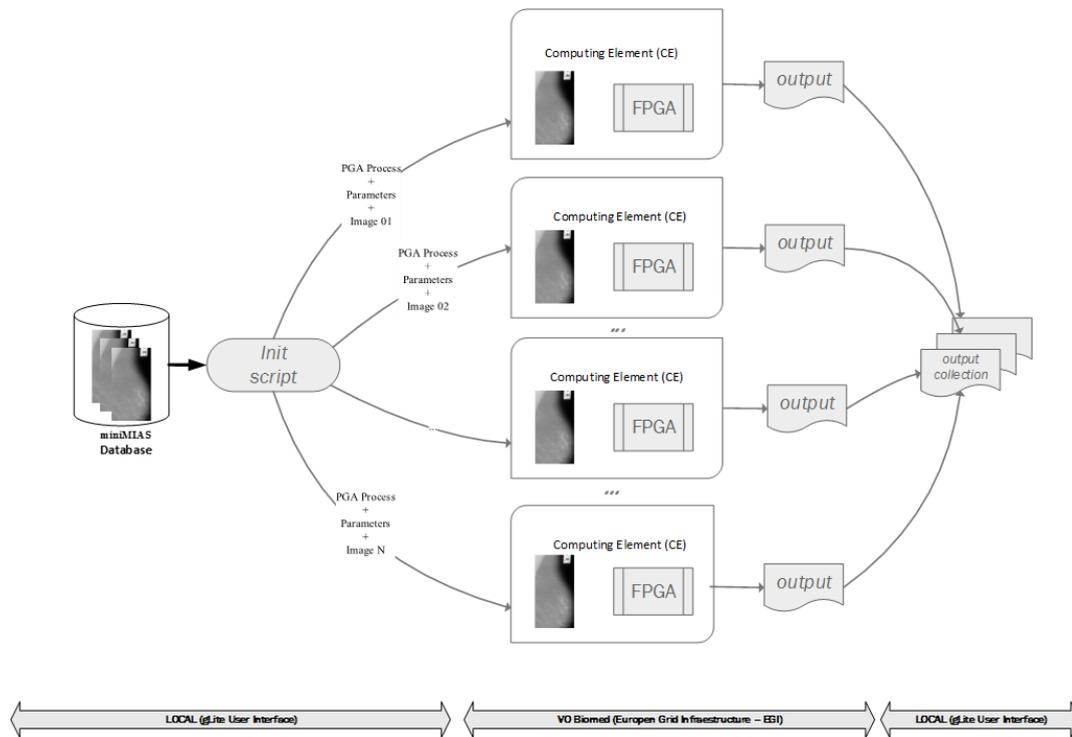


FIGURA 4.2: Flujo de la ejecución grid.

```
glite-wms-job-submit <delegation-opts> [options] <jdl_file>
```

Para semiautomatizar el proceso se ha implementado un *script* que abarca el lanzamiento de toda la colección de trabajos. Además, se han desarrollado otros *scripts* para la monitorización, cancelación y relanzamiento de trabajos (en caso de fallo) y obtención de los datos de salida.

4.3. Resultados

Los resultados del experimento multiparamétrico para obtener los parámetros óptimos de entrada F_t y F_σ para Fuzzy PGA, utilizando la imagen número 6 de la base de datos mini-MIAS, se muestran en las figuras 4.3 y 4.4. El resto de las imágenes de la base de datos que fueron analizadas siguen el mismo comportamiento en relación a los valores de F_t y F_σ . El mejor PSNR obtenido fue de 41,24 para los parámetros $F_t = 0,85$ y $F_\sigma = 480$. En un estudio anterior de este filtro [4] se dan unos valores para los parámetros de entrada distintos (para ruido de intensidad media $F_t = 0,15$ y $F_\sigma = 100$), las diferencias se deben a varios factores, como el tipo de imagen, que en nuestro estudio son mamogramas en blanco y negro, y en el estudio presentado por

Morillas se tratan de imágenes convencionales a color (3 canales), y por otra parte que la implementación es distinta, por ejemplo la función de similitud (o distancia) en el que está involucrado el parámetro F_σ depende del número de canales de la imagen.

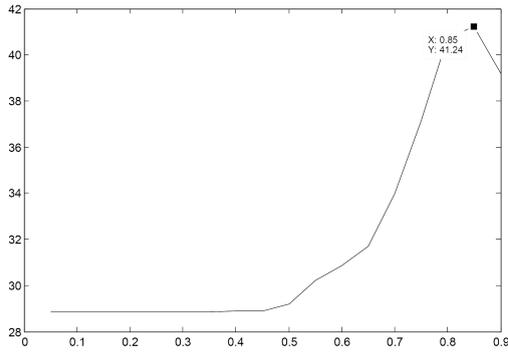


FIGURA 4.3: Valor PSNR del filtrado mediante Fuzzy PGA en función del F_t con un $F_\sigma = 480$ (imagen 6 de mini-MIAS con ruido impulsivo).

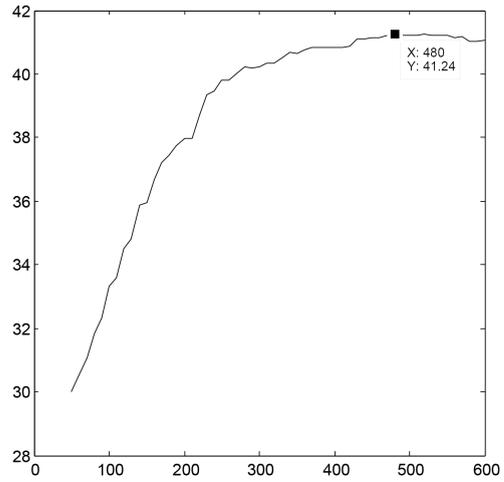


FIGURA 4.4: Valor PSNR del filtrado mediante Fuzzy PGA en función del F_σ con un $F_t = 0,85$ (imagen 6 de mini-MIAS con ruido impulsivo).

Para imágenes contaminadas con otros ruidos se realiza el mismo experimento, la tabla 4.1 muestra los valores óptimos para cada tipo de ruido.

TABLA 4.1: Valores óptimos para F_t y F_σ

	F_t	F_σ
Ruido Impulsivo ($D = 0,10$)	0.85	480
Ruido Gaussiano ($\sigma = 0,01$)	0.85	340
Ruido Impulsivo y Gaussiano	0.85	350

4.4. Conclusiones

El experimento multiparamétrico planteado se ha ejecutado correctamente utilizando infraestructura grid. Tras analizar de manera local los datos recogidos se establecen los parámetros de entrada óptimos para el filtro Fuzzy PGA en $F_t = 0,85$ y $F_\sigma = 480$. Cabe destacar que, como se extrae de los resultados, es posible que estos valores oscilen levemente sin alterar significativamente la eficacia del filtro.

La obtención de estos parámetros es crucial para el buen comportamiento del filtro, ya que, como se ha mencionado anteriormente, cualquier cambio en la implementación o en el tipo de imagen objetivo requiere de una nueva configuración del filtro.

Capítulo 5

Estudio Comparativo de Técnicas de Filtrado

Este capítulo abarca el estudio comparativo de técnicas de filtrado. La primera parte presenta las técnicas con las que se compara nuestra implementación del filtro Fuzzy PGA, son analizadas en detalle en [45]. La segunda parte describe el procedimiento del estudio, y por último se presentan los resultados y conclusiones obtenidas.

5.1. Técnicas de Filtrado

Las técnicas de filtrado utilizadas como punto de partida para la comprobación de que nuestra implementación del filtro Fuzzy PGA se comporta adecuadamente y es eficaz son: la técnica de filtrado *Peer Group Fuzzy Metric*, basada en métricas *fuzzy* y que se compone de dos fases, una de detección y otra de corrección, el filtro de Difusión no Lineal, basada en las ecuaciones de difusión no lineal y una técnica que combina los dos anteriores, filtrado por Difusión no Lineal Difusa (*fuzzy*).

5.1.1. Filtro Peer Group Fuzzy Metric (PGFM)

Este método también se ejecuta en dos fases. La primera fase trata de detectar píxeles erróneos (ruidosos) y la segunda intentará corregirlos. Durante la fase de detección se emplea una métrica *fuzzy* entre dos píxeles vecinos x_i y x_j tal como se describe en [42], dicha métrica está dada por la siguiente función:

$$M(x_i, x_j) = \frac{\min(x_i, x_j) + k}{\max(x_i, x_j) + k}, \quad k > 0 \quad (5.1)$$

El valor k reduce la no uniformidad o evita que los datos de dos pares consecutivos (en forma de vectores de distancias) pierdan valor significativo. La métrica *fuzzy* empleada para componer el conjunto de pares $P(x_i, d)$ es la indicada en (5.1), donde x_i es el píxel central de una ventana W de tamaño $n \times n$ (en este trabajo se utiliza en todo momento un tamaño $n = 3$) y $d \in [0, 1]$. La representación de $P(x_i, d)$ se expresa como sigue:

$$P(x_i, d) = \{x_j \in W : M(x_i, x_j) \geq d\} \quad (5.2)$$

El conjunto de pares [43] asociado con el píxel central x_i de W es el conjunto formado por el píxel central y los píxeles vecinos que forman parte de la ventana cuya distancia según una métrica *fuzzy* desde x_i es mayor que d .

La fase de detección se compone de dos pasos. El primer paso consiste en calcular el conjunto de pares para x_i en la ventana W , y todas las distancias del conjunto de pares con respecto a él. El píxel se declara como no corrupto si la cardinalidad de $P(x_i, d)$ es mayor que $(m + 1)$, donde m es el umbral definido. Si no, se etiqueta como no diagnosticado. En el segundo paso los píxeles etiquetados como no diagnosticados se analizan. Entonces, todos los píxeles que pertenecen al conjunto de pares se etiquetan como no corruptos si la cardinalidad de $P(x_i, d)$ es mayor que $(m + 1)$, y si no el píxel central se marca como corrupto.

Los parámetros k , d y m se determinan heurísticamente, sus valores dependen de la imagen analizada. Por ejemplo, el valor de d depende de la cantidad y tipo de ruido que posea la imagen.

Durante la fase de corrección, cada píxel x_i previamente etiquetado como corrupto se reemplaza mediante la técnica de la Media Aritmética [72] (Arithmetic Mean Filter), que consiste en realizar la media aritmética de los píxeles vecinos (del conjunto de pares en la ventana W) que fueron etiquetados como no corruptos.

5.1.2. Filtro de Difusión no Lineal (Non-linear Diffusive Filter, NDF)

Una clase de métodos de restauración de imagen se basa en el uso de ecuaciones de difusión no lineal [8–12, 73], que podrían obtenerse de la minimización de la función apropiada. La elección de una función particular dependerá del problema. En el área de la imagen médica estos métodos ya han sido empleados con éxito [13].

Considerando la siguiente función [14]:

$$J(u, \beta, \mu, \varepsilon) = \int_{\Omega} (\sqrt{\beta^2 + \|\vec{\nabla}u\|^2} + \frac{\mu}{2}(u - I_0)^2 + \frac{\varepsilon}{2}(\vec{\nabla}u)^2) dx \quad (5.3)$$

donde I_0 es la imagen observada (contaminada con ruido), u es la imagen filtrada, μ y ε son constantes, y Ω es una región convexa en R^2 que constituye el espacio que contiene a la superficie $u(x, y)$ que representa la imagen. El primer término de la función para $\beta = 1$ representa el área de la superficie que representa la imagen [73], el segundo término está relacionado con la distancia entre la imagen observada y la solución deseada u , y el tercer término controla la regularidad de la solución.

Para el método concreto utilizado en este trabajo se considera el problema de minimización siguiente [9, 73]:

$$\min_u J(u, \beta, \mu, \varepsilon) \quad \text{subject to} \quad \frac{\int_{\Omega} (u - I_0)^2 dx}{\int_{\Omega} dx} = \sigma^2 \quad (5.4)$$

por lo tanto, se busca la imagen u que minimice la función $J(u, \beta, \mu, \varepsilon)$ y presente una varianza con respecto a la imagen observada I_0 igual a σ^2 ; σ , la desviación estándar del ruido de la imagen a priori, es desconocida, pero es importante conocer su valor para minimizar la ecuación anterior 5.4. Para ello, se estima σ mediante el método propuesto en [16].

Para la discretización del tiempo, se emplea un esquema semiimplícito y para resolver las ecuaciones se emplea AOS (Additive Operator Splitting) [10, 14]. El criterio de parada de este método se propone en [15], y está basado en un criterio de correlación.

5.1.3. Filtro Fuzzy Non-linear Diffusion (FNLDF)

Esta técnica es la combinación de los métodos PGFM y NDF. La secuencia de aplicación de los métodos es como sigue: primero se ejecuta el filtro PGFM que se encarga de limpiar el ruido impulsivo y después se ejecuta NDF que mejorará el ruido gaussiano.

5.2. Descripción del Estudio Comparativo

El experimento realizado obtiene las medidas de calidad PSNR, MAE y MSE para cada uno de los filtros utilizados. Todos se aplican con sus parámetros óptimos y sobre la misma imagen contaminada. Se realiza el filtrado para los distintos tipos de ruido descritos (impulsivo, gaussiano y la mezcla de ambos). De este modo es posible realizar una comparación rigurosa, y examinar si el comportamiento del filtro Fuzzy PGA es el deseado.

De manera similar al estudio multiparamétrico presentado en el capítulo anterior, se realizó el estudio para encontrar los parámetros óptimos de las técnicas con las que

TABLA 5.1: Valores óptimos para m y d

	m	d
Ruido Impulsivo ($D = 0,10$)	5	0.85
Ruido Gaussiano ($\sigma = 0,01$)	8	0.92
Ruido Impulsivo y Gaussiano	8	0.92

comparamos nuestro filtro. En este caso los parámetros relevantes son d y m de la técnica PGFM. La tabla 5.1 muestra los mejores valores para cada tipo de ruido, ya que estos parámetros dependen de la cantidad de ruido introducido en la imagen, y de la naturaleza de éste. Para la métrica *fuzzy* de estas técnicas se necesita obtener el valor óptimo de k que define cómo de acusado será el filtrado en función las características de la imagen. Se obtuvo un $k = 1024$ mediante el procedimiento descrito en [73].

5.3. Resultados

TABLA 5.2: Resultados de calidad para una imagen con ruido impulsivo ($D = 0,10$)

	MSE	PSNR	MAE
Filtrado con PGFM	5.5365	40.6985	0.1793
Filtrado con NDF	36.4523	24.3934	9.4899
Filtrado con FNLDF	7.1813	39.5688	0.759
Filtrado con Fuzzy PGA	4.8948	41.2335	1.1214
Imagen ruidosa	1.90e+02	15.3459	12.821

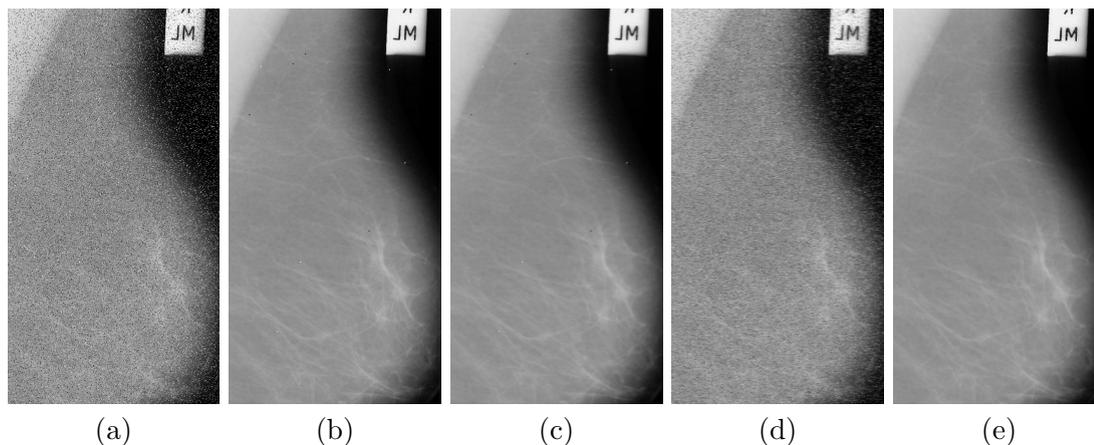


FIGURA 5.1: (a) Imagen con ruido impulsivo ($D=0.10$), (b) Filtrada (PGFM), (c) Filtrada (FNLDF), (d) Filtrada (NDF), (e) Filtrada (Fuzzy PGA)

Como se puede ver en la tabla 5.2 y figura 5.1, cuando la imagen contiene solamente ruido impulsivo, el mejor método es Fuzzy PGA. Los métodos PGFM y FNLDF también se comportan bastante bien, con poca diferencia con respecto a Fuzzy PGA, pero el método NDF no proporciona buenos resultados para esta imagen, como era de esperar, ya que

es un filtro diseñado para eliminar ruido gaussiano. En este caso llama la atención que el error absoluto para Fuzzy PGA, aunque no es muy elevado, sí es superior al obtenido por los métodos PGFM y FNLDF.

TABLA 5.3: Resultados de calidad para una imagen con ruido gaussiano ($\sigma = 0,01$)

	MSE	PSNR	MAE
Filtrado con PGFM	305.8954	23.2751	13.266
Filtrado con NDF	88.5022	28.6613	7.3277
Filtrado con FNLDF	99.7893	28.14	6.9826
Filtrado con Fuzzy PGA	68.9717	29.7441	6.4761
Imagen ruidosa	621.5723	20.1959	19.672

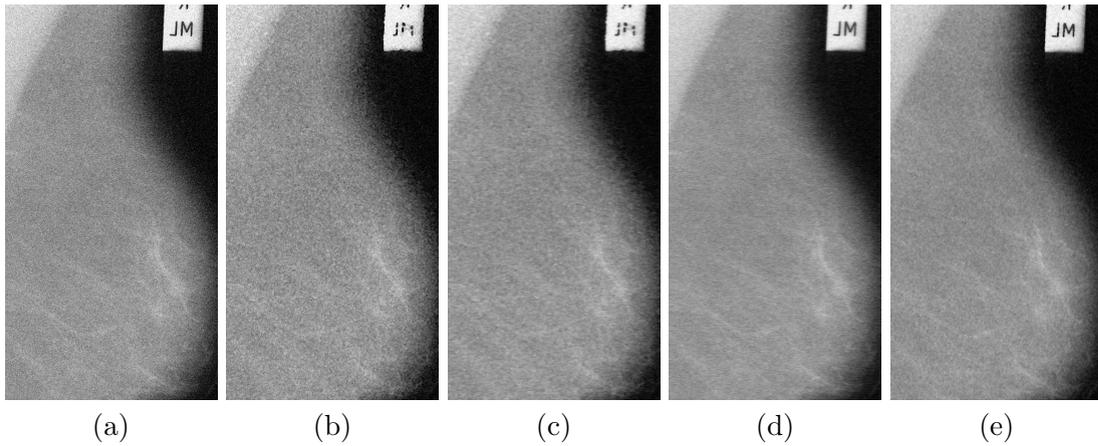


FIGURA 5.2: (a) Imagen con ruido gaussiano ($\sigma = 0,01$), (b) Filtrada (PGFM), (c) Filtrada (FNLDF), (d) Filtrada (NDF), (e) Filtrada (Fuzzy PGA)

En el caso de la imagen con ruido gaussiano, de nuevo Fuzzy PGA supera al resto de métodos en eficacia. NDF y FNLDF obtienen resultados similares, mejores que el método PGFM, diseñado para la eliminación de ruido impulsivo. La tabla 5.3 muestra los resultados, y la figura 5.2 muestra las imágenes resultantes.

TABLA 5.4: Resultados de calidad para una imagen con ruido impulsivo ($D = 0,10$) y gaussiano ($\sigma = 0,01$)

	MSE	PSNR	MAE
Filtrado con PGFM	322.9214	23.0398	13.6645
Filtrado con NDF	299.4993	23.3668	12.5675
Filtrado con FNLDF	103.1628	27.9956	7.3411
Filtrado con Fuzzy PGA	74.9976	29.3803	6.6121
Imagen ruidosa	2.23E+03	14.6383	29.6629

En el caso de la imagen contaminada tanto con ruido impulsivo como con ruido gaussiano (tabla 5.4 y figura 5.3), de nuevo Fuzzy PGA es mejor que el resto de técnicas. Resulta ligeramente mejor que FNLDF que obtiene a su vez una puntuación de PSNR superior en 4 puntos al resto de técnicas.

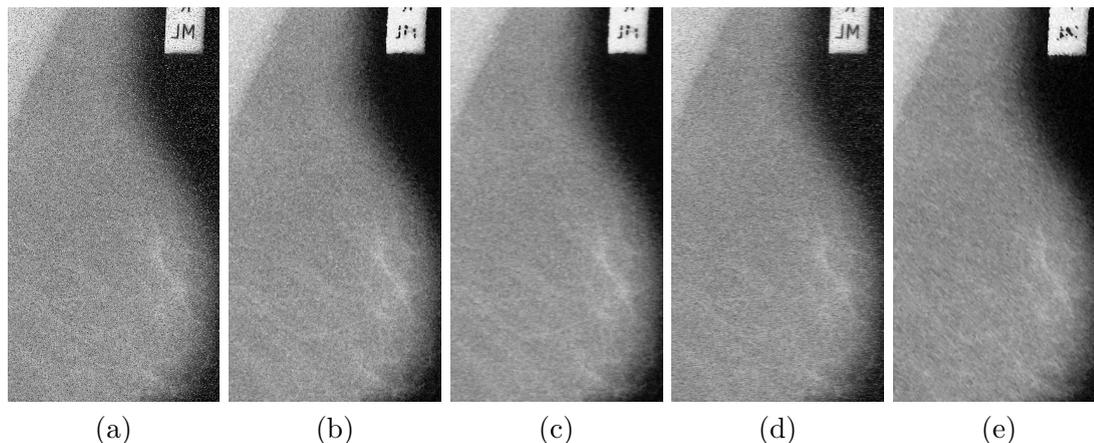


FIGURA 5.3: (a) Imagen con ruido impulsivo ($D = 0,10$) y gaussiano ($\sigma = 0,01$), (b) Filtrada (PGFM), (c) Filtrada (FNLDF), (d) Filtrada (NDF), (e) Filtrada (Fuzzy PGA)

5.4. Conclusiones

En este capítulo se ha presentado el estudio de eficacia de nuestra implementación del filtro Fuzzy PGA. Se ha comparado con los filtros FNLDF, PGFM y NLF anteriormente estudiados en otros trabajos utilizando imágenes contaminadas con ruido impulsivo, gaussiano y la mezcla de ambos.

El filtro Fuzzy PGA se ha comportado mejor que el resto de filtros según las puntuaciones de PSNR obtenidas para todos los casos. En el primero, sobre imágenes con ruido impulsivo, Fuzzy PGA obtiene los mejores resultados, aunque los métodos PGFM y FNLDF se comportan de manera similar. Sin embargo, en el caso de ruido gaussiano, PGFM no obtiene muy buena puntuación, y en el caso de la mezcla de ruidos, los resultados son similares.

En la comparación de la eficacia del filtrado de ruido impulsivo, se observa que el filtro Fuzzy PGA obtiene un mayor error de MAE que las técnicas de limpieza de ruido impulsivo, PGFM y FNLDF. Esto se puede deber a la acumulación de pequeños errores en el conjunto de la imagen. Sin embargo el MSE es menor en el caso de Fuzzy PGA que en el resto, lo que indica que no hay errores de valores dispares.

Esto nos lleva a considerar a Fuzzy PGA como un método de filtrado eficaz, ya que se comporta bien en los tres casos. Y podría ser un candidato a ser utilizado en filtrado de imagen CT para la reducción de dosis radiactiva.

Capítulo 6

Estudio de Estabilidad del Filtro Fuzzy PGA en la Base de Datos mini-MIAS

El objetivo de este estudio es determinar si el filtro Fuzzy PGA implementado es estable, es decir, se comporta correctamente sea cual sea el tipo de tejido dominante o de la anormalidad encontrada en el mamograma. Un filtro para imagen médica correcto no debe alterar la información que contiene la imagen, de modo que no influya en un posible diagnóstico, eso significa que no debe de existir correlación entre las medidas de calidad extraídas, y las características del mamograma analizado (tejido, anormalidad y diagnóstico).

Para averiguar si existe una correlación se aplica el filtrado Fuzzy PGA sobre cada una de las imágenes de la base de datos que han sido contaminadas con una combinación de ruido impulsivo y gaussiano. Para cuantificar la eficacia del filtrado se utilizará como medida de calidad el valor de PSNR (*Peak Signal to Noise Ratio*) para cada una de las reconstrucciones y se compararán los resultados obtenidos.

En la primera parte de este capítulo se describe el estudio realizado, la segunda parte muestra los resultados y finalmente, se presentan las conclusiones.

6.1. Descripción del Estudio de Estabilidad

Este experimento consiste en la aplicación del filtro Fuzzy PGA sobre nuestra colección de imágenes mamográficas extraída de la base de datos mini-MIAS. Como se ha indicado anteriormente, esta base de datos está compuesta por 322 mamogramas correspondientes

a 161 pacientes. Cada una de estas imágenes está clasificada según el tipo de tejido de la mama, el tipo de anomalía encontrada si existe y en el caso de que se haya encontrado una masa, si es benigna o maligna.

El objetivo del trabajo es establecer si existe o no correlación entre la eficacia del filtro y el diagnóstico asociado a la imagen filtrada. La eficacia del filtro se establecerá mediante la medida PSNR, que será mejor cuanto mayor sea.

Para este estudio se ha contado con 322 imágenes de la colección, en concreto con aquellas que se contaminaron con ruido impulsivo y gaussiano.

Una vez obtenida la colección se aplica el filtro sobre las imágenes contaminadas obteniendo una imagen filtrada 6.1 que se puede comparar con la original a fin de obtener nuestra medida de calidad (PSNR).

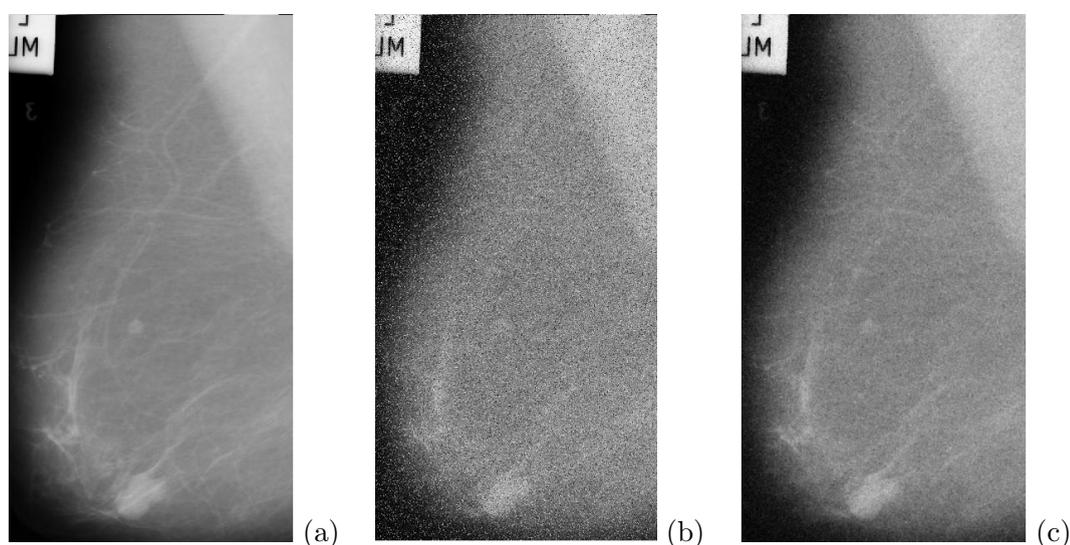


FIGURA 6.1: (a) Imagen original, (b) Imagen ruidosa (impulsivo, $D = 0,10$ y gaussiano, $\sigma = 0,01$), (c) Filtrada con Fuzzy PGA

Como salida de la función de filtrado obtenemos entre otras medidas el PSNR, que servirá en nuestra evaluación para comparar la eficacia del filtro en cada una de las imágenes. Por otra parte se realiza una carga de los datos de clasificación de las mamografías en una matriz numérica que indicará si posee o no la característica de entre las explicadas en el capítulo 3. En la tabla 6.1 se muestra una representación de estos datos para una parte de las imágenes.

6.2. Resultados

Para analizar el cruzado de datos entre los valores PSNR obtenidos en el filtrado de cada imagen y las características de cada imagen se utiliza el histograma de frecuencias. Este

TABLA 6.1: Ejemplo de datos cargados de la base de datos mini-MIAS

	F	G	D	NORM	CALC	CIRC	SPIC	MISC	ARCH	...
Imagen 1	No	Sí	No	No	No	Sí	No	No	No	
Imagen 2	No	Sí	No	No	No	Sí	No	No	No	
Imagen 3	No	No	Sí	Sí	No	No	No	No	No	
Imagen 4	No	No	Sí	Sí	No	No	No	No	No	
...										
Imagen 322	No	No	Sí	Sí	No	No	No	No	No	

diagrama representa con qué frecuencia una imagen ha obtenido un determinado PSNR (dividido en intervalos de una unidad). Esto facilita la interpretación de los datos, ya que muestra la variación en la eficacia del filtro, y es posible observar variaciones en función de una característica concreta de la imagen.

El primer histograma muestra los valores de PSNR obtenidos según el tipo de tejido subyacente. Tal y como se observa en la figura 6.2 los valores de PSNR siguen una distribución aproximadamente normal, siendo los valores para las imágenes de tejido graso ligeramente superiores.

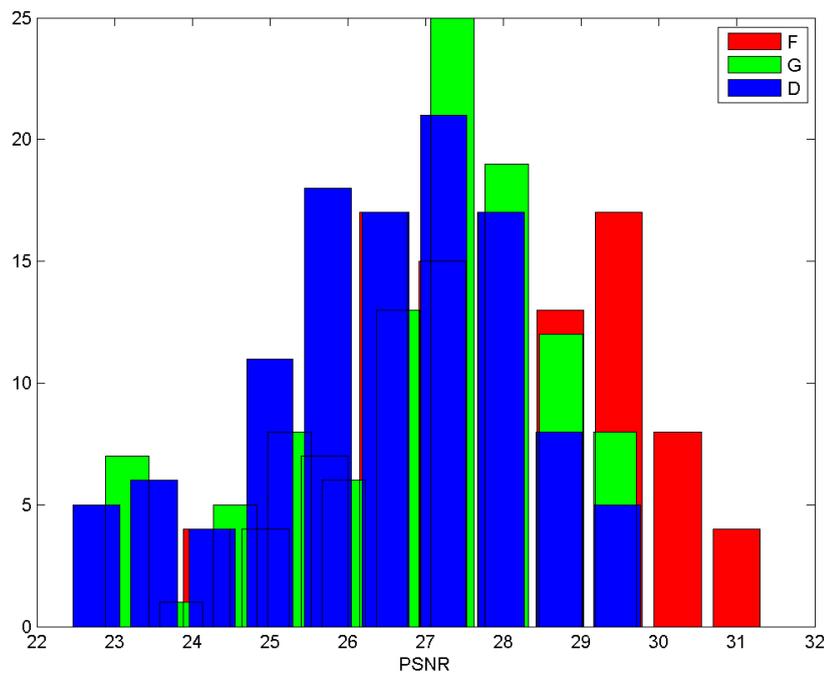


FIGURA 6.2: Histograma de frecuencia en función de los valores de PSNR entre imagen original y filtrada por clase de tejido: F (graso), G (graso-glandular), D (denso-glandular).

El segundo histograma (figura 6.3) muestra los valores de PSNR obtenidos según el tipo de anomalía. En este gráfico se observa una diferencia de altura acusada entre

las masas normales y el resto, debido a que no han sido normalizados los valores en función de la frecuencia de aparición, y por lo tanto, como es normal, existen muchos menos casos de mamogramas con anomalía que mamogramas normales. A pesar de las pocas muestras de casos con presencia de anomalía, sí que se puede distinguir que la mayoría de casos se agrupan en torno al centro, tendiendo a de nuevo a la distribución normal.

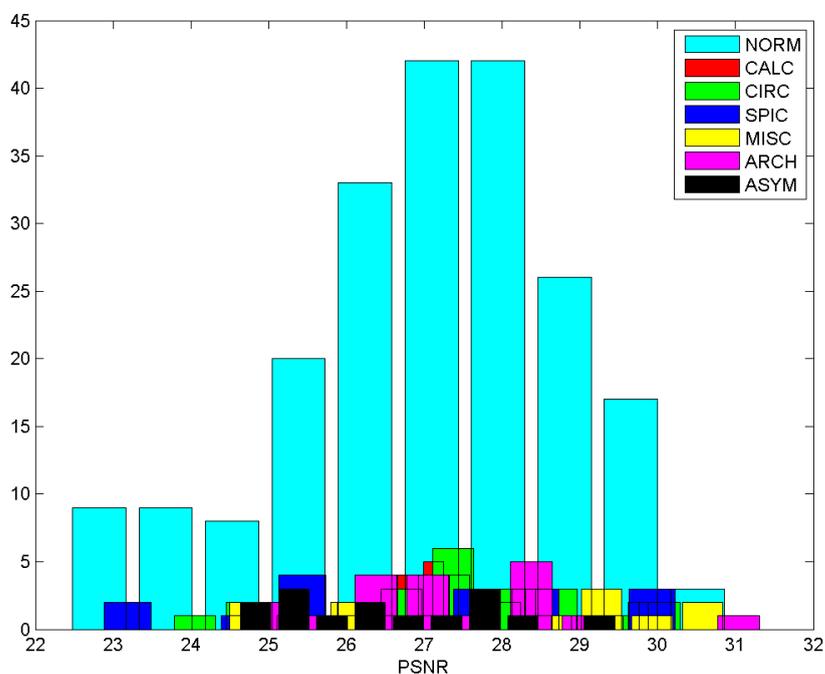


FIGURA 6.3: Histograma de frecuencia en función de los valores de PSNR entre imagen original y filtrada por clase de anomalía NORM (normal), CALC (calcificación), CIRC (masa bien definida/circunscrita), SPIC (masa espigada), MISC (otras, masas no definidas), ARCH (distorsión estructural), ASYM (asimetría).

Por último, el tercer histograma 6.4 relacionado con el diagnóstico de la masa encontrada, que muestra resultados similares a los anteriores. En todos los casos los valores de PSNR se distribuyen a lo largo de una serie de valores cercanos a 27, el valor central, una puntuación que es consistente con el experimento realizado en el capítulo 5.

6.3. Conclusiones

Es este estudio hemos presentado los resultados, en términos del PSNR obtenido, de filtrado de imágenes según tipo de tejido, anomalía y diagnóstico.

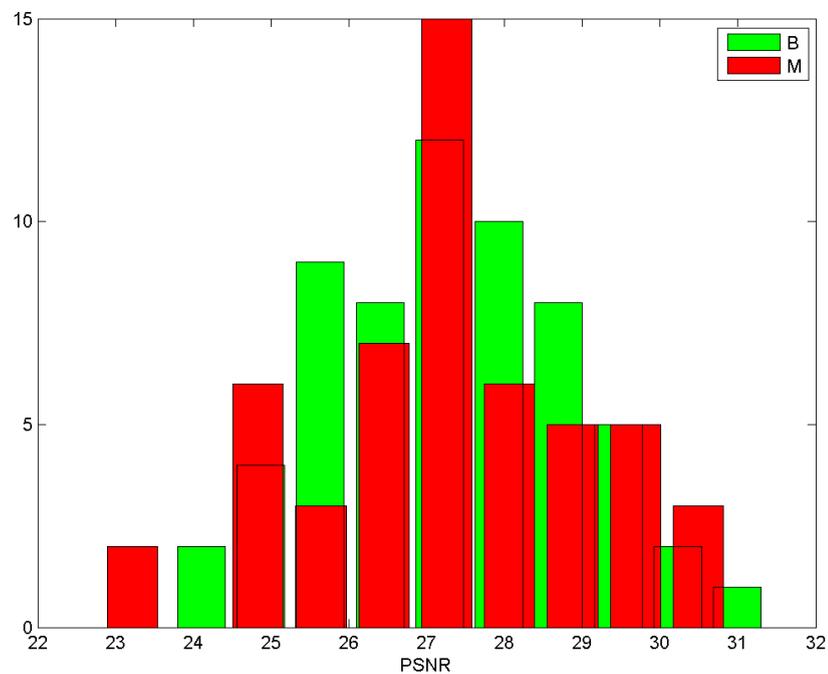


FIGURA 6.4: Histograma de frecuencia en función de los valores de PSNR entre imagen original y filtrada por tipo de diagnóstico: B (benigno), M (maligno).

Se ha demostrado que el filtro funciona correctamente ya que los valores del PSNR que comparan imagen original y filtrada son mejores que los valores para la imagen ruidosa y filtrada.

Se ha comprobado que no existe correlación entre el diagnóstico de la mamografía y el funcionamiento del filtro, incluyendo en este diagnóstico el tipo de tejido, el tipo de masa y la benignidad o malignidad de la masa, ya que las distintas distribuciones se centran en el mismo valor de PSNR.

Capítulo 7

Implementación Paralela en GPU del Filtro Fuzzy PGA

En este capítulo se presenta una implementación paralela utilizando la potencia de la GPU. El objetivo de esta parte del estudio es incrementar la eficiencia temporal del filtro Fuzzy PGA, de manera que pueda ser aplicado sobre un conjunto numeroso de imágenes, como por ejemplo en una CT, o incluso en tiempo real para el envío de vídeo vía *streaming*.

El hardware utilizado en las pruebas de ejecución de esta implementación del filtro ha sido una GPU especialmente diseñada para aplicaciones de análisis de datos y cálculo científico, el modelo Tesla k20. Esta tarjeta está basada en arquitectura Kepler, pensada especialmente para computación de altas prestaciones. Posee 2496 *cores* a una frecuencia de reloj de 706 MHz, y un tamaño de memoria de 5 GB y un ancho de banda de memoria de 208 GB/s. Las pruebas de la versión secuencial se han realizado en un PC con un sistema operativo Linux de 64 bits, de procesador Intel Core i7-3820 a 3,60 GHz.

La primera parte describe la implementación del filtro Fuzzy PGA para GPU, la segunda parte muestra las pruebas de rendimiento realizadas, y por último las conclusiones de esta parte del estudio.

7.1. Introducción a la computación en GPU

La idea de la computación paralela surge hace décadas buscando aumentar el rendimiento computaciones para la resolución de problemas grandes. No es desde hace muchos años que es posible utilizar procesadores multicore y unidades gráficas, y no exclusivamente de procesamiento gráfico, sino también para el propósito general.

En este trabajo se utiliza CUDA (Compute Unified Device Architecture) que hace referencia tanto a una plataforma de computación paralela como a un modelo de programación creado por NVIDIA, con el objetivo de resolver problemas grandes de una manera más eficiente que la CPU. CUDA se compone de una serie de librerías, una extensión de los lenguajes de programación C, C++ y Fortran y un compilador.

Un programa en CUDA nos permite ejecutar código en ambos procesadores con sólo un programa, de manera que podamos utilizar la potencia de la GPU. Una parte del programa se ejecutará en la CPU, que en CUDA se denomina *host*, y la otra parte en la GPU, o *device*, que será la parte paralela. El compilador se encargará de separar el programa para cada procesador y generar las instrucciones para cada uno.

CUDA asume que la GPU es un coprocesador de la CPU y que cada uno tiene su propia memoria para el almacenamiento de datos. La CPU es responsable de enviar los datos a la GPU y también de pedirle los resultados una vez haya finalizado su ejecución, esto lo hace con la directiva `cudaMemcpy`, análoga al `memcpy` de C. También se encarga de reservar la memoria en la GPU, mediante `cudaMalloc`. A continuación la CPU lanza el *kernel*, una función especial que ejecuta la GPU. Finalmente se copian los datos resultantes y se libera la memoria. La forma definitiva de un programa en CUDA se muestra en la figura 7.1.

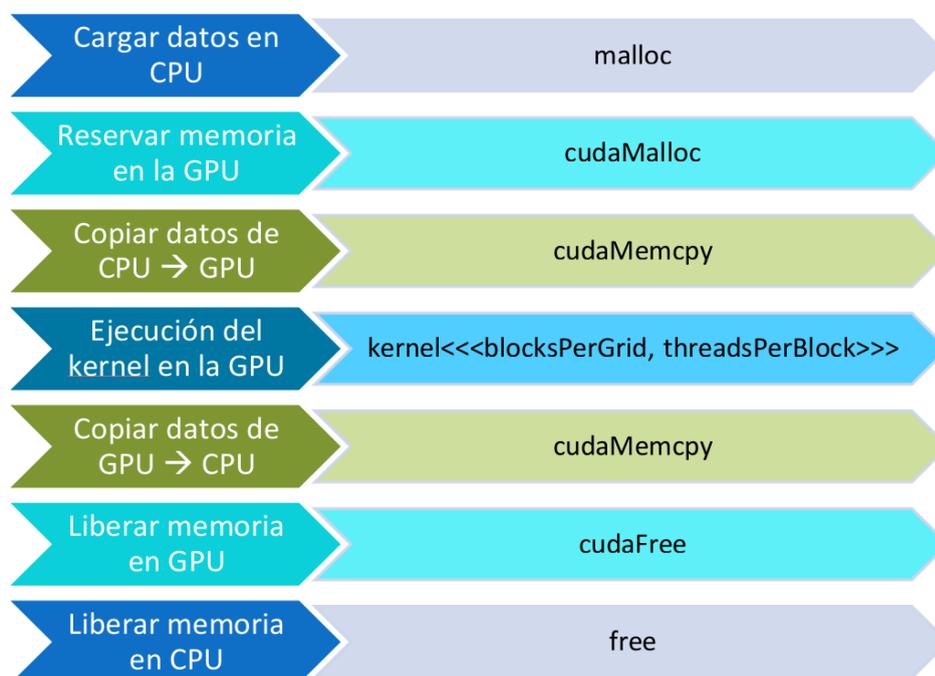


FIGURA 7.1: Modelo de programación de CUDA.

Como se ha mencionado anteriormente, un kernel es una función para la GPU, está escrita en C y es invocada de una manera especial de modo que sea ejecutada N veces en

paralelo por N hilos CUDA distintos. La llamada es de tipo `kernel«<blocksPerGrid, threadsPerBlock»`, el primer parámetro hace referencia al número de bloques que deseamos, y el segundo parámetro al número de hilos por bloque. Por ejemplo, si se desea ejecutar un programa de 1280 hilos en total se podría llamar al kernel mediante `kernel«<10, 128»` o bien `kernel«<5, 256»`, siempre teniendo en cuenta el número máximo de hilos disponible en la GPU, que es suele ser 1024.

De esta manera obtenemos una serie de bloques de hilos que colaboran para solucionar un problema. La GPU se encarga de distribuir los bloques en los llamados *streaming multiprocessors* (SMs), un SM puede ejecutar más de un bloque, pero un bloque no se puede distribuir entre varios SMs. CUDA garantiza que todos los hilos de un bloque se ejecutan en el mismo SM a la vez y que todos los bloques de un *kernel* (se le denomina *grid* al conjunto de bloques) terminan antes de que se ejecute cualquier otro bloque de otro *kernel*, es decir, contiene una barrera implícita.

Para poder crear programas más eficientes que manejen adecuadamente el acceso a memoria es necesario conocer la organización del hardware en la GPU. Según el modelo expuesto en la figura 7.2, podemos ver que dentro de la GPU cada hilo tiene su propia memoria local (la memoria más rápida), cada SM tiene su memoria compartida entre todos sus hilos y existe una memoria global accesible desde cualquier hilo de la GPU. Por otra parte tenemos la memoria de la CPU (la más lenta) desde la que se comunicarán los datos a la GPU.

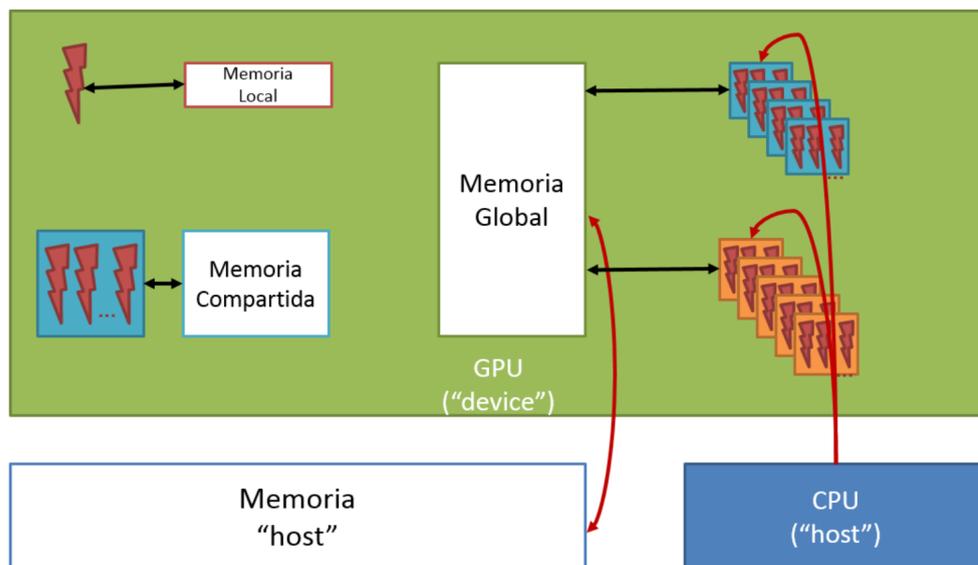


FIGURA 7.2: Modelo de memoria para la programación en CUDA.

El propósito de conocer el modelo de memoria es ser capaz de diseñar programas que hagan un buen uso de ella, de manera que se pueda optimizar el tiempo de acceso a

memoria más el tiempo de comunicación. Sabiendo esto en los programas muchas veces es necesario llegar a una solución de compromiso entre los dos tiempos citados.

7.2. Fuzzy PGA para GPU

Para la implementación del filtro Fuzzy PGA en GPU se utilizó el entorno de programación CUDA descrito anteriormente. En esta implementación el filtro Fuzzy PGA se divide en 3 fases, tras cada fase se sincronizarán los hilos ya que en las siguiente es necesario el conocimiento de los datos calculados en la anterior. Como la imagen es lo suficientemente pequeña para ser enviada a la memoria de la GPU, no se divide en bloques, de manera que el *kernel* se ejecuta en 960 *cores*, y cada *core* ejecuta 512 hilos. De este modo cada hilo se encarga de un solo píxel, así es posible ejecutar el cálculo de todos los píxeles al mismo tiempo. Los datos de la imagen son lo suficientemente pequeños para ser copiados en la memoria local de la GPU. El esquema de ejecución se muestra en el algoritmo 1.

Algoritmo 1 Fuzzy PGA para GPU

Entrada: Imagen ruidosa I_i , F_t , F_σ

Salida: Imagen filtrada I_o , MAE, MSE, PSNR

- 1: I_i CPU \rightarrow GPU
 - 2:
 - 3: **kernel Fuzzy PGA:**
 - 4: Marcado de píxeles ruidosos
 - 5: *sincronización de hilos*
 - 6: Filtrado de ruido impulsivo: *VMF*
 - 7: *sincronización de hilos*
 - 8: Filtrado de ruido gaussiano
 - 9:
 - 10: I_o CPU \leftarrow GPU
 - 11:
 - 12: Cálculo de medidas de calidad: MAE, MSE y PSNR
-

Como objetivo de mejorar todo lo posible el tiempo de ejecución del *kernel* se tienen en cuenta distintas técnicas:

- Minimización de trasiego de memoria entre CPU y GPU.
- Evitar tamaños de imagen / bloques con dimensiones que sean potencias de 2 por problemas de cache.
- Uso de `__ldg()` para el uso de memoria de texturas no explícito.

Para comprobar que esta implementación mejora el rendimiento del filtro, se compara con la ejecución de la implementación secuencial de Fuzzy PGA. El tiempo de ejecución incluirá en cualquier caso los movimientos de memoria necesarios.

7.3. Resultados

Para comparar las ejecuciones de la implementación secuencial y la implementación para GPU del filtro Fuzzy PGA se ha realizado el tiempo promedio de 10 ejecuciones consecutivas, variando ciertos ajustes el filtro: aplicando sólo el paso de limpieza de ruido impulsivo, aplicando sólo el paso de limpieza de ruido gaussiano y aplicando los dos. En la figura 7.3 podemos ver que que la versión en GPU es mucho más rápida, teniendo en cuenta que el *speedup* se define como:

$$S_P = \frac{T_{secuencial}}{T_{paralelo}} \quad (7.1)$$

Entonces se consigue un *speedup* de aproximadamente 12 unidades en el caso de que se apliquen los pasos de limpieza de ambos ruidos, de 8 en el caso de que se aplique sólo el de ruido impulsivo, y de 4 el caso de que se aplique sólo el de ruido gaussiano. Además, el tiempo observado en la ejecución del *kernel* que ejecuta el filtro en GPU es de 0,000016 ms.

Con respecto a las mejoras comentadas en la sección anterior, la mejora del trasiego de memoria implicó el uso de un solo *kernel* en lugar de los tres que se utilizaban en un primer momento. El primer diseño tenía tres *kernels* que se correspondían con cada uno de los pasos del filtro Fuzzy PGA: detección, limpieza de ruido impulsivo y limpieza de ruido gaussiano. Sin embargo esta configuración requería la comunicación de ciertos datos entre los distintos *kernels* así que se optó por unirlos en uno y agregar sincronización de hilos entre pasos.

La segunda técnica de mejora comentada, con respecto a los tamaños de imágenes, se probó mediante el uso de imágenes de distinto tamaño. Y la tercera mejora se incluyó en el código del *kernel*.

7.4. Conclusiones

La implementación del filtro Fuzzy PGA en GPU consigue una gran mejora de rendimiento en base a su versión secuencial. El bajo tiempo de ejecución del *kernel* implica

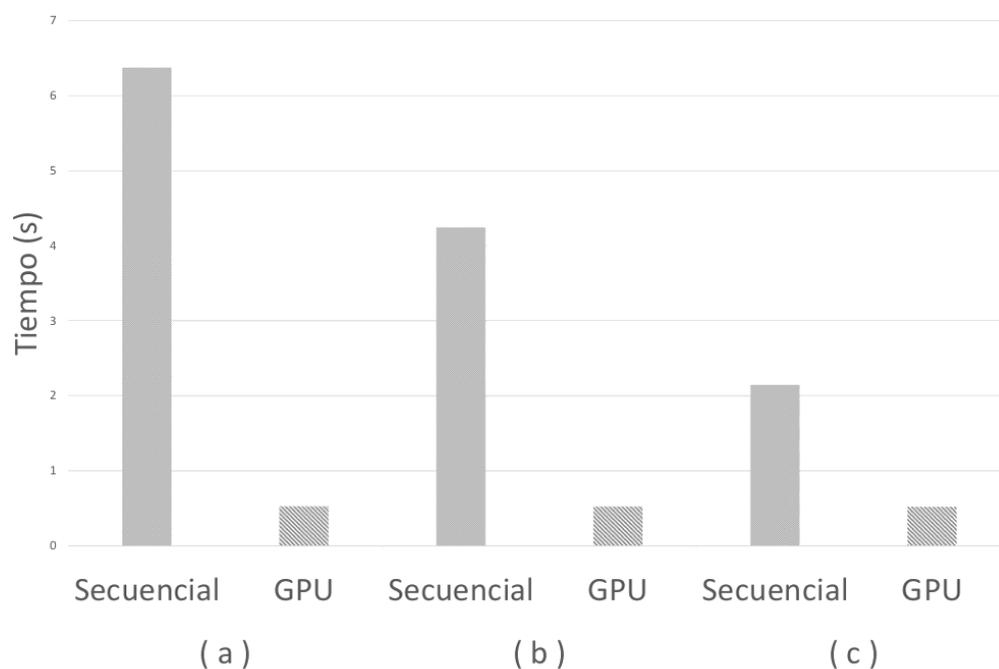


FIGURA 7.3: Comparación de tiempo de ejecución entre la implementación secuencial ejecutada en CPU y la implementación paralela para GPU del filtro Fuzzy PGA: (a) aplicando los dos pasos de filtrado, (b) aplicando el paso de filtrado de ruido impulsivo y (c) aplicando el paso de filtrado de ruido gaussiano.

que prácticamente todo el tiempo medido en el estudio corresponde al movimiento de los datos en memoria.

Con el tiempo de ejecución conseguido se podrían tratar 2 imágenes por segundo, considerando una CT compuesta de 360 imágenes, se podrían postprocesar con este filtro en aproximadamente 180 segundos, 3 minutos. Estos resultados están aún lejos de poder utilizarse en *streaming* de vídeo. Se necesitaría poder procesar 24 imágenes (en vídeo denominadas *frames*) por segundo. Teniendo en cuenta que el *kernel* se ejecuta en sólo 0,000016 ms, es decir, que el cuello de botella se encuentra en el uso de la memoria, se podría realizar un *buffering* de los datos y trabajar en el procesamiento paralelo de varios *frames*.

Las técnicas de mejora implementadas incidían sobre la ejecución del *kernel*, no sobre el cuello de botella detectado, de modo que para el caso de imágenes pequeñas no son apreciables, sin embargo podrían ser interesantes para el uso del filtro en imágenes de alta resolución mucho más grandes que las utilizadas en este estudio.

Capítulo 8

Conclusiones

En este trabajo se ha estudiado un filtro para la reducción de ruido impulsivo y gaussiano en imagen médica. Se ha utilizado una implementación propia del filtro Fuzzy PGA y como base de datos de imágenes una colección de mamogramas denominada mini-MIAS.

A fin de adecuar el filtro al tipo de imagen filtrada se deben encontrar los valores óptimos para los parámetros de entrada F_t y F_σ . Se realiza un experimento multiparamétrico que consiste en aplicar el filtro sobre cada una de las imágenes que han sido previamente contaminada con ruido impulsivo y gaussiano. Este experimento se realizó utilizando infraestructura grid, y se encontró que los valores óptimos eran $F_t = 0,85$ y $F_\sigma = 480$.

Una vez obtenidos los parámetros de entrada del filtro Fuzzy PGA, se evalúa su eficacia comparándolo con otros métodos que han sido validados en estudios anteriores: FNLDF, PGFM y NLF. Esta evaluación se realiza usando imágenes mamográficas contaminadas con distintos tipos de ruido. En todos los casos Fuzzy PGA muestra un comportamiento correcto, obteniendo mejores resultados, en base a la puntuación PSNR, que el resto de técnicas de filtrado. Concretamente, el filtrado de mamogramas con ruido impulsivo con Fuzzy PGA obtiene un $PSNR = 41,2335$, mejorando a PGFM en 0,54 puntos, a NDF en 16,85 y a FNLDF en 1,67. En el de caso de ruido gaussiano Fuzzy PGA obtiene un $PSNR = 29,7441$, mejorando a PGFM en 6,47 puntos, a NDF en 1,09 y a FNLDF en 1,6. Y por último, para la mezcla de ambos ruidos Fuzzy PGA obtiene un $PSNR = 29,3803$, mejorando a PGFM en 6,34 puntos, a NDF en 6,01 y a FNLDF en 1,39. Esto significa, no sólo que Fuzzy PGA supera al resto de filtros considerados si no que además muestra un buen comportamiento sea cual sea el tipo de ruido que posea la imagen mamográfica (de entre los considerados).

Habiendo demostrado que el filtro Fuzzy PGA es un método de reducción de ruido eficaz en radiografía mamográfica, se puede considerar para su aplicación en la reducción de

dosis radiactiva tanto en mamografía como en otros ámbitos relacionados como podría ser en la CT, donde la dosis radiactiva a la que se somete al paciente es muy alta debido a que es necesaria la toma de muchas imágenes desde distintos ángulos, para su posterior reconstrucción.

En imagen médica es tan importante que el filtrado limpie el ruido encontrado en una imagen como que esa limpieza no implique la destrucción de información relevante. Esto podría tener serias consecuencias en el diagnóstico de una enfermedad. En el caso de la mamografía, que se realiza con dosis mínimas de radiación, se emplean para localizar posibles microcalcificaciones que aparecen como puntos blanquecinos en la imagen. Si se altera esta información podrían desaparecer estas alteraciones. Para comprobar que el filtro Fuzzy PGA se comporta adecuadamente sean cuales sean las características de la imagen se realiza un estudio de estabilidad del filtro.

El estudio de estabilidad demuestra que en todos los casos, sea cual sea el tejido subyacente de la mama, la anormalidad presente (exista o no), y el diagnóstico de ésta, el filtro Fuzzy PGA traza siempre una distribución normal, en base a los resultados de PSNR que comparan imagen filtrada con imagen original, y que estas distribuciones se centran en el intervalo de valores [27, 28], lo cual es consistente con el estudio de eficacia realizado. Por tanto, se concluye que no existe correlación entre las características de la mamografía y el funcionamiento del filtro, incluyendo en estas características el tipo de tejido, el tipo de masa y la benignidad o malignidad de la masa.

Por último, es deseable que cualquier postproceso de imagen se realice en un tiempo razonable, muy importante si se deben procesar colecciones de imágenes numerosas como podría ser en el caso de una CT, compuesta por tantas imágenes como ángulos de proyección se empleen. La solución aportada es la implementación del filtro Fuzzy PGA para GPU, utilizando la tecnología CUDA. Con esta implementación se consigue un *speedup* de hasta 12 unidades, en el caso de limpieza de tanto ruido impulsivo como gaussiano. La implementación en GPU es capaz de procesar 2 imágenes por segundo aproximadamente. Tiempo que resulta aún insuficiente para el *streaming* de vídeo, que necesitaría el procesamiento de 24 imágenes por segundo.

Sin embargo, se observa que la ejecución del propio filtro de la versión GPU, es decir, la ejecución del *kernel*, emplea tan solo 0,000016ms del tiempo total. Esto significa que la mayor parte del tiempo se emplea en operaciones de intercambio de memoria y establecimiento del contexto, y que el filtro Fuzzy PGA se adecua muy bien al empleo de GPUs.

En definitiva, este es trabajo multidisciplinar que aúna principios algorítmicos, medicina radiológica y computación de altas prestaciones. Se ha conseguido implementar un filtro

Fuzzy PGA para su uso en imagen médica, concretamente en imagen mamográfica, mejorando tanto el rendimiento en eficacia con respecto a otros filtros evaluados en estudios preliminares, como la eficiencia temporal mediante el empleo de una implementación en GPU. Para todo ello se han utilizando parte de los recursos aprendidos en el Máster Universitario en Computación Paralela y Distribuida de la Universitat Politècnica de València.

Publicaciones

E. Parceró, D. Segrelles, V. Vidal, I. Blanquer, G. Verdú. *Towards Grid Environment to Perform Filtering Techniques for Improving Medical Images: an Use case of Mammographic Images*. IBERGRID, 183-192, Portugal (2014).

E. Parceró, V. Vidal, G. Verdú, P. Mayo. *Estudio de Estabilidad de Técnicas de Filtrado en Imágenes de la Base de Datos mini-MIAS*. SNE, España (2014).

E. Parceró, V. Vidal, G. Verdú, J. Arnal, P. Mayo. *Análisis del Filtro FPGA en Imágenes de Tomografía Computarizada para la Reducción de Dosis Radiactiva*. SNE, España (2014).

E. Parceró, D. Segrelles, V. Vidal, I. Blanquer, G. Verdú. *Fuzzy PGA Filter Technique Performance on the mini-MIAS Database of Mammograms*. Aceptado para su publicación en *International Journal of Image Mining*, 2015.

J. Arnal, E. Parceró, V. Vidal. *Real Time Medical Image Restoration*. Bajo revisión en el congreso ICCS 2015.

Capítulo 9

Trabajo Futuro

Como trabajo futuro se plantea la realización de un estudio detallado de la efectividad y cuantificación del filtro Fuzzy PGA para la reducción de dosis radiactiva. Que consistiría en la aplicación del filtro sobre imágenes tomadas con distintas dosis y su posterior análisis.

En cuanto a la mejora de rendimiento temporal, se propone la paralelización del filtro con tecnologías que operen sobre la CPU, y la optimización de la implementación realizada en GPU. Además, esta última puede resultar de especial interés en el filtrado de imágenes de gran tamaño, por tanto se sugiere la adaptación de la implementación en GPU para tamaños de imagen grandes y la realización de un análisis más profundo del rendimiento en función de dichos tamaños.

El área de la imagen médica es muy amplia, y no sólo se emplean radiografías, por tanto sería de gran interés el estudio de otro tipo de imágenes como por ejemplo MRI (Imagen de Resonancia Magnética). En este caso sería incluso necesario incorporar técnicas de filtrado de distinta naturaleza, para la reducción de otros tipos de ruido.

Bibliografia

- [1] R.C. Gonzalez and R.E. Woods. *Digital Image Processing*. Pearson/Prentice Hall, 2008. ISBN 9780131687288. URL <http://books.google.es/books?id=8uG0njRGEzoC>.
- [2] MS May, A Eller, C Stahl, W Wuest, M Scharf, M Hammon, P Dankerl, PM Schlechtweg, T Allmendinger, M Sedlmair, et al. Dose reduction in computed tomography of the chest: image quality of iterative reconstructions at a 50 % radiation dose compared to filtered back projection at a 100 % radiation dose. *RoFo: Fortschritte auf dem Gebiete der Rontgenstrahlen und der Nuklearmedizin*, 186(6):576–584, 2014.
- [3] Joan Gerard Camarena Estruch. Aplicació de mètriques fuzzy en la millora computacional d'algorismes de filtratge d'imatges en color. 2009.
- [4] S. Morillas, V. Gregori, and A. Hervás. Fuzzy peer groups for reducing mixed gaussian-impulse noise from color images. *IEEE Transactions on Image Processing*, 18(7):1452–1466, July 2009. ISSN 1057-7149. doi: 10.1109/TIP.2009.2019305.
- [5] Michael Lindenbaum, M Fischer, and A Bruckstein. On gabor's contribution to image enhancement. *Pattern Recognition*, 27(1):1–8, 1994.
- [6] J.C. Russ. *The Image Processing Handbook, Fifth Edition*. CRC Press, December 2006. ISBN 9780203881095.
- [7] M. Bertero and P. Boccacci. *Introduction to Inverse Problems in Imaging*. CRC Press, January 1998. ISBN 9781439822067.
- [8] F. Catté, P. Lions, J. Morel, and T. Coll. Image selective smoothing and edge detection by nonlinear diffusion. *SIAM Journal on Numerical Analysis*, 29(1):182–193, February 1992. ISSN 0036-1429. doi: 10.1137/0729012. URL <http://epubs.siam.org/doi/abs/10.1137/0729012>.

- [9] L.I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1–4):259–268, November 1992. ISSN 0167-2789. doi: 10.1016/0167-2789(92)90242-F. URL <http://www.sciencedirect.com/science/article/pii/016727899290242F>.
- [10] J. Weickert, B.M.T.H. Romeny, and M.A. Viergever. Efficient and reliable schemes for nonlinear diffusion filtering. *IEEE Transactions on Image Processing*, 7(3):398–410, March 1998. ISSN 1057-7149. doi: 10.1109/83.661190.
- [11] A. Marquina and S. Osher. Explicit algorithms for a new time dependent model based on level set motion for nonlinear deblurring and noise removal. *SIAM Journal on Scientific Computing*, 22(2):387–405, January 2000. ISSN 1064-8275. doi: 10.1137/S1064827599351751. URL <http://epubs.siam.org/doi/abs/10.1137/S1064827599351751>.
- [12] T. Chan, A. Marquina, and P. Mulet. High-order total variation-based image restoration. *SIAM Journal on Scientific Computing*, 22(2):503–516, January 2000. ISSN 1064-8275. doi: 10.1137/S1064827598344169. URL <http://epubs.siam.org/doi/abs/10.1137/S1064827598344169>.
- [13] S.L. Keeling. Total variation based convex filters for medical imaging. *Applied Mathematics and Computation*, 139(1):101–119, July 2003. ISSN 0096-3003. doi: 10.1016/S0096-3003(02)00171-6. URL <http://www.sciencedirect.com/science/article/pii/S0096300302001716>.
- [14] J. Weickert. Efficient image segmentation using partial differential equations and morphology. *Pattern Recognition*, 34(9):1813–1824, September 2001. ISSN 0031-3203. doi: 10.1016/S0031-3203(00)00109-6. URL <http://www.sciencedirect.com/science/article/pii/S0031320300001096>.
- [15] P. Mrázek and M. Navara. Selection of optimal stopping time for nonlinear diffusion filtering. *International Journal of Computer Vision*, 52(2-3):189–203, May 2003. ISSN 0920-5691, 1573-1405. doi: 10.1023/A:1022908225256. URL <http://link.springer.com/article/10.1023/A%3A1022908225256>.
- [16] D.L. Donoho. De-noising by soft-thresholding. *IEEE Transactions on Information Theory*, 41(3):613–627, May 1995. ISSN 0018-9448. doi: 10.1109/18.382009.
- [17] Pietro Perona and Jitendra Malik. Scale-space and edge detection using anisotropic diffusion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(7): 629–639, 1990.

- [18] Luis Alvarez, Pierre-Louis Lions, and Jean-Michel Morel. Image selective smoothing and edge detection by nonlinear diffusion. ii. *SIAM Journal on numerical analysis*, 29(3):845–866, 1992.
- [19] Leonid I Rudin and Stanley Osher. Total variation based image restoration with free local constraints. In *Image Processing, 1994. Proceedings. ICIP-94., IEEE International Conference*, volume 1, pages 31–35. IEEE, 1994.
- [20] Eitan Tadmor, Suzanne Nezzar, and Luminita Vese. A multiscale image representation using hierarchical (bv, l 2) decompositions. *Multiscale Modeling & Simulation*, 2(4):554–579, 2004.
- [21] Stanley Osher, Martin Burger, Donald Goldfarb, Jinjun Xu, and Wotao Yin. An iterative regularization method for total variation-based image restoration. *Multiscale Modeling & Simulation*, 4(2):460–489, 2005.
- [22] Leonid P Yaroslavsky and LP Yaroslavskij. Digital picture processing. an introduction. *Digital picture processing. An introduction.. LP Yaroslavsky (LP Yaroslavskij). Springer Series in Information Sciences, Vol. 9. Springer-Verlag, Berlin-Heidelberg-New York-Tokyo. 12+ 276 pp. Price DM 112.00 (1985). ISBN 3-540-11934-5 (FR Germany), ISBN 0-387-11934-5 (USA).*, 1, 1985.
- [23] Leonid P. Yaroslavsky and Murray Eden. *Fundamentals of Digital Optics: Digital Signal Processing in Optics and Holography*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1st edition, 1996. ISBN 0817638229.
- [24] Stephen M Smith and J Michael Brady. Susan—a new approach to low level image processing. *International journal of computer vision*, 23(1):45–78, 1997.
- [25] Leonid P Yaroslavsky. Local adaptive image restoration and enhancement with the use of dft and dct in a running window. In *SPIE's 1996 International Symposium on Optical Science, Engineering, and Instrumentation*, pages 2–13. International Society for Optics and Photonics, 1996.
- [26] David L Donoho and Jain M Johnstone. Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, 81(3):425–455, 1994.
- [27] Ronald R Coifman and David L Donoho. *Translation-invariant de-noising*. Springer, 1995.
- [28] Tsachy Weissman, Erik Ordentlich, Gadiel Seroussi, Sergio Verdú, and Marcelo J Weinberger. Universal discrete denoising: Known channel. *Information Theory, IEEE Transactions on*, 51(1):5–28, 2005.

-
- [29] Patrick Guidotti and James V Lambers. Two new nonlinear nonlocal diffusions for noise reduction. *Journal of Mathematical Imaging and Vision*, 33(1):25–37, 2009.
- [30] Mohammad Reza Hajiaboli. An anisotropic fourth-order diffusion filter for image noise removal. *International Journal of Computer Vision*, 92(2):177–191, 2011.
- [31] XY Liu, C-H Lai, KA Pericleous, and MQ Wang. On a modified diffusion model for noise removal. *Journal of Algorithms & Computational Technology*, 6(1):35–58, 2012.
- [32] E Umbaugh Scott. Computer vision and image processing-a practical approach using cviptools. *Practice Hall PTR*, 1998.
- [33] Ioannis Pitas and Anastasios N Venetsanopoulos. *Nonlinear digital filters*, volume 84. Springer, 1990.
- [34] Herbert Aron David and Haikady Navada Nagaraja. *Order statistics*. Wiley Online Library, 1970.
- [35] J. Astola, P. Haavisto, and Y. Neuvo. Vector median filters. *Proceedings of the IEEE*, 78(4):678–689, April 1990. ISSN 0018-9219. doi: 10.1109/5.54807.
- [36] DRK Brownrigg. The weighted median filter. *Communications of the ACM*, 27(8):807–818, 1984.
- [37] Alfredo Restrepo and Alan Conrad Bovik. Adaptive trimmed mean filters for image restoration. *Acoustics, Speech and Signal Processing, IEEE Transactions on*, 36(8):1326–1337, 1988.
- [38] Sung-Jea Ko and Yong Hoon Lee. Center weighted median filters and their applications to image enhancement. *Circuits and Systems, IEEE Transactions on*, 38(9):984–993, 1991.
- [39] Tong Sun and Yrjö Neuvo. Detail-preserving median based filters in image processing. *Pattern Recognition Letters*, 15(4):341–347, 1994.
- [40] L Garcia-Cabrera, PL Luque-Escamilla, J Martinez-Aroza, AM Robles-Perez, and R Roman-Roldan. Two pixel-preselection methods for median-type filtering. *IEE Proceedings-Vision, Image and Signal Processing*, 145(1):30–40, 1998.
- [41] Y Xu and EM-K Lai. Restoration of images contaminated by mixed gaussian and impulse noise using a recursive minimum–maximum method. *IEE Proceedings-Vision, Image and Signal Processing*, 145(4):264–270, 1998.

- [42] J.G. Camarena, V. Gregori, S. Morillas, and A. Sapena. Fast detection and removal of impulsive noise using peer groups and fuzzy metrics. *Journal of Visual Communication and Image Representation*, 19(1):20–29, January 2008. ISSN 1047-3203. doi: 10.1016/j.jvcir.2007.04.003. URL <http://www.sciencedirect.com/science/article/pii/S1047320307000259>.
- [43] B. Smolka and A. Chydzinski. Fast detection and impulsive noise removal in color images. *Real-Time Imaging*, 11(5–6):389–402, October 2005. ISSN 1077-2014. doi: 10.1016/j.rti.2005.07.003. URL <http://www.sciencedirect.com/science/article/pii/S1077201405000689>.
- [44] M.G. Sánchez, V. Vidal, J. Bataller, and J. Arnal. A fuzzy metric in GPUs: Fast and efficient method for the impulsive image noise removal. In Erol Gelenbe, Ricardo Lent, and Georgia Sakellari, editors, *Computer and Information Sciences II*, pages 323–330. Springer London, January 2012. ISBN 978-1-4471-2154-1, 978-1-4471-2155-8. URL http://link.springer.com/chapter/10.1007/978-1-4471-2155-8_41.
- [45] M.G. Sanchez, V. Vidal, G. Verdu, P. Mayo, and F. Rodenas. Medical image restoration with different types of noise. In *IEEE, Engineering in Medicine and Biology Society*, pages 4382–4385, 2012.
- [46] V. Vidal, J Bataller, J. Arnal, and M.G. Sanchez. A parallel method for impulsive image noise removal on hybrid cpu/gpu systems. In *International Conference on Computational Science (ICCS): Computation at the Frontiers of Science*, pages 2504–2507, 2013.
- [47] G Verdu, B. Jeanine, V. Vidal, and F. Rodenas. Dose reduction using non-linear diffusion and smoothing filters in computed radiography. In *12th International Symposium on Radiation Physics*, page 120, 2012.
- [48] G Verdu, B. Jeanine, V. Vidal, and P. Mayo. Estimated radiation dose reduction using non-linear diffusion method in computed radiography. In *34th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, pages 1502–1505, 2012.
- [49] Joan-Gerard Camarena, Valentín Gregori, Samuel Morillas, and Almanzor Sapena. Two-step fuzzy logic-based method for impulse noise detection in colour images. *Pattern Recognition Letters*, 31(13):1842–1849, 2010.
- [50] Tom Mélange, Mike Nachtegael, and Etienne E Kerre. Fuzzy random impulse noise removal from color image sequences. *Image Processing, IEEE Transactions on*, 20(4):959–970, 2011.

- [51] Eric Michel-González, Min Hyoung Cho, and Soo Yeol Lee. Geometric nonlinear diffusion filter and its application to x-ray imaging. *Biomedical engineering online*, 10(1):47, 2011.
- [52] Abdullah Toprak and İnan Güler. Impulse noise reduction in medical images with the use of switch mode fuzzy adaptive median filter. *Digital signal processing*, 17(4):711–723, 2007.
- [53] Adriënne M Mendrik, E-J Vonken, Annemarieke Rutten, Max A Viergever, and Bram van Ginneken. Noise reduction in computed tomography scans using 3-d anisotropic hybrid diffusion with continuous switch. *Medical Imaging, IEEE Transactions on*, 28(10):1585–1594, 2009.
- [54] Paul Bao and D Zhang. Noise reduction for magnetic resonance images via adaptive multiscale products thresholding. *Medical Imaging, IEEE Transactions on*, 22(9):1089–1099, 2003.
- [55] MadhuS. Nair and J. Reji. An efficient directional weighted median switching filter for impulse noise removal in medical images. In Ajith Abraham, Jaime-Lloret Mauri, JohnF. Buford, Junichi Suzuki, and SabuM. Thampi, editors, *Advances in Computing and Communications*, volume 192 of *Communications in Computer and Information Science*, pages 276–288. Springer Berlin Heidelberg, 2011. ISBN 978-3-642-22719-6. doi: 10.1007/978-3-642-22720-2_28. URL http://dx.doi.org/10.1007/978-3-642-22720-2_28.
- [56] A Padma, R Sukanesh, and A Santhana Vijayan. A new genetic based multilayered fuzzy image filter for removing additive identical independent distribution impulse noise from medical images. *International Journal of Computer Applications*, 1(8):102–113, 2010.
- [57] Alvin C Silva, Holly J Lawder, Amy Hara, Jennifer Kujak, and William Pavlicek. Innovations in ct dose reduction strategy: application of the adaptive statistical iterative reconstruction algorithm. *American Journal of Roentgenology*, 194(1):191–199, 2010.
- [58] Jonathon Leipsic, Troy M LaBounty, Brett Heilbron, James K Min, GB John Mancini, Fay Y Lin, Carolyn Taylor, Allison Dunning, and James P Earls. Estimated radiation dose reduction using adaptive statistical iterative reconstruction in coronary ct angiography: the erasir study. *American Journal of Roentgenology*, 195(3):655–660, 2010.

- [59] Sarabjeet Singh, Mannudeep K Kalra, Michael A Moore, Randheer Shailam, Bob Liu, Thomas L Toth, Ellen Grant, and Sjik J Westra. Dose reduction and compliance with pediatric ct protocols adapted to patient size, clinical indication, and number of prior studies 1. *Radiology*, 252(1):200–208, 2009.
- [60] B Juste, JI Villaescusa, R Tortosa, R Miro, and G Verdu. Analysis of cr dose reduction in pediatric patients, based on computer-simulated noise addition. In *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, pages 2697–2700. IEEE, 2008.
- [61] NVIDIA Corporation. *NVIDIA CUDA Compute Unified Device Architecture Programming Guide*. NVIDIA Corporation, 2007.
- [62] Advanced Micro Devices, Inc. *ATI CTM Guide. Technical Reference Manual*. Advanced Micro Devices, Inc., 2006.
- [63] Piotr S Windyga. Fast impulsive noise removal. *Image Processing, IEEE Transactions on*, 10(1):173–179, 2001.
- [64] VF Kravchenko, VI Ponomaryov, and VI Pustovoit. Three-dimensional filtration of multichannel video sequences on the basis of fuzzy-set theory. In *Doklady Physics*, volume 55, pages 58–63. Springer, 2010.
- [65] Liwei Guo, Oscar C Au, Mengyao Ma, and Peter HW Wong. Integration of recursive temporal lmmse denoising filter into video codec. *Circuits and Systems for Video Technology, IEEE Transactions on*, 20(2):236–249, 2010.
- [66] Simon Harding. Evolution of image filters on graphics processor units using cartesian genetic programming. In *Evolutionary Computation, 2008. CEC 2008. (IEEE World Congress on Computational Intelligence). IEEE Congress on*, pages 1921–1928. IEEE, 2008.
- [67] J. Arnal, L.B. Sucar, M.G. Sanchez, and V. Vidal. Parallel filter for mixed gaussian-impulse noise removal. In *Signal Processing: Algorithms, Architectures, Arrangements, and Applications (SPA), 2013*, pages 236–241, September 2013.
- [68] J. Suckling et al. The mammographic image analysis society digital mammogram database. In *Exerpta Medica*, pages 375–378, 1994.
- [69] European grid infrastructure, June 2014. URL <http://www.egi.eu>.
- [70] C. Lazou. Plans for a future european grid infrastructure organization. *European Parliament Magazine*, 265:8–9, April 2008. URL http://web.eu-egi.eu/fileadmin/public/documents/Article_Parliament_Magazine.pdf.

-
- [71] CERN. gLite: Lightweight middleware for grid computing. URL <http://glite.web.cern.ch/glite/>.
- [72] I. Pitas and A.N. Venetsanopoulos. Nonlinear mean filters in image processing. *IEEE transactions on acoustics, speech, and signal processing*, 3:573, 1986.
- [73] C. Vogel and M. Oman. Iterative methods for total variation denoising. *SIAM Journal on Scientific Computing*, 17(1):227–238, January 1996. ISSN 1064-8275. doi: 10.1137/0917016. URL <http://epubs.siam.org/doi/abs/10.1137/0917016>.