



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Estudio y desarrollo de métodos de *zoom* para imágenes de
gammagrafía obtenidas mediante la aplicación de
meta-yodobenzilguanidina

Autor: Juan Cartes Font
Director: Samuel Morillas Gómez

Ingeniería Informática
Escuela Técnica Superior de Ingeniería Informática
Universitat Politècnica de València

Valencia, 28 de septiembre de 2011

Resumen

El presente proyecto trata de profundizar en las técnicas actuales para proporcionar un *zoom* digital aplicable a todo tipo de imágenes digitales, especialmente a imágenes médicas como los son las gammagrafías. Este proyecto se encuentra incluido en otro más amplio que tiene por objeto apoyar a la toma de decisiones por parte de los médicos en el proceso de diagnóstico de neuroblastomas.

En esta memoria se detallan una serie de técnicas interpolatorias no lineales y se proponen posibles implementaciones, con el objeto de elegir aquella que mejores resultados proporcione y que pueda ser incorporada al proyecto marco, como una funcionalidad más.

En la sección de experimentos, se realizarán diversas pruebas con cada una de las técnicas que serán evaluadas mediante el uso de ciertos indicadores para imágenes digitales, de manera que quede justificada alguna de ellas frente otras para un determinado tipo de imágenes.

PALABRAS CLAVE: Interpolación, Neuroblastoma, *zoom*, imagen médica, gammagrafía.

Índice general

1. Introducción	15
1.1. El neuroblastoma	15
1.1.1. Cuadro clínico	16
1.1.2. Diagnóstico	16
1.2. Contexto	17
1.3. Objetivos	17
1.3.1. Filtrado en el dominio de la frecuencia	18
1.3.2. Filtrado en el dominio del espacio o realce	18
1.3.3. <i>Zoom</i>	19
1.4. Estructura de la memoria	20
2. Imagen médica	21
2.1. Las imágenes DICOM	21
2.1.1. Historia	22
2.1.2. Estructura	22
3. Técnicas de interpolación no lineales	31
3.1. Introducción	31
3.2. Multirresolución de Harten	33
3.3. Métodos de interpolación no lineales	38
3.3.1. Interpolación ENO	40
3.3.2. Interpolación ENO <i>Subcell Resolution</i>	47
3.3.3. Interpolación WENO	54
3.3.4. Interpolación Racional	61
3.3.5. Interpolación PPH	65
4. Experimentos y resultados	71
4.1. Indicadores medibles para la evaluación de la calidad	71
4.1.1. Introducción	71
4.1.2. Error Cuadrático Medio	71
4.1.3. Relación señal a ruido de pico (PSNR)	72

4.1.4.	Correlación cruzada normalizada	73
4.1.5.	Diferencia media	73
4.1.6.	Diferencia Máxima	73
4.1.7.	Error Absoluto Medio	73
4.2.	Experimentos realizados	74
4.2.1.	Introducción	74
4.2.2.	Descripción y procedimientos	74
4.3.	Resultados	79
4.3.1.	Resultados con la imagen <i>Dicom1</i>	81
4.3.2.	Resultados con la imagen <i>Dicom2</i>	84
4.3.3.	Resultados con la imagen <i>Lena</i>	87
4.3.4.	Resultados con la imagen <i>Geo</i>	90
4.3.5.	Resultados con la imagen <i>Tac1</i>	93
4.3.6.	Resultados con la imagen <i>Tac2</i>	96
4.3.7.	Resultados con la imagen <i>Pet-Tc1</i>	99
4.3.8.	Resultados con la imagen <i>Pet-Tc2</i>	102
4.3.9.	Resultados con la imagen <i>Pet1</i>	105
4.3.10.	Resultados con la imagen <i>Pet2</i>	108
4.3.11.	Análisis de los resultados obtenidos	121
5.	Conclusiones	129
I.		131

Índice de figuras

2.1. Capas DICOM.	23
3.1. Ilustración del concepto « <i>píramide de resolución.</i> »	34
3.2. Representación de tres niveles de resolución, X^k , $X^{k+\frac{1}{2}}$ y X^{k+1} respectivamente. Se parte de un nivel de resolución k con el conjunto de puntos $\{x_i^k, y_j^k\}_{i,j=0}^{J_k}$. Mediante la técnica del <i>producto tensor</i> se obtienen los valores interpolados para las nuevas filas, es decir los valores para $\{x_i^{k+\frac{1}{2}}, y_j^k\}_{0 \leq i \leq J_{k+1}, 0 \leq j \leq J_k}$ correspondientes al nivel de resolución $k + \frac{1}{2}$. Por último, se obtienen los valores para las columnas alcanzando el nivel de resolución $k+1$, es decir, con el conjunto de puntos $\{x_i^{k+1}, y_j^{k+1}\}_{i,j=0}^{J_{k+1}}$. En la figura se aprecian los píxeles originales (\odot), los detalles interpolados verticales (\otimes), los interpolados horizontales (\oplus) y los mixtos (\star).	35
4.1. Imágenes anterior (a) y posterior (b) de un paciente, denominadas respectivamente <code>Dicom1</code> y <code>Dicom2</code> . Imágenes fotográficas <code>Lena.jpg</code> (a) y <code>Geo.jpg</code> (b).	77
4.2. Resto de imágenes utilizadas en los experimentos. <code>Lena.jpg</code> (a), <code>Tac1.jpg</code> (b), <code>Tac2.jpg</code> (c), <code>Pet-Tc1.jpg</code> (d), <code>Pet-Ct2.jpg</code> (e), <code>Pet1.jpg</code> (f) y <code>Pet2.jpg</code> (g).	78
4.3. Imágenes <code>Dicom1</code> con un nivel de zoom (x2) acotadas por una ventana de 175×151 píxeles. Reconstrucción con ENO no jerárquico (a), con ENO jerárquico (b), con ENO-SR (c), con WENO (d), con Racional (e), y con la técnica PPH (f).	82
4.4. Imágenes <code>Dicom1</code> con un nivel de 2 (x4) acotadas por una ventana de 175×151 píxeles. Reconstrucción con ENO no jerárquico (a), con ENO jerárquico (b), con ENO-SR (c), con WENO (d), con Racional (e), y con la técnica PPH (f).	83

4.5.	Imágenes <i>Dicom2</i> con un nivel de zoom (x2) acotadas por una ventana de 174×151 píxeles. Reconstrucción con ENO no jerárquico (<i>a</i>), con ENO jerárquico (<i>b</i>), con ENO-SR (<i>c</i>), con WENO (<i>d</i>), con Racional (<i>e</i>), y con la técnica PPH (<i>f</i>).	85
4.6.	Imágenes <i>Dicom2</i> con un nivel 2 (x4) de <i>zoom</i> acotadas por una ventana de 174×151 píxeles. Reconstrucción con ENO no jerárquico (<i>a</i>), con ENO jerárquico (<i>b</i>), con ENO-SR (<i>c</i>), con WENO (<i>d</i>), con Racional (<i>e</i>), y con la técnica PPH (<i>f</i>).	86
4.7.	Imágenes <i>Lena.jpg</i> con un nivel de zoom (x2) acotadas por una ventana de 256×256 píxeles. Reconstrucción con ENO no jerárquico (<i>a</i>), con ENO jerárquico (<i>b</i>), con ENO-SR (<i>c</i>), con WENO (<i>d</i>), con Racional (<i>e</i>), y con la técnica PPH (<i>f</i>).	88
4.8.	Imágenes <i>Lena.jpg</i> con un factor de zoom 2 (x4) acotadas por una ventana de 255×255 píxeles. Reconstrucción con ENO no jerárquico (<i>a</i>), con ENO jerárquico (<i>b</i>), con ENO-SR (<i>c</i>), con WENO (<i>d</i>), con Racional (<i>e</i>), y con la técnica PPH (<i>f</i>).	89
4.9.	Imágenes <i>Geo.jpg</i> con un nivel de zoom (x2) acotadas por una ventana de 255×255 píxeles. Reconstrucción con ENO no jerárquico (<i>a</i>), con ENO jerárquico (<i>b</i>), con ENO-SR (<i>c</i>), con WENO (<i>d</i>), con Racional (<i>e</i>), y con la técnica PPH (<i>f</i>).	91
4.10.	Imágenes <i>Geo.jpg</i> con un factor de zoom 2 (x4) acotadas por una ventana de 255×255 píxeles. Reconstrucción con ENO no jerárquico (<i>a</i>), con ENO jerárquico (<i>b</i>), con ENO-SR (<i>c</i>), con WENO (<i>d</i>), con Racional (<i>e</i>), y con la técnica PPH (<i>f</i>).	92
4.11.	Imágenes <i>Tac1.jpg</i> con un 1 nivel de zoom (x2) acotadas por una ventana de 512×542 píxeles. Reconstrucción con ENO no jerárquico (<i>a</i>), con ENO jerárquico (<i>b</i>), con ENO-SR (<i>c</i>), con WENO (<i>d</i>), con Racional (<i>e</i>), y con la técnica PPH (<i>f</i>).	94
4.12.	Imágenes <i>Tac1.jpg</i> con un factor de zoom 2 (x4) acotadas por una ventana de 512×542 píxeles. Reconstrucción con ENO no jerárquico (<i>a</i>), con ENO jerárquico (<i>b</i>), con ENO-SR (<i>c</i>), con WENO (<i>d</i>), con Racional (<i>e</i>), y con la técnica PPH (<i>f</i>).	95
4.13.	Imágenes <i>Tac2.jpg</i> con un 1 nivel de zoom (x2) acotadas por una ventana de 512×512 píxeles. Reconstrucción con ENO no jerárquico (<i>a</i>), con ENO jerárquico (<i>b</i>), con ENO-SR (<i>c</i>), con WENO (<i>d</i>), con Racional (<i>e</i>), y con la técnica PPH (<i>f</i>).	97
4.14.	Imágenes <i>Tac2.jpg</i> con un factor de zoom 2 (x4) acotadas por una ventana de 512×512 píxeles. Reconstrucción con ENO no jerárquico (<i>a</i>), con ENO jerárquico (<i>b</i>), con ENO-SR (<i>c</i>), con WENO (<i>d</i>), con Racional (<i>e</i>), y con la técnica PPH (<i>f</i>).	98

4.15. Imágenes *Pet1.jpg* con un nivel de zoom (x2) acotadas por una ventana de 961×512 píxeles. Reconstrucción con ENO no jerárquico (*a*), con ENO jerárquico (*b*), con ENO-SR (*c*), con WENO (*d*), con Racional (*e*), y con la técnica PPH (*f*). 100

4.16. Imágenes *Pet-Tc1.jpg* con un factor de zoom 2 (x4) acotadas por una ventana de 961×512 píxeles. Reconstrucción con ENO no jerárquico (*a*), con ENO jerárquico (*b*), con ENO-SR (*c*), con WENO (*d*), con Racional (*e*), y con la técnica PPH (*f*). 101

4.17. Imágenes *Pet-Ct2.jpg* con un nivel de zoom (x2) acotadas por una ventana de 516×542 píxeles. Reconstrucción con ENO no jerárquico (*a*), con ENO jerárquico (*b*), con ENO-SR (*c*), con WENO (*d*), con Racional (*e*), y con la técnica PPH (*f*). 103

4.18. Imágenes *Pet-Ct2.jpg* con un factor de zoom 2 (x4). Reconstrucción con ENO no jerárquico (*a*), con ENO jerárquico (*b*), con ENO-SR (*c*), con WENO (*d*), con Racional (*e*), y con la técnica PPH (*f*). 104

4.19. Imágenes *Pet1.jpg* con un nivel de zoom (x2) acotadas por una ventana de 255×255 píxeles. Reconstrucción con ENO no jerárquico (*a*), con ENO jerárquico (*b*), con ENO-SR (*c*), con WENO (*d*), con Racional (*e*), y con la técnica PPH (*f*). 106

4.20. Imágenes *Pet1.jpg* con un factor de zoom 2 (x4) acotadas por una ventana de 255×255 píxeles. Reconstrucción con ENO no jerárquico (*a*), con ENO jerárquico (*b*), con ENO-SR (*c*), con WENO (*d*), con Racional (*e*), y con la técnica PPH (*f*). 107

4.21. Imágenes *Pet2.jpg* con un nivel de zoom (x2) acotadas por una ventana de 400×256 píxeles. Reconstrucción con ENO no jerárquico (*a*), con ENO jerárquico (*b*), con ENO-SR (*c*), con WENO (*d*), con Racional (*e*), y con la técnica PPH (*f*). 109

4.22. Imágenes *Pet2.jpg* con un factor de zoom 2 (x4) acotadas por una ventana de 400×256 píxeles. Reconstrucción con ENO no jerárquico (*a*), con ENO jerárquico (*b*), con ENO-SR (*c*), con WENO (*d*), con Racional (*e*), y con la técnica PPH (*f*). 110

4.23. Gráfico que muestra los valores del PSNR obtenidos para cada técnica y para cada nivel, aplicados sobre la imagen *Dicom1*. 111

4.24. Gráfico que muestra los valores del PSNR obtenidos para cada técnica y para cada nivel, aplicados sobre la imagen *Dicom2*. 111

4.25. Gráfico que muestra los valores del PSNR obtenidos para cada técnica y para cada nivel, aplicados sobre la imagen *Lena.jpg*. 112

4.26. Gráfico que muestra los valores del PSNR obtenidos para cada técnica y para cada nivel, aplicados sobre la imagen *Geo.jpg*. 112

4.27.	Gráfico que muestra los valores del PSNR obtenidos para cada técnica y para cada nivel, aplicados sobre la imagen Tac1.jpg.	113
4.28.	Gráfico que muestra los valores del PSNR obtenidos para cada técnica y para cada nivel, aplicados sobre la imagen Tac2.jpg.	113
4.29.	Gráfico que muestra los valores del PSNR obtenidos para cada técnica y para cada nivel, aplicados sobre la imagen Pet-Tc1.jpg.	114
4.30.	Gráfico que muestra los valores del PSNR obtenidos para cada técnica y para cada nivel, aplicados sobre la imagen Pet-Tc2.jpg.	114
4.31.	Gráfico que muestra los valores del PSNR obtenidos para cada técnica y para cada nivel, aplicados sobre la imagen Pet1.jpg.	115
4.32.	Gráfico que muestra los valores del PSNR obtenidos para cada técnica y para cada nivel, aplicados sobre la imagen Pet2.jpg.	115
4.33.	Gráficos de medias obtenidas para cada técnica en cada nivel.	116
4.34.	Gráficos de medias obtenidas para cada imagen en cada nivel.	116
4.35.	Histograma que resume el porcentaje de veces que una técnica consigue mejores resultados que el resto para una configuración (<i><Imagen><Técnica><Nivel></i>). Cada barra aparece de un color diferente y atiende a la división en grupos establecida en base a la calidad obtenida. Aparecen en verde la técnica incluida en el grupo de <i>alta calidad</i> , en ámbar las clasificadas en el grupo de <i>calidad media</i> y en aquéllas del denominado <i>grupo de baja calidad</i>	119
4.36.	Gráfico de dispersión obtenido en función del PSNR obtenidos para cada tipo de imagen.	119
4.37.	Gráfico de dispersión obtenido en función del PSNR obtenidos para cada nivel de <i>zoom</i>	120
4.38.	Gráfico de dispersión obtenido en función del PSNR obtenidos para cada técnica interpolatoria estudiada.	120

Índice de tablas

2.1. Valores de los campos (1/2).	26
2.2. Valores de los campos (2/2).	27
2.3. Sintaxis de transferencia más usuales. La descripción técnica de todas ellas puede ser encontrada en [16]	29
3.1. Pesos óptimos	55
4.1. Resultados numéricos para la imagen <i>Dicom1</i> mediante el uso de las diferentes técnicas con diferentes factores de <i>zoom</i>	81
4.2. Resultados numéricos para la imagen <i>Dicom2</i> mediante el uso de las diferentes técnicas con diferentes factores de <i>zoom</i>	84
4.3. Resultados numéricos para la imagen <i>Lena.jpg</i> mediante el uso de las diferentes técnicas con diferentes factores de <i>zoom</i>	87
4.4. Resultados numéricos para la imagen <i>Geo.jpg</i> mediante el uso de las diferentes técnicas con diferentes factores de <i>zoom</i>	90
4.5. Resultados numéricos para la imagen <i>Tac1.jpg</i> mediante el uso de las diferentes técnicas con diferentes factores de <i>zoom</i>	93
4.6. Resultados numéricos para la imagen <i>Tac2.jpg</i> mediante el uso de las diferentes técnicas con diferentes factores de <i>zoom</i>	96
4.7. Resultados numéricos para la imagen <i>Pet-Tc1.jpg</i> mediante el uso de las diferentes técnicas con diferentes factores de <i>zoom</i>	99
4.8. Resultados numéricos para la imagen <i>Pet-Tc2.jpg</i> mediante el uso de las diferentes técnicas con diferentes factores de <i>zoom</i>	102
4.9. Resultados numéricos para la imagen <i>Pet1.jpg</i> mediante el uso de las diferentes técnicas con diferentes factores de <i>zoom</i>	105
4.10. Resultados numéricos para la imagen <i>Pet2.jpg</i> mediante el uso de las diferentes técnicas con diferentes factores de <i>zoom</i>	108
4.11. Tabla que resume los valores de PSNR obtenido para las distintas imágenes agrupados por tipo de técnica.	117
4.12. Media y desviación estándar de los valores del PSNR de todas las imágenes obtenidos para cada técnica y nivel de <i>zoom</i>	118

- 4.13. Media y desviación estándar de los valores del PSNR obtenidos para todas las técnicas agrupadas por tipo de imagen y nivel de *zoom*. . . 118

Índice de Algoritmos

1.	Selección jerárquica del <i>stencil</i> ENO	41
2.	Selección no jerárquica del <i>stencil</i> ENO	41
3.	Interpolación ENO	43
4.	ENO Zoom Parte 1/3	44
5.	ENO Zoom Parte 2/3	45
6.	ENO Zoom Parte 3/3	46
7.	Interpolación ENO-SR	49
8.	ENOSR Zoom Parte 1/3	50
9.	Stencil	51
10.	ENO-SR Zoom Parte 2/3	51
11.	Stencil	52
12.	ENO-SR Zoom Parte 2/3	52
13.	Funcion <i>Stencil</i>	53
14.	Interpolación WENO	58
15.	WENO Zoom Parte 1/2	59
16.	WENO Zoom Parte 2/2	60
17.	Interpolación Racional	63
18.	Racional Zoom	64
19.	Interpolación PPH	68
20.	PPH Zoom	69

Capítulo 1

Introducción

En presente proyecto se propone estudiar una serie de herramientas para el tratamiento de imágenes médicas de gammagrafía utilizadas en el diagnóstico y tratamiento de neuroblastomas. El objetivo general del proyecto es proporcionar a los profesionales de la medicina una herramienta útil para el diagnóstico y el seguimiento de enfermos de este tipo de tumores. Para la realización del mismo se ha contado con la colaboración del personal del Hospital La Fe de Valencia.

En aras de proporcionar una funcionalidad eficiente y útil de *zooming*, que pueda formar parte del software que se requiere fruto de la especificación exigida por el citado convenio entre el Hospital y la Universidad; se propone este estudio, donde se examinarán una serie de técnicas interpolatorias que satisfagan dicha necesidad y se evaluará cada una de ellas, a fin de determinar cuál o cuáles serían preferibles de ser implementadas en este contexto.

1.1. El neuroblastoma

Un neuroblastoma es un forma de cáncer infantil. Surge en el tejido nervioso y que frecuentemente suele comenzar en las glándulas suprarrenales, situadas en la parte superior de los riñones; aunque puede tener origen en otras partes del cuerpo humano.

Se desarrollo tiene lugar en unas células nerviosas primitivas, llamadas **neuroblastos**, que quedan en el cuerpo como remanentes de la fase embrionaria de desarrollo, desde antes del nacimiento.

El diagnóstico de este tipo de tumores es crítico, ya que son tumores típicamente agresivos, con una alta mortalidad y gran afectación del estado general del lactante. Éste suele producirse entre el primer y el tercer año de edad, y es frecuente que el diagnóstico se produzca descubriéndose de forma incidental durante la evaluación de traumatismos, infecciones u otros síntomas; momento en el cual el citado cáncer

ya se ha metastatizado hacia otros órganos.

Debido a la temprana aparición, muchos de los estudios realizados se han centrado en encontrar factores paternos relacionados con la concepción y la gestación. Estos factores incluyen, entre otros, la exposición a productos químicos en industrias específicas, el tabaquismo, el consumo de licor, el uso de medicamentos o fármacos durante el embarazo, además de otros factores relativos al nacimiento; sin embargo los resultados no han sido concluyentes.[1]

1.1.1. Cuadro clínico

El neuroblastoma puede ubicarse en cualquier punto a lo largo del sistema simpático y ello provoca que los signos posibles aparezcan en función de la ubicación del mismo. Entre muchos otros síntomas típicos, se hacen frecuentemente patentes dolores en los huesos, ojos protuberantes, hipertensión arterial y otros signos paraneoplásticos, fiebre elevada, anorexia, así como la existencia de masa palpable en el abdomen, el cuello o el tórax.

1.1.2. Diagnóstico

El diagnóstico ha de ser confirmado por un patólogo quirúrgico, teniendo en cuenta la presentación clínica, los hallazgos microscópicos y otras pruebas que pudieran haber sido llevadas a cabo en el laboratorio. Este tipo de pruebas son variables, y entre otras, podemos destacar las siguientes:

- **Análisis de orina:** En un 90 % de los casos en los que se presenta este tipo de cáncer, las células tumorales producen niveles elevados de ciertas hormonas. El cuerpo las convierte en ácidos (llamados HVA y VMA) que se excretan en la orina. Si los resultados son elevados, esta prueba puede ser una manera fácil de seguir la enfermedad y la reacción de un niño al tratamiento. También pueden ser fácilmente medidas después de que la terapia termine para determinar si la enfermedad está menguando. El problema de esta técnica es que no todos los pacientes han de tener necesariamente altos niveles de concentración de HVA y VMA.
- **Análisis de sangre:** Mediante un *conteo sanguíneo completo* o (CBC) se puede revisar si las cuentas de sangre están bajas, efecto de el crecimiento del tumor en la médula. Asimismo mediante un panel de química sanguínea puede ser supervisada la función de los riñones y el hígado; y buscar la presencia de algunas sustancias, a modo de indicadores, que puedan aumentar como resultado del crecimiento de un posible tumor.

- **Biopsia:** Para realizar un diagnóstico de neuroblastoma, se requiere una muestra del tumor del paciente mediante cirugía. El patólogo examina esta parte del tumor extirpado y determina sus características, en aras de facilitar un tratamiento más acertado.
- **Neuroimágenes:** Actualmente los estudios de diagnóstico mediante imágenes se realizan para obtener imágenes del interior del cuerpo del paciente con el objetivo de determinar la exacta ubicación de los diferentes tumores. La exploración por gammagrafía con MIBG (meta-yodobenzilguanidina), es la sustancia idónea que actúa como marcador. El MIBG es absorbido por las neuronas simpáticas en un funcionamiento análogo al del neurotransmisor norepinefrina. Las exploraciones con MIBG se han convertido en una delicada y precisa técnica de buscar la propagación de neuroblastomas. Se pueden utilizar dosis más altas de yodo radiactivo para concentrarlo en las células tumorales, lo que da una forma de radioterapia muy localizada, que puede llegar a matar el cáncer. Se trata de un nuevo enfoque al tratamiento que es utilizado cada vez más en niños en estados avanzados, o en aquéllos que el neuroblastoma ha reincidentado después de un tratamiento convencional.

1.2. Contexto

Este estudio se encuentra enmarcado dentro de otro proyecto mucho más amplio, que tiene por objeto el diseño de un *software* que ayude a los médicos en el proceso de toma de decisiones en lo que al diagnóstico, seguimiento y al tratamiento de los pacientes afectados por el neuroblastoma se refiere.

Este proyecto marco es fruto de la colaboración entre el Hospital La Fe de Valencia y la Universitat Politècnica de València, y pretende responder a las necesidades que el personal facultativo ha requerido en los sucesivos encuentros mantenidos con el Departamento de Matemática Aplicada para el establecimiento de los requisitos.

Así pues, los resultados que en esta memoria se recogen sobre la elección de una técnica interpolatoria para llevar a cabo un *zoom* digital, serán tratados y analizados a la hora de incluirlos en el mencionado *software* como una funcionalidad más.

1.3. Objetivos

El objetivo del presente estudio, es el de estudiar las técnicas de *zoom* aplicadas a imágenes digitales. El *zooming* digital está enmarcado dentro del **procesamiento digital de imágenes**, que es el conjunto de todas aquellas técnicas aplicadas a las

imágenes digitales, y destinadas a mejorar la calidad de las imágenes o a facilitar la búsqueda de información en ellas.

De entre las diversas técnicas de procesamiento digital, se pueden destacar el *filtrado*, el *realce* y el *zoom*.

1.3.1. Filtrado en el dominio de la frecuencia

El filtrado digital de imágenes se basa en la operación de convolución entre una imagen y una función filtro. El cambio de dominio de la imagen, del espacio de descripción al frecuencial, permite sustituir las convoluciones por productos, con ventajas para el proceso de cálculo.

Este tipo de filtrado permite mayor flexibilidad ya que hace posible seleccionar no solo la dirección de filtrado, sino también los intervalos de frecuencia que han de ser eliminados.

El filtrado en el dominio de la frecuencia es sencillo, poderoso y flexible. A grandes rasgos se trata de aplicar una determinada máscara o función de filtrado sobre una función, en este caso sobre una imagen en el dominio de la frecuencia. Dependiendo del tipo de filtro empleado, se eliminarán unas frecuencias u otras, alterando el espacio frecuencial de la imagen de destino. Como ejemplos de altas frecuencias se pueden citar los bordes, las líneas así como el ruido en ciertas imágenes. En contraposición, las bajas frecuencias son producidas por los cambios graduales de brillo en la imagen.

1.3.2. Filtrado en el dominio del espacio o realce

El realce, como parte integrante del procesamiento digital de imágenes, comprende una serie de operaciones que tienen por objeto mejorar la calidad de las imágenes.

Estas operaciones permiten realzar las características de brillo y contraste de una imagen, reducir su contenido de ruido, o agudizar ciertos detalles que se puedan presentar en ella.

Estas mencionadas operaciones que componen la técnica del realce, pueden ser divididas en dos grupos, según su tipo de procesamiento. Por un lado están aquellas operaciones de procesamiento puntual o de «píxel por píxel»; y por otro, aquellas de procesamiento por grupo de píxeles, o también llamadas «sobre vecindades».

Se parte de dos imágenes disponibles; una imagen de entrada, cuyos datos serán procesados y, una imagen de salida, que será el resultado de el realce.

El primer tipo de operaciones, tiende a mejorar el contraste tonal de la imagen, es decir, mejoran la diferencia entre los valores más oscuros y los más claros que se visualizan en un monitor. Este procesamiento altera los niveles de gris de los píxeles de una imagen. En la imagen de entrada, cada píxel es modificado por un

nuevo valor mediante una serie de operaciones matemáticas o relaciones lógicas. El valor resultante es colocado en la imagen de salida ocupando la misma posición que poseía en la imagen de entrada. De ahí que reciba el nombre de «píxel por píxel», ya que la transformación sucede a nivel individual y los píxeles en posiciones vecinas no tienen ningún tipo de influencia.

El segundo tipo, las llamadas operaciones de procesamiento por vecindades, mejoran el contraste espacial de la imagen, esto es, la diferencia entre el valor digital de brillo de un determinado píxel y la de sus vecinos. El objetivo es suavizar o reforzar estos contrastes espaciales de manera que los valores de brillo de cada píxel se asemejen o se distancien (en términos de brillo) más o menos respecto de sus vecinos. Como se ve, este tipo de operaciones opera sobre un conjunto de píxeles de la imagen de entrada, para producir el valor de un solo píxel en la imagen de salida; mediante la valiosa aportación de sus vecinos.

1.3.3. *Zoom*

El *zoom* digital es un método para disminuir el ángulo de visión de una imagen digital. Se logra recortando una imagen con el mismo radio de aspecto que la original, e interpolando el resultado.

En contraposición al denominado «*zoom* clásico», el *zoom* digital puede lograr cualquier aumento aunque este es directamente proporcional a la pérdida de calidad.

Las técnicas de ampliación o *zooming*, emplean polinomios para averiguar el valor de los «nuevos» píxeles que carecen de valor asignado, al redimensionar la matriz original en un cierto factor.

En la elección de estos polinomios es donde se pone de manifiesto la *linealidad* o *no linealidad* de las técnicas de ampliación. De esta manera, un algoritmo de ***zooming* lineal** es aquél que se basa en algún polinomio lineal de interpolación, por ejemplo el de Lagrange, para obtener el valor de los píxeles desconocidos a priori. Este tipo de técnicas siempre se aplica de igual manera, y no tiene en cuenta las particularidades que pueda tener una determinada imagen.

Análogamente, un algoritmo ***zooming* no lineal** hace uso de una técnica no lineal de interpolación, es decir una técnica que tenga en cuenta las discontinuidades de la imagen que se está tratando. Es por ello que éstos últimos obtienen, en principio, una mayor calidad de imagen, ya que la obtención del valor de un determinado píxel puede variar, dependiendo de la imagen de la que se parta. Es por ello que en este estudio se ha trabajado sobre éstas últimas técnicas de interpolación.

1.4. Estructura de la memoria

La presente memoria ha sido estructurada en cinco capítulos con el objetivo de facilitar tanto la necesidad y motivaciones por las que este proyecto ha surgido, como la aproximación a las técnicas planteadas para proporcionar soluciones al problema planteado.

En la primera sección se exponen las motivaciones que desencadenan la realización de este proyecto. También son descritas las condiciones contextuales en las que este se halla enmarcado, así como los objetivos y la estructura que regirá la memoria de este proyecto.

En el segundo capítulo, se profundiza en el contexto de la imagen médica y se relata brevemente la situación que a lo largo de los años ha requerido la creación de estándares en este contexto. En este bloque se exponen también algunas características técnicas de las imágenes de gammagrafía, profundizando en aquellos aspectos que conciernen más directamente al objeto de estudio.

A continuación, se encuentra un capítulo dedicado a las técnicas interpolatorias no lineales, y al marco teórico que subyace en las técnicas de *zooming* sobre imágenes digitales, es decir, a la Multirresolución de Harten. Aunque son explicados de forma teórica, se pueden encontrar detallados los algoritmos de las citadas técnicas en sus respectivas secciones.

El cuarto capítulo comprende la documentación relativa a los experimentos llevados a cabo de forma práctica, para evaluar de manera directa el impacto de las técnicas anteriormente descritas, y tienen por objeto obtener indicadores medibles, a fin de cuantificar el beneficio que éstas puedan aportar.

Por último, se encuentra la sección de conclusiones donde se exponen las ideas que tras la realización de este proyecto han surgido, y otras que pudieran surgir pero que exceden de los objetivos previamente fijados en este trabajo.

A este último capítulo le sigue un anexo, donde se puede encontrar documentación relativa implementación que se ha realizado de las distintas técnicas propuestas.

Capítulo 2

Imagen médica

Recibe el nombre de imagen médica el conjunto de «*técnicas y procesos usados para crear imágenes del cuerpo humano, o partes de él, con propósitos clínicos o para la ciencia médica*»¹. En el campo de la investigación científica, la imagen médica constituye una subdisciplina de la ingeniería biomédica, la física médica o la medicina, dependiendo del contexto de estudio. Este contexto es muy amplio y comprende actividades como la investigación el desarrollo en el área de instrumentación, adquisición de imágenes, el modelado y la cuantificación son normalmente reservadas para la ingeniería biomédica, física médica y ciencias de la computación; la investigación en la aplicación e interpretación de las imágenes médicas se reserva normalmente a la radiología y las subdisciplinas médicas relevantes en la enfermedad médica o área de la ciencia médica bajo investigación.

2.1. Las imágenes DICOM

DICOM (*Digital Imaging and COmmunication in Medicine*) es un estándar reconocido mundialmente para el intercambio de imágenes médicas para el almacenamiento, manipulación, impresión y transmisión de imágenes médicas. Nació como un acuerdo entre la ACR² (*American College of Radiology*) y la NEMA³ (*National Electrical Manufacturers Association*) ante la necesidad inminente de interconectar distintos aparatos de adquisición de imagen radiológica, ya que en aquel momento cada equipo de adquisición contaba hasta entonces con su propio protocolo propietario.

¹http://es.wikipedia.org/wiki/Imagen_m%C3%A9dica

²<http://www.rheumatology.org/>

³<http://www.nema.org/>

2.1.1. Historia

En 1983, el ACR y la NEMA formaron un comité cuya misión era diseñar y desarrollar una interfaz entre el equipamiento existente y cualquier otro dispositivo que el usuario quisiera conectar. Además de las especificaciones para la conexión del *hardware*, el estándar sería desarrollado para permitir además la inclusión de un diccionario de los elementos de datos necesarios para la interpretación y la manipulación de imágenes. Debido a todo ello, en 1985 surgió la primera versión del estándar y tres años después se lanzó la segunda.

El principal problema de esta nueva versión era que los usuarios requerían una interfaz entre los distintos dispositivos, y una red, el protocolo de la cual no poseía la robustez necesaria para soportar las comunicaciones necesarias. Este problema propició el posterior rediseño del proceso en su totalidad, dando lugar a la tercera versión del estándar, el DICOM 3.0; cuya división en capas podemos ver en la figura (2.1).

Con la aparición de los ordenadores y la tecnología de la imagen digital (TAC, Radiología Digital, PET, SPECT, . . .) fueron desarrollados diversos sistemas con la intención de integrar el historial clínico del paciente y las diferentes pruebas que se le hubieran desarrollado para contribuir a un diagnóstico más aproximado.

Estos desarrollos desembocaron en lo que hoy se conoce como PACS (*Picture Archiving and Communication Systems*), sistemas informáticos que aportan nuevos modos de trabajo a la radiología diagnóstica. Tienen por objetivo final el de permitir el funcionamiento de un servicio de radiología integrando las imágenes y la información clínica. Constan de un sistema central de gestión y archivo, y de diferentes sistemas de adquisición, visualización y archivo de imágenes, unidos por redes de comunicaciones. El problema de interconexión entre éstos equipos de naturaleza heterogénea quedaba solventado así gracias a la tercera versión del estándar DICOM.

2.1.2. Estructura

El formato de un fichero DICOM es muy complejo, debido a la gran cantidad de campos que se especifican en la cabecera, así como los diferentes tipos de cabecera que permite, y la multitud de formatos en los que puede estar grabada la imagen. El fichero DICOM se puede dividir en:

1. Un preámbulo y prefijo identificativo del fichero.
2. Una meta-cabecera.
3. Una cabecera.

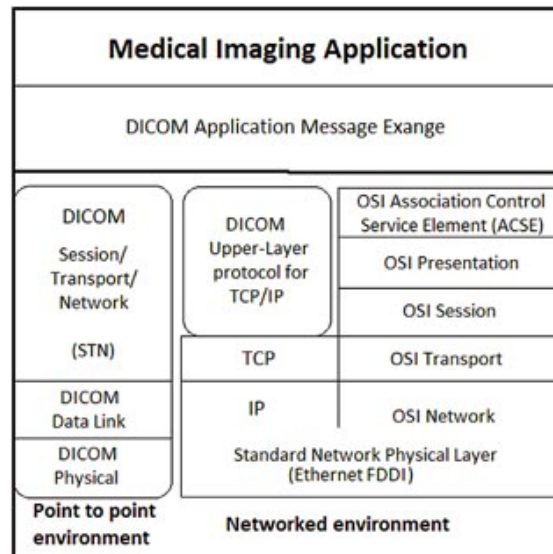


Figura 2.1: Capas DICOM.

4. La imagen propiamente dicha (un elemento más de la cabecera según el punto de vista de la cabecera).

Preámbulo

El estándar DICOM especifica que un fichero en formato DICOM ha de comenzar necesariamente con un preámbulo. Éste tiene un tamaño fijo de 128 bytes, y su uso es dependiente de la implementación. Tampoco está especificada la manera en la que los datos han de ser estructurados, delegando ésta decisión a de los encargados del diseño de la implementación.

En caso de que no se haga uso de él, debe estar presente con todos sus bytes puestos al valor 00h.

Prefijo

Se designa *prefijo identificativo* a aquel conjunto de datos que sigue al preámbulo.

Este prefijo consta de cuatro bytes que contienen la cadena de caracteres DICOM. Esta cadena debe estar codificada siempre con las letras en mayúscula, y usando el conjunto de caracteres especificados en la ISO 8859G0. El propósito de dicho prefijo es permitir a las implementaciones diferenciar si un fichero está o no en formato DICOM.

Elementos de datos

El resto de elementos (cabecera y meta-cabecera) consisten en una serie de campos con toda la información sobre la imagen, incluyendo a ésta.

En estos campos se encuentra información de muy distinta naturaleza; aunque los más interesantes y valor añadido poseen, desde el punto de vista técnico, son aquellos que contienen información para el procesado y la visualización de la imagen.

Al conjunto de la información codificada sobre un campo se le conoce con el nombre de **Elemento de Datos** o Data Element. A continuación se expone cómo se codifican estos *Elementos de Datos*, paso previo para la descripción posterior de la cabecera y la meta-cabecera. Un *Elemento de Datos* está definido por los siguientes campos:

- Etiqueta del Elemento de Datos (*Data Element Tag*): Su misión es la de identificar cada elemento de datos de forma unívoca. Una etiqueta está constituida por un *Número de Grupo* (*Group Number*) y un *Número de Elemento* (*Element Number*). En la documentación del estándar están las descripciones de todos los Elementos de Datos, ordenados según ésta etiqueta. Asimismo se explica el propósito de cada uno de ellos y su requerida obligatoriedad o no. Suelen ser representados como un vector de dos dimensiones, en cuya primera dimensión se encuentra el *Número de Grupo* y en la segunda el *Número de Elemento*, en hexadecimal, mediante cuatro dígitos.
- Representación del Valor (*Value Representation, VR*): Indica la forma en que se codifica el valor del elemento. Este campo no siempre está codificado en un *Elemento de Datos*, sino que depende de la sintaxis de transferencia.
- Longitud del Valor (*Value Length*): Entero que se corresponde con la longitud del campo *Valor*.
- Valor (*Value*): es el valor del elemento de datos codificado según el campo *VR* y con la longitud que indica el campo *Longitud del Valor*.

I. Campos

Todos los campos definidos por DICOM se encuentran listados en una base de datos que se encuentra en el documento número seis del estándar, y se la conoce como *Registro de los Elementos de Datos DICOM* (*Registry of DICOM Data Elements*).

Cada elemento está indexado por su etiqueta (*Número de Elemento* y *Número de Grupo*), y para cada uno de ellos se halla especificado:

- **Nombre:** Nombre del elemento y pequeña descripción de su función.

- **VR:** Representación del valor de cada elemento.
- **VM:** Cantidad de valores del mismo tipo que puede contener el campo *Valor* del elemento de datos.
- También, en caso de que el un elemento de datos esté obsoleto y haya sido retirado en una versión actual del estándar, poseerá el identificador RET.

DICOM establece la obligatoriedad de cada uno de sus campos mediante una clasificación basada en tipos.

- **Tipo 1:** Este tipo es de inclusión obligatoria. La longitud del campo no puede ser cero, y debe tener un valor válido.
- **Tipo 1C:** Tipo de inclusión obligatoria siempre que se den ciertas condiciones. Si éstas tienen lugar, el elemento es, a todos los efectos, perteneciente al grupo 1.
- **Tipo 2:** Este tipo también es de inclusión obligatoria, con la salvedad que puede tener una longitud de campo igual a cero y sin campo *Valor*. Ésto último solo tiene lugar bajo varios supuestos específicos.
- **Tipo 2C:** Análogamente al Tipo1, existe un Tipo 2C, que equivale al Tipo 2 solo bajo la existencia de ciertas condiciones.
- **Tipo 3:** Este tipo de campos es opcional y carente de las restricciones de los otros tipos.

II. Representación del valor

El estándar DICOM define una serie de VR con diferentes características, con la intención de que el campo *Valor* de cada *Elemento de Datos* esté codificado correctamente según aquello que represente.

El listado con las descripciones resumidas puede ser encontrado en las tablas (2.1) y (2.2).

Nombre VR	Descripción	Longitud
Application Entity (AE)	Cadena que identifica una Entidad de Aplicación.	16 Bytes
Age String (AS)	Cadena de caracteres que expresa la fecha.	4 Bytes
Attribute Tag (AT)	Par ordenado de enteros sin signo de 16 bits.	
Code String (CS)	Cadena de caracteres, siendo los espacios no significativos.	16 Bytes
Date (DA)	Cadena de caracteres con formato yyymmdd (año-mes-día).	8 Bytes
Decimal String (DS)	Cadena de caracteres que representa un número en coma fija o flotante.	16 Bytes
Date Time (DT)	Fecha.	26 Bytes
Floating Point Single (FL)	Número en coma flotante de simple precisión.	4 Bytes
Floating Point Double (FD)	Número en coma flotante.	8 Bytes
Integer String (IS)	Cadena de caracteres que respresenta un entero en base decimal.	12 Bytes
Long String (LS)	Cadena de caracteres.	64 caracteres
Long Text (LT)	Cadena de caracteres que puede contener uno o más párrafos.	10240 caracteres
Other Byte String (OS)	Cadena de bytes. Significado dependiente de la sintaxis de transferencia.	<i>Depende de la sintaxis</i>
Other Float String (OF)	Cadena de números en coma flotante de simple precisión.	

Tabla 2.1: Valores de los campos (1/2).

Nombre VR	Descripción	Longitud
Other Word String (OW)	Cadena de palabras de 16 bits. Significado dependiente de la sintaxis de transferencia.	<i>Depende de la sintaxis</i>
Person Name (PN)	Cadena de caracteres de cinco componentes: nombre1, nombre2, apellido, prefijo y sufijo.	64 caracteres
Short String (SH)	Cadena de caracteres.	16 caracteres
Signed Long (SL)	Entero con signo de 32 bits en complemento a dos.	4 Bytes
Sequence of Items (SQ)	Secuencia de ítems	
Signed Short (SS)	Entero con signo de 16 bits en complemento a dos.	2 Bytes
Short Text (ST)	Cadena de caracteres de uno o más párrafos.	1024 caracteres
Time (TM)	Cadena de caracteres con el formato <i>hh:mm:ss</i> . Formato 24 horas	16 Bytes
Unique Identifier (UI)	Cadena de caracteres que representa un identificador único.	64 Bytes
Unsigned Long (UL)	Entero sin signo de 32 bits.	4 Bytes
Unknown (UN)	Cadena de bytes con significado desconocido.	
Unsigned Short (US)	Entero sin signo de 16 bits.	2 Bytes
Unlimited Text (UT)	Cadena de caracteres con uno o más párrafos.	

Tabla 2.2: Valores de los campos (2/2).

III. Codificación

Existen dos tipos de codificación para los elementos *VR Implícita* y *VR Explícita*.

La primera se usa para la *Sintaxis de Transferencia* por defecto, y la diferencia más importante reside en que la VR de cada *Elemento de Datos* no se codifica en un fichero, haciendo más difícil su decodificación y obligando a recurrir a la documentación para obtenerla.

Sintaxis de transferencia

Se refiere a la manera en que están descritos los *Elemento de Datos*, la cual generalmente es la misma para todos los elementos contenidos en un mismo archivo. Ésta sintaxis determina bajo qué ordenamiento está escrito cada *Elemento de Datos*.

Concretamente, la sintaxis está especificada mediante un identificador único (UID) que describe la forma en que se va a codificar la cabecera, o la cabecera y los datos de la imagen; en caso de que los datos estén codificados en algún formato encapsulado.

La sintaxis de transferencia determina si el *Elemento de Datos* está escrito en un ordenamiento *Big* o *Little Endian*, si el valor de presentación está o no en el elemento (valor de representación explícito o implícito) y el tipo de compresión de la imagen. Así el estándar DICOM define veintidós sintaxis de transferencia, las cuales se conocen como estándar. A pesar de las diferentes opciones definidas para la escritura de dichos archivos, algunos fabricantes definen sintaxis de transferencia privadas, es decir, conservan la misma estructura de los archivos DICOM, pero con especificaciones diferentes para sus *Elementos de Datos*. Estos formatos se conocen como formatos DICOM compatibles, no estándar, o directamente, como formatos privados.

A modo de ejemplo, a continuación se exponen algunas de las sintaxis existentes.

Cabecera

La cabecera consta de un *Data Set* con aquella información necesaria codificada según la sintaxis de transferencia mencionada en la sección anterior. En la cabecera de un fichero DICOM es donde se proporciona la mayor parte de la información necesaria para la correcta visualización de las imágenes. Cabe recordar que los datos de la imagen son considerados una parte más de la cabecera. Los tipos de cabecera son dependientes de la *Sintaxis de Transferencia* utilizada y, por ello, existen cuatro tipos de cabeceras; tres para formatos no encapsulados y una para

Sintaxis de transferencia	Identificador único	Descripción
Implicit VR Little Endian	1.2.840.10008.1.2	Sintaxis de transferencia por defecto
Explicit VR Little Endian	1.2.840.10008.1.2.1	Se especifica la <i>Representación del Valor</i> de cada elemento de la cabecera usando codificación <i>Little Endian</i> .
Explicit VR Big Endian	1.2.840.10008.1.2.2	Se especifica la <i>Representación del Valor</i> de cada elemento de la cabecera usando codificación <i>Big Endian</i> .

Tabla 2.3: Sintaxis de transferencia más usuales. La descripción técnica de todas ellas puede ser encontrada en [16]

imágenes en formato encapsulado. La explicación en detalle de cada una de ellas excede de este estudio y puede ser encontrada en [16].

Capítulo 3

Técnicas de interpolación no lineales

3.1. Introducción

El problema típico en teoría de aproximación consiste en reconstruir una función a partir de un conjunto discreto de datos. Para abordar este problema podemos partir de dos puntos de vista conocidos que son la interpolación de valores puntuales y la interpolación mediante medias en celda.

El primer punto de vista, es aplicable cuando se dispone de un conjunto de valores que toma una función $f(x)$ desconocida en una discretización dada, para obtener otra que coincida con $f(x)$ en los nodos de la discretización anterior. El segundo, de manera análoga al primero, se aplica cuando aquello que se conoce son las medias de $f(x)$ en una discretización concreta, y el objetivo es encontrar una nueva función que posea las mismas medias que $f(x)$ en la citada discretización.

Ambas técnicas persiguen el mismo objetivo, y su elección en el momento de aplicar una y otra, depende puramente de la naturaleza de los datos de partida. De este modo, la nueva función dada puede ser aproximada por un polinomio que posea los mismos valores en el conjunto de puntos, o las mismas medidas en los intervalos de dicha función.

Las técnicas interpolatorias lineales, independientes de los datos, son frecuentemente empleadas para reconstruir funciones a las que los métodos lineales no consiguen aproximarse con la calidad deseada. El principal problema es que el orden de aproximación se ve reducido cuando el conjunto de puntos empleados para construir la función interpolante contiene alguna singularidad. Visualmente, estos errores se manifiestan de tres formas distintas: el *suavizado de los contornos*, el *efecto escalera* en los contornos oblicuos y la aparición del fenómeno *rizado*[5].

- El **suavizado** se produce porque los contornos, más concretamente aquellos que tienen una gran agudeza, pueden considerarse localmente como escalones bidimensionales, es decir, señales no suaves y de ancho de banda infinito.

- El **efecto escalera** se debe a una ineficiente eliminación de las repeticiones del espectro que introduce el insertador de ceros.
- El **rizado** se corresponde con oscilaciones en la amplitud de la imagen interpolada que no están presentes en la imagen original. La razón de ello es que la función interpolante no decrece de forma monótona a medida que $|t|$ crece, sino que presenta oscilaciones que, en general, decrecen a medida que nos alejamos del origen.

La multiresolución de Harten es una herramienta para el procesamiento de imágenes. El objetivo de ésta técnica es establecer un marco para llevar a cabo las transformaciones entre distintos niveles de multiresolución, utilizando una serie de operadores. Estos operadores están íntimamente relacionados con la reconstrucción y la discretización de la función objeto de estudio, y permiten conectar diferentes niveles discretos de resolución con un espacio funcional adecuado, el cual es dependiente de las aplicaciones. Es en el operador reconstrucción el que adquiere más importancia en nuestro caso, porque será el que implemente alguna técnica de interpolación no lineal que determinará la calidad de la aproximación empleada.

En una primera aproximación, podríamos pensar que dichas singularidades se verían resueltas aumentando el orden de la función, pero, en caso de implementarlo, se puede ver que la discontinuidad acaba por afectar a un mayor número de conjuntos de puntos, denominados *stencils*, aumentando una zona de la función o de la imagen en este caso, donde la calidad no es óptima.

El punto crucial radica en la selección de los nodos adecuados para construir el interpolante, de modo que el conjunto de puntos elegido no contenga ninguna singularidad. Un primer acercamiento a la solución del problema implicaría la utilización del algoritmo ENO, en el cual el polinomio interpolado se construye tomando información de las zonas donde la función interpolada es suave. De esta manera, si las singularidades de la función están lo suficientemente aisladas, es posible reducir la zona donde la aproximación se ve degradada, al intervalo que lo contiene. Si se conoce la localización exacta de la singularidad, se puede acotar la pérdida de exactitud a un entorno alrededor de la singularidad mediante el algoritmo ENO-SR.

Seguidamente, la técnica interpolatoria WENO constituye a priori una mejora notable de la técnica ENO. Consiste en construir la función interpolante mediante combinaciones convexas de todas las aproximaciones obtenidas a partir de *stencils* que contienen el intervalo a interpolar, de modo que en la combinación se priman las aproximaciones de aquellos puntos de zonas suaves y si la función es suave en todos ellos, se obtiene una aproximación de orden óptimo.

A continuación se presentarán las técnicas Racional y PPH. La primera se considera como una modificación de la técnica WENO con una selección particular

de los pesos; mientras que la segunda, se detalla como una interpolación con idénticos resultados que la interpolación lineal en regiones suaves, y con resultados aceptables en regiones en las cuales se halla alguna singularidad presente.

3.2. Multirresolución de Harten

La multirresolución de Harten es una herramienta muy eficaz para el procesamiento de imágenes. El objetivo de ésta técnica es obtener una reordenación multiescala de la información contenida en un conjunto de datos discretos; y el resultado puede ser interpretado como una aproximación de la información inicial en un nivel de resolución menor, más unos detalles que en principio nos permiten recuperar datos iniciales.

Formalmente partimos de un espacio V^k , en el que k indica el nivel de resolución, y de una función f perteneciente a dicho espacio. Un mayor valor de k indica un mayor nivel de resolución. En el caso que nos ocupa, la reconstrucción mediante valores puntuales, podemos considerar que los datos discretos son valores puntuales en una malla dada.

La multirresolución se apoya en los operadores decimación y predicción que permiten la transición entre dos niveles consecutivos de resolución. Ambos se definen como sigue:

- **Decimación:** Proporciona información discreta a un nivel de resolución $k-1$, a partir de un nivel de resolución k . Formalmente se denota por $D_k^{k-1} : V_k \rightarrow V_{k-1}$.
- **Predicción:** Es el operador que proporciona una aproximación discreta del nivel k a partir de la información contenida en un nivel $k+1$ y al que además no se le exige que sea lineal. Siguiendo la notación, se denotará como $P_{k-1}^k : V_{k-1} \rightarrow V_k$.

Los datos discretos se obtienen a partir de la discretización de una función f , para lo cual existen distintos tipos de operadores. Dependiendo del operador discretización utilizado, la secuencia de datos f^k es diferente. El objetivo del enfoque propuesto por Harten es la construcción de esquemas multirresolución adaptados a cada proceso de discretización. Esto se consigue definiendo un operador reconstrucción apropiado. Estos dos últimos operadores, son los elementos a partir de los cuales se construyen los operadores de decimación y predicción del esquema de multirresolución. Para entender la terminología empleada por los distintos operadores convendría consultar el ejemplo 3.2.

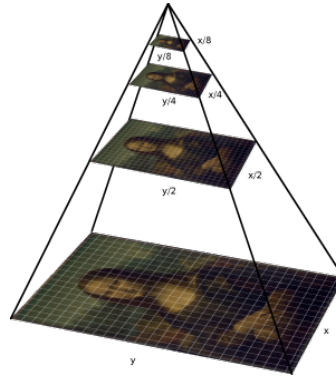


Figura 3.1: Ilustración del concepto «píramide de resolución.»

Para definir estos operadores formalmente, consideremos \mathcal{F} un espacio de funciones:

$$\mathcal{F} \subset \{f \mid f : \Omega \subset \mathbb{R}^m \longrightarrow \mathbb{R}\}$$

Se define el operador **discretización**, \mathcal{D}_k , como aquél operador que asigna a cada elemento de este espacio, $f \in \mathcal{F}$, una secuencia de f^k de datos discretos perteneciente al espacio V^k . De este modo el operador discretización:

$$\mathcal{D}_k : \mathcal{F} \rightarrow V^k = \mathcal{D}_k(f)$$

que ha de ser lineal y sobreyectivo, y que a cada $f \in \mathcal{F}$ le asocia:

$$f^k = \mathcal{D}_k(f)$$

La **reconstrucción** ofrece la equivalencia en sentido inverso, tomando una secuencia de datos discretos para reconstruir, a partir de la información proporcionada por dichos datos, la función de la cual provienen:

$$\mathcal{R}_k : V^k \rightarrow \mathcal{F}$$

A este operador no se le exige que sea lineal, ésta es la principal novedad introducida por Harten.

Los operadores decimación y reconstrucción deben verificar una condición de consistencia, la cual pretende asegurar que la reconstrucción de un conjunto discreto de datos contenga exactamente la misma información que el conjunto inicial de datos.

$$\mathcal{D}_k \mathcal{R}_k v^k, \forall v^k \in V^k, \text{ es decir } \mathcal{D}_k \mathcal{R}_k = I_{V^k}. \quad (3.1)$$

$$\begin{aligned}
X^k &= \begin{bmatrix} \odot & \odot & \odot & \odot & \odot \\ \odot & \odot & \odot & \odot & \odot \\ \odot & \odot & \odot & \odot & \odot \\ \odot & \odot & \odot & \odot & \odot \end{bmatrix} \\
X^{k+\frac{1}{2}} &= \begin{bmatrix} \odot & \otimes & \odot & \otimes & \odot & \otimes & \odot & \otimes & \odot \\ \odot & \otimes & \odot & \otimes & \odot & \otimes & \odot & \otimes & \odot \\ \odot & \otimes & \odot & \otimes & \odot & \otimes & \odot & \otimes & \odot \\ \odot & \otimes & \odot & \otimes & \odot & \otimes & \odot & \otimes & \odot \end{bmatrix} \\
X^{k+1} &= \begin{bmatrix} \odot & \otimes & \odot & \otimes & \odot & \otimes & \odot & \otimes & \odot \\ \oplus & \star & \oplus & \star & \oplus & \star & \oplus & \star & \oplus \\ \odot & \otimes & \odot & \otimes & \odot & \otimes & \odot & \otimes & \odot \\ \oplus & \star & \oplus & \star & \oplus & \star & \oplus & \star & \oplus \\ \odot & \otimes & \odot & \otimes & \odot & \otimes & \odot & \otimes & \odot \\ \oplus & \star & \oplus & \star & \oplus & \star & \oplus & \star & \oplus \\ \odot & \otimes & \odot & \otimes & \odot & \otimes & \odot & \otimes & \odot \\ \oplus & \star & \oplus & \star & \oplus & \star & \oplus & \star & \oplus \\ \odot & \otimes & \odot & \otimes & \odot & \otimes & \odot & \otimes & \odot \end{bmatrix}
\end{aligned}$$

Figura 3.2: Representación de tres niveles de resolución, X^k , $X^{k+\frac{1}{2}}$ y X^{k+1} respectivamente.

Se parte de un nivel de resolución k con el conjunto de puntos $\{x_i^k, y_j^k\}_{i,j=0}^{J_k}$. Mediante la técnica del *producto tensor* se obtienen los valores interpolados para las nuevas filas, es decir los valores para $\{x_i^{k+\frac{1}{2}}, y_j^k\}_{0 \leq i \leq J_{k+1}, 0 \leq j \leq J_k}$ correspondientes al nivel de resolución $k + \frac{1}{2}$. Por último, se obtienen los valores para las columnas alcanzando el nivel de resolución $k + 1$, es decir, con el conjunto de puntos $\{x_i^{k+1}, y_j^{k+1}\}_{i,j=0}^{J_{k+1}}$.

En la figura se aprecian los píxeles originales (\odot), los detalles interpolados verticales (\otimes), los interpolados horizontales (\oplus) y los mixtos (\star).

Dada una secuencia de operadores discretización $\{\mathcal{D}_k\}$, y de reconstrucción $\{\mathcal{R}_k\}$ se definen los operadores decimación y predicción de la siguiente manera:

$$\begin{aligned} D_k^{k-1} &= \mathcal{D}_{k-1}\mathcal{R}_k, \\ P_{k-1}^k &= \mathcal{D}_k\mathcal{R}_{k-1}. \end{aligned}$$

Como se observa, se satisface la restricción de consistencia establecida en (3.1).

Se dice que una sucesión de operadores discretización, \mathcal{D}_k , es anidada si cumple la siguiente expresión Aunque a priori parezca que el operador decimación dependa de la elección del operador reconstrucción, diremos que una operación de discretización es *ennidada* si verifica:

$$\mathcal{D}_k f = 0 \Rightarrow \mathcal{D}_{k-1} f = 0, \forall f \in \mathcal{F} \quad (3.2)$$

Esta propiedad significa que la información contenida en los datos a un cierto nivel de resolución k no será nunca mayor que la información contenida en un nivel de resolución superior. En caso de que se cumpla esta propiedad, se tendrá la garantía de que el operador decimación será independiente del operador reconstrucción. A modo de demostración, si consideramos dos secuencias de operadores reconstrucción, \mathcal{D}_k y \mathcal{D}'_k que verifican la ecuación (3.1), se tiene:

$$\mathcal{D}_{k-1}(\mathcal{R}_k v^k - \mathcal{R}'_k v^k) = \mathcal{D}_k \mathcal{R}_k v^k - \mathcal{D}_k \mathcal{R}'_k v^k = v^k - v^k = 0, \forall v^k \in V^k. \quad (3.3)$$

Verificando que ambos operadores son independientes, como sigue:

$$\mathcal{D}_{k-1}(\mathcal{R}_k v^k - \mathcal{R}'_k v^k) = 0 \Rightarrow \mathcal{D}_{k-1} \mathcal{R}_k v^k = \mathcal{D}_{k-1} \mathcal{R}'_k v^k, \forall v^k \in V^k. \quad (3.4)$$

A partir de las definiciones (3.2), (3.3) y (3.4) se deduce la relación de consistencia para los operadores decimación y predicción, análogamente a la ecuación (3.1). Si decimamos la información obtenida a partir de la predicción realizada sobre una información con resolución dada por V^{k-1} , obtenemos exactamente la misma información de partida, sin haber introducido ningún elemento nuevo.

$$D_k^{k-1} P_{k-1}^k = \mathcal{D}_{k-1} \mathcal{R}_k \mathcal{D}_k \mathcal{R}_{k-1} = \mathcal{D}_{k-1} \mathcal{R}_{k-1} = I_{V^{k-1}} \quad (3.5)$$

Si denotamos por v^k a aquella información discreta en un nivel de resolución k , al aplicarle el operador decimación sobre ella, obtenemos v^{k-1} , es decir, la información contenida en el nivel de resolución $k - 1$:

$$v^{k-1} = D_k^{k-1} v^k$$

Dado que $P_{k-1}^k D_k^{k-1} v^k$ constituye una aproximación a v^k , el error queda definido como sigue:

$$e^k = v^k - P_{k-1}^k D_k^{k-1} v^k = (I_V^k - P_{k-1}^k D_k^{k-1}) v^k = Q_k v^k \in V^k.$$

De esta manera, conocido $v^{k-1} D_k^{k-1} v^k \in V^k$ y e^k puede ser recuperado v^k , conteniendo la misma información tanto el conjunto v^k como el conjunto $\{v^{k-1}, e^k\}$, es decir:

$$v^k \equiv \{v^{k-1}, e^k\} \quad (3.6)$$

haciendo obvia la siguiente relación $v^k = P_{k-1}^k v^k + e^k$.

El problema es que siguiendo este procedimiento se tiene información redundante, pues si suponemos V^k es un espacio de dimensión finita, $\dim V^k = N_k$, resulta que $\{v^{k-1}, e^k\}$ consta de $N_{k-1} + N_k$ elementos, aun conteniendo $\{v^{k-1}, e^k\}$ y v^k la misma información.

Esta información redundante puede ser eliminada, como sigue:

$$\begin{aligned} D_k^{k-1} e^k &= D_k^{k-1} (I_V^k - P_{k-1}^k D_k^{k-1}) v^k \\ &= D_k^{k-1} v^k - D_k^{k-1} P_{k-1}^k D_k^{k-1} v^k \\ &= D_k^{k-1} v^k - D_k^{k-1} v^k = 0. \end{aligned}$$

es decir, $e^k \in N(D_k^{k-1}) = \{v^k \in V_k : D_k^{k-1} v^k = 0\}$ cuya dimensión es $\dim N(D_k^{k-1}) = \dim V^k - \dim V^{k-1} = N_k - N_{k-1}$.

Sea μ_i^k el conjunto definido por los elementos que generan el espacio $N(D_k^{k-1})$. Entonces el error e^k se define como $e^k = \sum d_i^k \mu_i^k$.

Si definimos G_k como el operador que a cada elemento de $e^k \in N(D_k^{k-1})$ asocia un elemento del conjunto de coeficientes d_i^k correspondientes a la base μ_i^k ; y sea E_k el operador que dada una serie de coeficientes d_i^k les asocia $\sum_i d_i^k \mu_i^k$, se establece la equivalencia siguiente:

$$v^k \equiv \{v^{k-1}, d^k\} \quad (3.7)$$

donde ahora ambos conjuntos tienen igual cantidad de elementos, pues el número de elementos de f^{k-1, d^k} será igual a $\dim F^{k-1} + \dim N(D_k^{k-1}) = N_{k-1} + (N_k - N_{k-1}) = N_k = \dim V^k$. Destacar que mediante las siguientes expresiones queda definida la equivalencia entre f^k y $f^k - 1$:

$$\begin{aligned} v^k - 1 &= D_k^{k-1} v^k, \\ d^k &= G_k (I - P_{k-1}^k D_k^{k-1}) v^k, \end{aligned}$$

y el paso contrario, mediante

$$v^k = P_{k-1}^k v^{k-1} + E_k d^k$$

relación extraída directamente a partir de la equivalencia $e^k = E_k d^k$.

Mediante la equivalencia anterior (3.6) se obtiene la descomposición multiescala de v^k . Por ejemplo si consideramos que los datos originales parten de un nivel L de resolución, se tiene:

$$v^k \equiv \{v^0, d^1, \dots, d^1\}$$

$$v^L \xrightarrow{d^L} v^{L-1} \xrightarrow{d^{L-1}} v^{L-2} \xrightarrow{\dots}$$

Y los correspondientes algoritmos para obtener la transformación multiescala y su paso inverso, son los siguientes:

Algoritmo de transformación directa

$$v^L \rightarrow Mv^L = \{v^0, d^1, \dots, d^L\} = \begin{cases} \text{Hacer } k = L, \dots, 1 \\ v^k - 1 = D_k^{k-1} f^k \\ d^k = G_k(v^k - P_{k-1}^k v^{k-1}) \end{cases}$$

Algoritmo de transformación inversa

$$Mv^L \rightarrow M^{-1}Mv^L = \begin{cases} \text{Hacer } k = L, \dots, 1 \\ v^k = P_{k-1}^k v^{k-1} + E_k d^k \end{cases}$$

Llegados a este punto, es obvio que el paso crucial en la construcción de un esquema de multirresolución es la definición de un operador reconstrucción apropiado para la discretización que se esté considerando. De ello dependerán tanto la calidad final de la imagen como el coste computacional total del proceso de ampliación. Habitualmente se utilizan dos tipos de reconstrucción en la multirresolución de Harten, y son la discretización por valores puntuales y la discretización por medias en celda.

A continuación se exponen una serie de algoritmos aptos para ser implementados como operador reconstrucción, todos ellos a partir de valores puntuales debido a la naturaleza de los datos de entrada (i.e. un conjunto de píxeles).

3.3. Métodos de interpolación no lineales

Los algoritmos de *zoom* que se exponen a continuación han sido definidos para secuencias de datos dos dimensionales. La estrategia llevada a cabo por las diversas técnicas es la de *producto tensor* la cual se describe a continuación. Sea f un array bidimensional definido como $f = (f_{i,j}^0)_{(i,j)=0}^{J_0}$ al que denotamos como A , y donde $A = A^0$. La estrategia seguida en las sucesivas técnicas consiste en aplicar el proceso

de *zoom* primero sobre las filas y a continuación sobre las columnas, de manera independiente, en contraposición a aquellos algoritmos que actúan directamente de manera bidimensional sobre los datos, esto es, seleccionando un stencil de más de una dimensión.

3.3.1. Interpolación ENO

La interpolación ENO (*Essentially Non Oscillatory*) tiene como objetivo construir trozos o partes de polinomios usando, en la medida de lo posible, datos pertenecientes a regiones suaves de una función. El punto clave de esta técnica interpolatoria es el proceso por el cual se selecciona el *stencil* que se intenta elegir dentro de una región suave de una función dada, $f(x)$, esto es, f es infinitamente diferenciable en todos sus órdenes. Este proceso de selección trabaja de la manera siguiente: para cada intervalo $I_i = [x_{j-1}, x_j]$, se consideran todos los posibles conjuntos con $r \geq 2$ puntos, incluyendo los puntos x_{j-1}^k, x_j^k . Después de seleccionar el stencil según alguno de los dos métodos de selección que a continuación veremos, el stencil ENO, queda de la siguiente manera:

$$S^{ENO} = \left\{ x_{s_j-1}^k, \dots, x_{s_j+r-1}^k \right\}$$

siendo $r + 1$ el orden de interpolación.

Para la selección de dicho stencil existen dos estrategias. Ambas producen asintóticamente conjuntos de puntos de interpolación que se mueven lejos de la discontinuidad. Consecuentemente, el orden de aproximación del operador de predicción ENO sigue siendo $r + 1$ siempre que sea posible evitar dichas discontinuidades.

Algoritmos para la selección del *stencil*

1. **Selección jerárquica** Básicamente consiste en, partiendo de los extremos del intervalo, ir añadiendo progresivamente puntos a derecha o izquierda del mismo, comparando las diferencias divididas correspondientes a los conjuntos formados por los extremos del intervalo, más los puntos añadidos, y escogiendo aquella de menor valor absoluto.
2. **Selección no jerárquica** Esta selección, por contra, considera las diferencias divididas de mayor orden correspondientes a todos los *stencils* posibles y calcula el mínimo entre todos los valores absolutos de dichas diferencias.

Tanto si empleamos el primero como el segundo, los nodos x_{i-1}, x_{i+1} pertenecen al *stencil* S^{ENO} . En caso de que f tenga alguna discontinuidad de salto en $x_d \in I_i$, y sean S un *stencil* que no cruza dicha discontinuidad y S^* un stencil conteniendo a los nodos x_{i-1} y x_i , ambos con $s + 1$ nodos. Tenemos entonces:

$$f[S] = O(1); \quad f[S^*] = O\left(\frac{1}{h^2}\right)$$

Y si la discontinuidad pertenece a la primera derivada, obtenemos

$$f[S] = O(1); \quad f[S^*] = O\left(\frac{1}{h^{s-1}}\right)$$

Algoritmo 1 Selección jerárquica del *stencil* ENO

Entrada: \vec{f}, r \vec{f} - Vector de datos. r - Orden de la reconstrucción.**Salida:** \mathcal{S} \mathcal{S} - *Stencil* seleccionado para formar el polinomio interpolador.

```

1: for  $i = 1, \dots, J$  do
2:    $s_0 = i$ 
3:   for  $l = 0, \dots, r - 2$  do
4:     si  $|f[x_{s_l-2}, \dots, x_{s_l+l+1}]| < |f[x_{s_l-1}, \dots, x_{s_l+l+1}]|$  then
5:        $s_{l+1} = s_l + 1$ 
6:     fin si
7:   fin for
8:    $s_i = s_{r-1}$ 
9: fin for

```

Algoritmo 2 Selección no jerárquica del *stencil* ENO

Entrada: \vec{f}, r \vec{f} - Vector de datos. r - Orden de la reconstrucción.**Salida:** \mathcal{S} \mathcal{S} - *Stencil* seleccionado para formar el polinomio interpolador.

```

1: for  $i = 1, \dots, J$  do
2:   elegir  $s_i$  que verifique
3:   for  $l = 0, \dots, r - 2$  do
4:      $|f[x_{s_i-1}, \dots, x_{s_i+r-1}]| < \min \{ |f[x_{l-1}, \dots, x_{l+r-1}]|, i - r + 1 \leq l \leq i \}$ 
5:   fin for
6:    $s_i = s_{r-1}$ 
7: fin for

```

Entonces, en caso de que f solamente presentara discontinuidades en el intervalo I_i , ya que los algoritmos anteriores de selección se alejan de las discontinuidades, los polinomios interpoladores verificarían

$$f(x) = q_l(x) + O(h^{r+1}), \quad x \in [x_{l-1}, x_l], \quad l \leq i-1, \geq i+1$$

obteniendo una aproximación óptima en todos los intervalos excepto en el I_i que contiene la singularidad. De esta manera observamos que, en presencia de discontinuidades aisladas, tanto en la primera derivada como en la propia función, ENO consigue que las zonas donde la aproximación esta afectada por las singularidades quede acotada únicamente a los intervalos que las contienen, utilizando cualquiera de los dos métodos de partición antes expuestos. Por todo ello, $q_l(x) = f(x) + O(h^{r+1})$, $x \in I_l, \forall l$ pues ningún stencil cruza ninguna discontinuidad, consiguiendo una aproximación óptima en el intervalo dado. Particularizando el problema para construir una técnica de interpolación de cuarto orden, esto es, $r = 3$, tenemos:

$$\left\{ \begin{array}{l} \text{Hacer } k = 1, \dots, L \\ f_{2j}^k = f_j^{k-1} \\ f_{2j-1}^k = \begin{cases} 5f_{j-3}^{k-1} + 15f_{j-2}^{k-1} + 5f_{j-1}^{k-1} + f_j^{k-1}, & \text{si } S_j^1 \\ \frac{-f_{j-2}^{k-1} + 9f_{j-1}^{k-1} + 9f_j^{k-1} - f_{j+1}^{k-1}}{16}, & \text{si } S_j^2 \\ 5f_{j-1}^{k-1} + 15f_j^{k-1} + 5f_{j+1}^{k-1} + f_{j+2}^{k-1}, & \text{si } S_j^3 \end{cases} \end{array} \right.$$

Algoritmo

A continuación se sugiere un algoritmo de reconstrucción basado en la técnica ENO.

Algoritmo 3 Interpolación ENO

Entrada: *niv*, *orden*, *im*, *met_particion**niv* - Niveles de zoom.*orden* - Orden del polinomio interpolatorio.*im* - Imagen original sobre la cual aplicar el zoom.*met_particion* - Booleano para el método de partición utilizado. 0 para no jerárquico, 1 para jerárquico.**Salida:** *b**b* - Imagen interpolada.

```

1:  $[n \ m] = \text{size}(a)$ 
2: //Bucle para los niveles de zoom.
3: for  $k = 1$  hasta  $niv$  do
4:    $n\_filas = 2 * n - 1$ 
5:    $n\_columnas = 2 * m - 1$ 
6:    $b = \text{zeros}(n\_filas, n\_columnas)$ 
7:    $b(1 : 2 : n\_filas, 1 : 2 : n\_columnas) = a(1 \dots n, 1 \dots m)$ 
8:   // Predicción de las columnas.
9:   for  $j = 1$  hasta  $n\_columnas$  a incrementos de 2 do
10:     $b(1 \dots n\_filas, j) = \text{eno\_zoom}(b(1:2:n\_filas, j), n, \text{orden}, \text{met\_particion})'$ 
11:   fin for
12:   // Predicción de las filas.
13:   for  $j = 1$  hasta  $n\_filas$  do
14:     $b(i, 1 \dots n\_columnas) = \text{eno\_zoom}(b(i, 1:2:n\_filas), n, \text{orden}, \text{met\_particion})$ 
15:   fin for
16:   //Actualización de las variables.
17:    $n = n\_filas; m = n\_columnas$ 
18:    $a = b$ 
19: fin for

```

Algoritmo 4 ENO Zoom Parte 1/3

Entrada: $v, n, orden, met_particion$ \vec{v} - Vector de datos. n - Longitud del vector \vec{v} . $orden$ - Orden del polinomio interpolatorio. $met_particion$ - Booleano para el método de partición utilizado.**Salida:** f \vec{f} - Vector con los valores interpolados.

```

1:  $\vec{f} = \text{zeros}(1, 2 * n - 1)$ 
2:  $\vec{f}(1 : 2 : 2 * n - 1) = \vec{v}(1 \dots n)$ 
3: //Obtenemos las máscaras correspondientes a cada uno de los stencils posibles
   a elegir.
4:  $m_1 = \text{getMask}(orden/2, orden/2)$ 
5:  $m_2 = \text{getMask}(orden/2 - 1, orden/2 + 1)$ 
6:  $m_3 = \text{getMask}(orden/2 + 1, orden/2 - 1)$ 
7: //Predicción del primer elemento con máscara lineal.
8:  $\vec{v}' = \vec{v}[v_1 \dots v_{orden}]$ 
9:  $f_2 = \vec{m}_2 \times \vec{v}'$ 
10: si  $met\_particion == 0$  then
11:    $\vec{v}' = \vec{v}[v_1 \dots v_4]$ 
12:    $f_2 = \vec{m}_1 \times \vec{v}'$ 
13:   si  $|(v_3 - v_2) - (v_2 - v_1)| \leq |(v_4 - v_3) - (v_3 - v_2)|$  then
14:      $\vec{v}' = \vec{v}[v_1 \dots v_4]$ 
15:      $f_4 = \vec{m}_1 \times \vec{v}'$ 
16:   else si  $|(v_4 - v_3) - (v_3 - v_2)) - ((v_3 - v_2) - (v_2 - v_1))| \leq |((v_5 - v_4) - (v_4 -$ 
    $v_3)) - ((v_4 - v_3) - (v_3 - v_2))|$  then
17:      $\vec{v}' = \vec{v}[v_1 \dots v_4]$ 
18:      $f_4 = \vec{m}_1 \times \vec{v}'$ 
19:   else
20:      $\vec{v}' = \vec{v}[v_2 \dots v_5]$ 
21:      $f_4 = \vec{m}_2 \times \vec{v}'$ 
22:   fin si
23:   for  $i = 6$  hasta  $2 * n - 6$  a incrementos de 2 do
24:     //Véase algoritmo 2
25:      $s = \text{selNoJerarquicaEno}(\vec{v}, orden)$ 

```

Algoritmo 5 ENO Zoom Parte 2/3

```

26:   si  $s \in S_1$  then
27:        $\vec{v}' = \vec{v}[v_{\frac{i}{2}-1} \dots v_{\frac{i}{2}+2}]$ 
28:        $f_j = \vec{m}_1 \times \vec{v}'$ 
29:   else si  $s \in S_2$  then
30:        $\vec{v}' = \vec{v}[v_{\frac{i}{2}-2} \dots v_{\frac{i}{2}+1}]$ 
31:        $f_j = \vec{m}_3 \times \vec{v}'$ 
32:   else
33:
34:        $\vec{v}' = \vec{v}[v_{\frac{i}{2}} \dots v_{\frac{i}{2}+3}]$ 
35:        $f_j = \vec{m}_2 \times \vec{v}'$ 
36:   fin si
37: fin for
38: //Predicción del penúltimo elemento.
39: si  $|v[v_{n-orden}, \dots, v_{n-1}]| < |v[v_{n-orden-1}, \dots, v_n]|$ 
40:      $\vec{v}' = \vec{v}[v_{n-4} \dots v_{n-1}]$ 
41:      $f_{2n-4} = \vec{m}_3 \times \vec{v}'$ 
42: else
43:      $\vec{v}' = \vec{v}[v_{n-3} \dots v_n]$ 
44:      $f_{2n-4} = \vec{m}_1 \times \vec{v}'$ 
45: fin si
46: //Predicción del último elemento
47:  $\vec{v}' = \vec{v}[v_{n-orden-1} \dots v_n]$ 
48:  $f_{2n-2} = \vec{m}_3 \times \vec{v}'$ 
49: else
50: //Véase algoritmo ??
51:  $s = selJerarquicaEno(\vec{v}, orden)$ 
52: si  $S \in S_2$  then
53:      $\vec{v}' = \vec{v}[v_1 \dots v_{orden}]$ 
54:      $f_4 = \vec{m}_1 \times \vec{v}'$ 
55: else
56:      $\vec{v}' = \vec{v}[v_2 \dots v_{orden+1}]$ 
57:      $f_4 = \vec{m}_3 \times \vec{v}'$ 
58: fin si
59: for  $i = 6$  hasta  $2 * n - 6$  a incrementos de 2 do
60:     //Véase algoritmo 2
61:      $s = selJerarquicaEno(\vec{v}, orden)$ 

```

Algoritmo 6 ENO Zoom Parte 3/3

```

62:     si  $S \in S_1$  then
63:          $\vec{v}' = \vec{v}[v_{\frac{i}{2}-1} \dots v_{\frac{i}{2}+2}]$ 
64:          $f_j = \vec{m}_1 \times \vec{v}'$ 
65:     else si  $S \in S_2$  then
66:          $\vec{v}' = \vec{v}[v_{\frac{i}{2}-2} \dots v_{\frac{i}{2}+1}]$ 
67:          $f_j = \vec{m}_3 \times \vec{v}'$ 
68:     else
69:
70:          $\vec{v}' = \vec{v}[v_{\frac{i}{2}} \dots v_{\frac{i}{2}+3}]$ 
71:          $f_j = \vec{m}_2 \times \vec{v}'$ 
72:     fin si
73: fin for
74: fin si
75: //Predicción del penúltimo elemento.
76: si then  $|v[v_{n-orden}, \dots, v_{n-1}]| < |v[v_{n-orden-1}, \dots, v_n]|$ 
77:      $\vec{v}' = \vec{v}[v_{n-4} \dots v_{n-1}]$ 
78:      $f_{2n-4} = \vec{m}_3 \times \vec{v}'$ 
79: else
80:      $\vec{v}' = \vec{v}[v_{n-3} \dots v_n]$ 
81:      $f_{2n-4} = \vec{m}_1 \times \vec{v}'$ 
82: fin si
83: //Predicción del último elemento
84:  $\vec{v}' = \vec{v}[v_n \dots v_{n-orden+1}]$ 
85:  $f_{2n-2} = \vec{m}_3 \times \vec{v}'$ 

```

3.3.2. Interpolación ENO *Subcell Resolution*

Esta técnica se presenta como una mejora de la interpolación ENO vista en 3.3.1. Supongamos que f es una función continua con una esquina en $x_d \in I_i = [x_{i-1}, x_i]$, es decir,

$$f(x) = \begin{cases} f_l(x), & x \leq x_d, \\ f_r(x), & x \geq x_d, \end{cases}$$

siendo f_l y f_r lo suficientemente suaves tal que $f_l(x_d) = f_r(x_d)$ y $f'_l(x_d^-) \neq f'_r(x_d^+)$.

Sean q_{i-1} y q_{i+1} los polinomios interpoladores de f obtenidos mediante interpolación ENO en los intervalos I_{i-1} , I_{i+1} respectivamente. Así obtenemos:

$$f(x) = \begin{cases} f(x) = q_{i-1}(x) + O(h^{r+1}), & \forall x \in I_{i-1}, \\ f(x) = q_{i+1}(x) + O(h^{r+1}), & \forall x \in I_{i+1}. \end{cases}$$

Se define la función:

$$G_i(x) = q_{i+1}(x) - q_{i-1}(x). \quad (3.8)$$

Mediante el desarrollo de Taylor alrededor de x_d , obtenemos;

$$\begin{aligned} q_{i+1}(x) &= f(x_d^+) + f'(x_d^+)(x - x_d) + O((x - x_d)^2) + O(h^{r+1}), \\ q_{i-1}(x) &= f(x_d^-) + f'(x_d^-)(x - x_d) + O((x - x_d)^2) + O(h^{r+1}). \end{aligned}$$

y de aquí

$$G_i(x) = [f']_{x_d}(x - x_d) + O((x - x_d)^2) + O(h^{r+1}) \quad (3.9)$$

Si tomamos h lo suficientemente pequeño, se verifica $G_i(x) \supset [f']_{x_d}(x - x_d)$, y la función $G_i(x)$ posee una única raíz en el intervalo $(x_i - 1, x_i)$, ya que si $(x_d - x_{i-1}) = ah$ con $0 < a < 1$, entonces:

$$G_i(x)G_i(x_i - 1) = a(a - 1)[f']_{x_d}^2 h^2 + = (h^2) \approx a(a - 1)[f']_{x_d}^2 h^2 < 0 \quad (3.10)$$

Sea ξ esta raíz, en este caso podemos probar que $|\xi - x_d| =$, para ello definimos $\tilde{G}(x) = f_r(x) - f_l(x)$, así $G_i(x) = \tilde{G}(x) + O(h^r + 1)$, y por lo tanto $0 = G_i(\xi) = \tilde{G}(\xi) + O(h^r + 1)$. Desarrollando por Taylor $\tilde{G}(x)$ alrededor de $G(x)$ alrededor de x_d :

$$0 = \tilde{G}(x_d) + (\xi - x_d)\tilde{G}'(x_d) + O(h^r + 1)$$

donde tomando h lo suficientemente pequeño como para que $\tilde{G}'(x_d) \neq 0$ y teniendo en cuenta que $\tilde{G}'(x_d) = 0$, obtenemos $|\xi - x_d = O(h^r + 1)|$.

Por tanto, una vez hallado ξ , podemos definir la función interpolante de f en el intervalo I_i de modo que a la izquierda de ξ coincida con q_{i-1} y a la derecha de ξ tome el valor de q_{i+1} , esto es:

$$\mathcal{I}^{SR}(x) = \begin{cases} q_l(x), & x \in I_l, l \neq i \\ q_{i-1}(x), & x \in [x_{i-1}, \xi] \\ q_{i+1}(x), & x \in [\xi, x_{i+1}] \end{cases}$$

Así definido, el error de interpolación es $\mathcal{I}(x) = f(x) + O(h^{r+1})$ excepto en una pequeña región alrededor de x_d .

El algoritmo quedara entonces como sigue:

$$\left\{ \begin{array}{l} \text{Hacer } k = 1, \dots, L \\ f_{2j}^k = f_j^{k-1} \\ f_{2j-1}^k = \left\{ \begin{array}{l} ENO \left\{ \begin{array}{ll} 5f_{j-3}^{k-1} + 15f_{j-2}^{k-1} + 5f_{j-1}^{k-1} + f_j^{k-1}, & \text{si } S_j^1 \\ \frac{-f_{j-2}^{k-1} + 9f_{j-1}^{k-1} + 9f_j^{k-1} - f_{j+1}^{k-1}}{16}, & \text{si } S_j^2 \\ 5f_{j-1}^{k-1} + 15f_j^{k-1} + 5f_{j+1}^{k-1} + f_{j+2}^{k-1}, & \text{si } S_j^3 \end{array} \right. \\ \\ ENO - SR \left\{ \begin{array}{ll} \frac{-5f_{j-4}^{k-1} + 21f_{j-3}^{k-1} - 35f_{j-2}^{k-1} + 35f_{j-1}^{k-1}}{16} & \text{si } q_{j-1} \\ \frac{35f_j^{k-1} - 35f_{j+1}^{k-1} + 21f_{j+2}^{k-1} - 5f_{j+3}^{k-1}}{16}, & \text{si } q_{j+1} \end{array} \right. \end{array} \right.$$

Algoritmo

Seguidamente se presenta el pseudocódigo correspondiente a ENO *Subcell Resolution*, estudiado en la sección anterior.

Algoritmo 7 Interpolación ENO-SR

Entrada: *niv*, *orden*, *im*, *met_particion**niv* - Niveles de zoom.*orden* - Orden del polinomio interpolatorio.*im* - Imagen original sobre la cual aplicar el zoom.**Salida:** *b**b* - Imagen interpolada.

```

1:  $[n \ m] = \text{size}(a)$ 
2: //Bucle para los niveles de zoom.
3: for  $k = 1$  hasta  $niv$  do
4:    $n\_filas = 2 * n - 1$ 
5:    $n\_columnas = 2 * m - 1$ 
6:    $b = \text{zeros}(n\_filas, n\_columnas)$ 
7:    $b(1 : 2 : n\_filas, 1 : 2 : n\_columnas) = a(1 \dots n, 1 \dots m)$ 
8:   // Predicción de las columnas.
9:   for  $j = 1$  hasta  $n\_columnas$  a incrementos de 2 do
10:     $b(1 \dots n\_filas, j) = \text{enosr\_zoom}(b(1:2:n\_filas, j), n, orden)'$ 
11:   fin for
12:   // Predicción de las filas.
13:   for  $j = 1$  hasta  $n\_filas$  do
14:     $b(i, 1 \dots n\_columnas) = \text{enosr\_zoom}(b(i, 1:2:n\_filas), n, orden)$ 
15:   fin for
16:   //Actualización de las variables.
17:    $n = n\_filas; m = n\_columnas$ 
18:    $a = b$ 
19: fin for

```

Algoritmo 8 ENOSR Zoom Parte 1/3

Entrada: v , n , $orden$, $met_particion$ \vec{v} - Vector de datos. $orden$ - Orden del polinomio interpolatorio. n - Longitud del vector \vec{v} .**Salida:** f \vec{f} - Vector con los valores interpolados.

```

1:  $\vec{f} = \text{zeros}(1, 2 * n - 1)$ 
2:  $\vec{f}(1 : 2 : 2 * n - 1) = \vec{v}(1 \dots n)$ 
3: //Obtenemos las máscaras correspondientes a cada uno de los stencils posibles
   a elegir.
4:  $m_1 = \text{getMask}(orden/2, orden/2)$ 
5:  $m_2 = \text{getMask}(orden/2 - 1, orden/2 + 1)$ 
6:  $m_3 = \text{getMask}(orden/2 + 1, orden/2 - 1)$ 
7:  $m_4 = \left\{ \frac{-5}{16}, \frac{21}{16}, \frac{-35}{16}, \frac{35}{16} \right\}$ 
8:  $m_5 = \left\{ \frac{35}{16}, \frac{-35}{16}, \frac{21}{16}, \frac{-5}{16} \right\}$ 
9: //Predicción del primer elemento con máscara lineal.
10:  $\vec{v}' = \vec{v}[v_1 \dots v_{orden}]$ 
11:  $f_2 = \vec{m}_2 \times \vec{v}'$ 
12: //Predicción del segundo elemento. Véase algoritmo ??
13:  $s = \text{selJerarquicaEno}(\vec{v}, orden)$ 
14: si  $s \in S_2$  then
15:    $\vec{v}' = \vec{v}[v_1 \dots v_{orden}]$ 
16:    $f_4 = \vec{m}_1 \times \vec{v}'$ 
17: else
18:    $\vec{v}' = \vec{v}[v_2 \dots v_{orden+1}]$ 
19:    $f_4 = \vec{m}_2 \times \vec{v}'$ 
20: fin si
21: //Predicción del tercer elemento. Véase algoritmo ??
22:  $s = \text{selJerarquicaEno}(\vec{v}, orden)$ 
23: si  $S \in S_2$  then
24:    $\vec{v}' = \vec{v}[v_1 \dots v_{orden}]$ 
25:    $f_4 = \vec{m}_1 \times \vec{v}'$ 
26: else
27:    $\vec{v}' = \vec{v}[v_2 \dots v_{orden+1}]$ 
28:    $f_4 = \vec{m}_2 \times \vec{v}'$ 
29: fin si
30: si  $s \in S_1$  then
31:    $\vec{v}' = \vec{v}[v_1 \dots v_{orden}]$ 
32:    $f_6 = \vec{m}_1 \times \vec{v}'$ 

```

Algoritmo 10 ENO-SR Zoom Parte 2/3

```

33: else si  $s \in S_2$  then
34:    $\vec{v}' = \vec{v}[v_3 \dots v_{orden+2}]$ 
35:    $f_6 = \vec{m}_3 \times \vec{v}'$ 
36: else
37:    $\vec{v}' = \vec{v}[v_2 \dots v_{orden+1}]$ 
38:    $f_6 = \vec{m}_2 \times \vec{v}'$ 
39: fin si
40: for  $j = 8$  hasta  $2 * n - 8$  a incrementos de 2 do
41:   //Véase algoritmo ??
42:    $s_i = stencil(\vec{v}[v_{\frac{j}{2}-3} \dots v_{\frac{j}{2}+2}])$ 
43:    $s_d = stencil(\vec{v}[v_{\frac{j}{2}-1} \dots v_{\frac{j}{2}+4}])$ 
44:   // Si no están descentrados.
45:   si  $Ss_i = 'I'$  and  $s_d = 'D'$  then
46:      $q_11 = v_{\frac{j}{2}}$ 
47:      $q_12 = -v_{\frac{j}{2}-3} + 4v_{\frac{j}{2}-2} - 6vv_{\frac{j}{2}-1} + 4v_{\frac{j}{2}}$ 
48:      $q_21 = 4v_{\frac{j}{2}+1} - 6vv_{\frac{j}{2}+2}v_{\frac{j}{2}+3}v_{\frac{j}{2}+4}$ 
49:      $q_22 = v_{\frac{j}{2}+1}$ 
50:      $g_1 = q_21 - q_11; g_2 = q_22 - q_12$ 
51:     si  $g_1 * g_2 < 0$  then
52:       //Existe una discontinuidad en el intervalo.
53:        $\vec{v}_1 = \vec{v}[v_{\frac{j}{2}-3} \dots v_{\frac{j}{2}}]$ 
54:        $p_1 = \vec{v}_1 \times \vec{m}_4$ 
55:        $\vec{v}_2 = \vec{v}[v_{\frac{j}{2}+1} \dots v_{\frac{j}{2}+4}]$ 
56:        $p_2 = \vec{v}_2 \times \vec{m}_5$ 
57:       // Y se evalúa g en el punto medio.
58:        $g = p_2 - p_1;$ 
59:       si  $g_1 * g < 0$  then
60:          $f_j = g_2$ 
61:       else
62:          $f_j = g_1$ 
63:       fin si
64:     fin si
65:      $\vec{v}' = \vec{v}[v_{\frac{j}{2}-1} \dots v_{\frac{j}{2}+2}]$ 
66:      $f_j = \vec{m}_1 \times \vec{v}'$ 
67:     //Si están descentrados, se aplica ENO jerárquico.
68:   else
69:     //Véase algoritmo 2
70:      $s = selJerarquicaEno(\vec{v}, orden)$ 

```

Algoritmo 12 ENO-SR Zoom Parte 2/3

```

71:      si  $s \in S_1$  then
72:           $\vec{v}' = \vec{v}[v_{\frac{i}{2}-1} \dots v_{\frac{i}{2}+2}]$ 
73:           $f_j = \vec{m}_1 \times \vec{v}'$ 
74:      else si  $s \in S_2$  then
75:           $\vec{v}' = \vec{v}[v_{\frac{i}{2}-2} \dots v_{\frac{i}{2}+1}]$ 
76:           $f_j = \vec{m}_3 \times \vec{v}'$ 
77:      else
78:
79:           $\vec{v}' = \vec{v}[v_{\frac{i}{2}} \dots v_{\frac{i}{2}+3}]$ 
80:           $f_j = \vec{m}_2 \times \vec{v}'$ 
81:      fin si
82:  fin si
83: fin for
84: // Predicción del antepenúltimo elemento
85:  $s = selJerarquicaEno(\vec{v}, orden)$ 
86: si  $s \in S_1$  then
87:      $\vec{v}' = \vec{v}[v_{n-4} \dots v_{n-1}]$ 
88:      $f_j = \vec{m}_1 \times \vec{v}'$ 
89: else si  $s \in S_2$  then
90:      $\vec{v}' = \vec{v}[v_{n-5} \dots v_{n-2}]$ 
91:      $f_j = \vec{m}_3 \times \vec{v}'$ 
92: else
93:
94:      $\vec{v}' = \vec{v}[v_{n-3} \dots v_n]$ 
95:      $f_j = \vec{m}_2 \times \vec{v}'$ 
96: fin si
97: // Predicción del penúltimo elemento
98:  $s = selJerarquicaEno(\vec{v}, orden)$ 
99: si  $s \in S_1$  then
100:     $\vec{v}' = \vec{v}[v_{n-3} \dots v_n]$ 
101:     $f_j = \vec{m}_1 \times \vec{v}'$ 
102: else si  $s \in S_2$  then
103:     $\vec{v}' = \vec{v}[v_{n-4} \dots v_{n-1}]$ 
104:     $f_j = \vec{m}_3 \times \vec{v}'$ 
105: else
106:
107:     $\vec{v}' = \vec{v}[v_{n-3} \dots v_n]$ 
108:     $f_j = \vec{m}_2 \times \vec{v}'$ 
109: fin si
110: // Predicción del último elemento
111:  $\vec{v}' = \vec{v}[v_n \dots v_{n-orden+1}]$ 
112:  $f_{2n-2} = \vec{m}_3 \times \vec{v}'$ 

```

Algoritmo 13 Funcion *Stencil*

Entrada: \vec{v} \vec{v} - Vector de seis elementos.**Salida:** s val - Descentramiento del vector ENO Jerárquico.

```

1: si  $|(v_4 - v_3) - (v_3 - v_2)| \leq |(v_5 - v_4) - (v_4 - v_3)|$  then
2:   si  $|((v_5 - v_4) - (v_4 - v_3)) - ((v_4 - v_3) - (v_3 - v_2))| \leq |((v_4 - v_3) - (v_3 -$ 
    $v_2)) - ((v_3 - v_2) - (v_2 - v_1))|$  then
3:      $val = ' C'$ 
4:   else
5:      $val = ' I'$ 
6:   fin si
7: else
8:   si  $|((v_5 - v_4) - (v_4 - v_3)) - ((v_4 - v_3) - (v_3 - v_2))| \leq |((v_5 - v_4) - (v_3 -$ 
    $v_2)) - ((v_5 - v_4) - (v_4 - v_3))|$  then
9:      $val = C$ 
10:  else
11:     $val = ' I'$ 
12:  fin si
13: fin si

```

3.3.3. Interpolación WENO

La interpolación WENO (*Weighed ENO*) es otra técnica de reconstrucción no lineal que se presenta como otra mejora respecto a la interpolación ENO. Como se ha tratado anteriormente, ésta última técnica selecciona el *stencil* más adecuado para intervalo, consiguiendo una aproximación del orden de $r + 1$ en aquellos intervalos que se hallan lo suficientemente aisladas. A pesar de ello, existen aspectos de la técnica que se pueden mejorar:

En primer lugar, el proceso de selección del *stencil* es demasiado sensible a las perturbaciones, y un error de redondeo dado entre dos diferencias divididas muy próximas, incurriría en un cambio de selección *stencil*.

En segundo lugar, otro punto a tener en cuenta es que, en aquellas regiones en las que la función es suave, no es necesario hacer esta selección del *stencil*, ya que la selección utilizada por cualquier método lineal obtendría el mismo conjunto de puntos.

Por último, sería posible aumentar el orden de exactitud de la aproximación, ya que tomando *stencils* de r intervalos, la interpolación ENO se lleva a cabo seleccionando uno de entre r posibles, obteniendo como se ha mencionado, una aproximación del orden de r . Pero dado que existen $2r - 1$ subintervalos contenidos en los r *stencils*, se pierde información proporcionada por $r - 1$ de estos *stencils*. Si la función es lo suficientemente suave en estas regiones, se podría llegar a obtener un orden de aproximación igual a $2r$ como máximo en estas regiones, utilizando la información dada por los $2r - 1$ *stencils*.

Para solucionar los dos primeros problemas, se presentó una estrategia de poda o sesgo, la cual consiste en tomar como base un *stencil* centrado en el intervalo donde se realiza la interpolación, y utilizarlo para modificar el criterio de selección del *stencil* con un parámetro de sesgo.

En contraposición con la técnica ENO, que construye el interpolante seleccionando un *stencil* para cada subintervalo, el método WENO, asigna a cada uno de éstos subintervalos todos los *stencils* posibles, y el polinomio se calcula como una combinación convexa de los polinomios correspondientes a dichos *stencils*. Con este tipo de construcción, se prioriza a aquellos polinomios construidos a partir de *stencils* en donde la función es suave, de forma que aquellos que poseen alguna discontinuidad contribuyen al cálculo de forma prácticamente nula. Por tanto, se conserva el efecto ENO (la interpolación en regiones cercanas a singularidades se obtiene mediante información solo de regiones donde la función es suave), y los errores cometidos por unos se pueden cancelar por otros (por la construcción mediante combinación convexa de polinomios), obteniéndose un orden de aproximación mayor.

De manera formal, sean \mathcal{S}_{i+k} , $k = 0, \dots, r - 1$ los r *stencils* conteniendo al intervalo I_i , y p_{i+k} el polinomio construido a partir de dicho *stencil* \mathcal{S}_{i+k} ; el polinomio

	$k = 0$	$k = 1$	$k = 2$
$r = 2$	1/2	1/2	1/2
$r = 3$	3/16	10/16	3/16

Tabla 3.1: Pesos óptimos

interpolador queda como sigue:

$$\mathcal{S}_{i+k} = \{x_{i+k-r}, \dots, x_{i+k}\}, k = 0, \dots, r-1$$

$$p_i^{WENO}(x) = \sum_{k=0}^{r-1} w_k^i p_{i+k}(x)$$

donde

$$w_k^i \geq 0, k = 0, \dots, r-1, \sum_{k=0}^{r-1} w_k^i = 1 \quad (3.11)$$

Como se ha mencionado, el interpolante WENO toma información de $2r$ nodos con la pretensión de alcanzar una aproximación de este orden en aquellos intervalos donde la función sea suave. Sea $\tilde{\rho}_i^{2r-1}$ la aproximación empleando $2s$ nodos $\{x_{i-r}, \dots, x_{i+r-1}\}$ y sean $\tilde{\rho}_i^r$ las aproximaciones obtenidas con los stencils \mathcal{S}_{i+k} . Dicha aproximación se puede expresar como una combinación lineal de las r aproximaciones de orden $r+1$, esto es:

$$\tilde{\rho}_i^{2r-1} = \sum_{k=0}^{r-1} C_k^r p_{i+k}^r \quad (3.12)$$

con $C_k^r \geq 0, \forall k$ y $\sum k = 0^{r-1} C_k^r = 1$. Estas constantes se denominan pesos óptimos que los podemos encontrar en [5].

Pasamos a definir los pesos de modo que verifiquen, por un lado:

$$w_k^i = C_k^r + O(h^{r-1}), k = 0, \dots, r-1 \quad (3.13)$$

para imponer que la aproximación obtenida sea de orden $2r$; y por otro lado para que la contribución de los polinomios que crucen alguna discontinuidad sea la menor posible. Para poder satisfacer esta última condición, definimos:

$$w_k^i = \frac{\alpha_k^i}{\sum_{s=0}^{r-1} \alpha_s^i}, k = 0, \dots, r-1, \quad (3.14)$$

con

$$\alpha_k^i = \frac{C_k^r}{(\epsilon + \mathcal{IS}_{i+k})^2}. \quad (3.15)$$

Destacar que \mathcal{IS}_{i+k} es un indicador de suavidad de $f(x)$ en el stencil \mathcal{S}_{i+k} , y ϵ una constante positiva introducida para evitar la anulación del denominador.

Llegados a este punto vemos que los pesos verifican la condición [3.11] independientemente del indicador de suavidad utilizado. Para que la contribución de los polinomios que cruzan alguna singularidad sea prácticamente nula, es suficiente con que $\mathcal{IS}_{i+k} = O(1)$ en aquellos stencils que la función presenta alguna discontinuidad.

El algoritmo propuesto queda como sigue:

$$\left\{ \begin{array}{l} \text{Hacer } k = 1 \dots, L \\ f_{2j}^k = f_{j-1}^{k-1} \\ f_{2j-1}^k = w_{j-1}^{k-1} s_{j-1}^{k-1} + w_j^{k-1} s_j^{k-1} + w_{j+1}^{k-1} s_{j+1}^{k-1} \end{array} \right.$$

donde

$$\left\{ \begin{array}{l} s_{j-1}^{k-1} = 5f_{j-3}^{k-1} + 15f_{j-2}^{k-1} + 5f_{j-1}^{k-1} + f_j^{k-1} \\ s_j^{k-1} = \frac{-f_{j-2}^{k-1} + 9f_{j-1}^{k-1} + 9f_j^{k-1} - f_{j+1}^{k-1}}{16} \\ s_{j+1}^{k-1} = 5f_{j-3}^{k-1} + 15f_{j-2}^{k-1} + 5f_{j-1}^{k-1} + f_j^{k-1} \end{array} \right.$$

Y los pesos w_{j-1}^{k-1} , w_j^{k-1} y w_{j+1}^{k-1} se calculan mediante las expresiones

$$\begin{aligned} w_{j-1}^{k-1} &= \frac{\alpha_{j-1}^{k-1}}{\alpha_{j-1}^{k-1} + \alpha_j^{k-1} + \alpha_{j+1}^{k-1}} \\ w_j^{k-1} &= \frac{\alpha_j^{k-1}}{\alpha_{j-1}^{k-1} + \alpha_j^{k-1} + \alpha_{j+1}^{k-1}} \\ w_{j+1}^{k-1} &= \frac{\alpha_{j+1}^{k-1}}{\alpha_{j-1}^{k-1} + \alpha_j^{k-1} + \alpha_{j+1}^{k-1}} \end{aligned}$$

Los valores de α_{j-1}^{k-1} , α_j^{k-1} , α_{j+1}^{k-1} son, respectivamente

$$\begin{aligned} \alpha_{j-1}^{k-1} &= \frac{\frac{3}{16}}{\epsilon + \mathcal{IS}_{j-1}^{k-1}} \\ \alpha_j^{k-1} &= \frac{\frac{10}{16}}{\epsilon + \mathcal{IS}_j^{k-1}} \\ \alpha_{j+1}^{k-1} &= \frac{\frac{3}{16}}{\epsilon + \mathcal{IS}_{j+1}^{k-1}} \end{aligned}$$

siendo ϵ una constante positiva introducida para evitar la anulación del denominador, y que típicamente adquiere valores tales como $\epsilon = 10^{-5}$ o $\epsilon = 10^{-6}$.

Finalmente solo queda definir cómo son calculados los indicadores de suavidad. En [9] se propone calcularlos como sigue:

$$\mathcal{IS}_{j-1}^{k-1} = \frac{1}{2} \left[\left(f [x_{j-2}^{k-1}, x_{j-1}^{k-1}] - f [x_{j-3}^{k-1}, x_{j-2}^{k-1}] \right)^2 + \left(f [x_{j-1}^{k-1}, x_j^{k-1}] - f [x_{j-2}^{k-1}, x_{j-1}^{k-1}] \right)^2 \right] \\ + \left(f [x_{j-1}^{k-1}, x_j^{k-1}] - 2f [x_{j-2}^{k-1}, x_{j-1}^{k-1}] + f [x_{j-3}^{k-1}, x_{j-2}^{k-1}] \right)^2$$

$$\mathcal{IS}_j^{k-1} = \frac{1}{2} \left[\left(f [x_{j-1}^{k-1}, x_j^{k-1}] - f [x_{j-2}^{k-1}, x_{j-1}^{k-1}] \right)^2 + \left(f [x_j^{k-1}, x_{j+1}^{k-1}] - f [x_{j-1}^{k-1}, x_j^{k-1}] \right)^2 \right] \\ + \left(f [x_j^{k-1}, x_{j+1}^{k-1}] - 2f [x_{j-1}^{k-1}, x_j^{k-1}] + f [x_{j-2}^{k-1}, x_{j-1}^{k-1}] \right)^2$$

$$\mathcal{IS}_{j+1}^{k-1} = \frac{1}{2} \left[\left(f [x_j^{k-1}, x_{j+1}^{k-1}] - f [x_{j-1}^{k-1}, x_j^{k-1}] \right)^2 + \left(f [x_{j+1}^{k-1}, x_{j+2}^{k-1}] - f [x_j^{k-1}, x_{j+1}^{k-1}] \right)^2 \right] \\ + \left(f [x_{j+1}^{k-1}, x_{j+2}^{k-1}] - 2f [x_j^{k-1}, x_{j+1}^{k-1}] + f [x_{j-1}^{k-1}, x_j^{k-1}] \right)^2$$

Algoritmo

A continuación se propone el algoritmo de reconstrucción siguiendo la técnica WENO.

Algoritmo 14 Interpolación WENO

Entrada: *niv*, *orden*, *im*, *met_particion**niv* - Niveles de zoom.*orden* - Orden del polinomio interpolatorio.*im* - Imagen original sobre la cual aplicar el zoom.**Salida:** *b**b* - Imagen interpolada.

```

1:  $[n \ m] = \text{size}(a)$ 
2: //Bucle para los niveles de zoom.
3: for  $k = 1$  hasta  $niv$  do
4:    $n\_filas = 2 * n - 1$ 
5:    $n\_columnas = 2 * m - 1$ 
6:    $b = \text{zeros}(n\_filas, n\_columnas)$ 
7:    $b(1 : 2 : n\_filas, 1 : 2 : n\_columnas) = a(1 : \dots n, 1 \dots m);$ 
8:   // Predicción de las columnas.
9:   for  $j = 1$  hasta  $n\_columnas$  a incrementos de 2 do
10:     $b(1 \dots n\_filas, j) = \text{weno\_zoom}(b(1:2:n\_filas, j), n, orden)'$ 
11:   fin for
12:   // Predicción de las filas.
13:   for  $j = 1$  hasta  $n\_filas$  do
14:     $b(i, 1 \dots n\_columnas) = \text{weno\_zoom}(b(i, 1:2:n\_filas), n, orden)$ 
15:   fin for
16:   //Actualización de las variables.
17:    $n = n\_filas; m = n\_columnas$ 
18:    $a = b$ 
19: fin for

```

Algoritmo 15 WENO Zoom Parte 1/2**Entrada:** v, n, orden \vec{v} - Vector de datos. n - Longitud del vector \vec{v} . orden - Orden del polinomio interpolatorio.**Salida:** f \vec{f} - Vector con los valores interpolados.

```

1:  $\vec{f} = \text{zeros}(1, 2 * n - 1)$ 
2:  $\vec{f}(1 : 2 : 2 * n - 1) = \vec{v}(1 \dots n)$ 
3: //Obtenemos las máscaras correspondientes a cada uno de los stencils posibles
   a elegir.
4:  $m_1 = \text{getMask}(\text{orden}/2, \text{orden}/2)$ 
5:  $m_2 = \text{getMask}(\text{orden}/2 - 1, \text{orden}/2 + 1)$ 
6:  $m_3 = \text{getMask}(\text{orden}/2 + 1, \text{orden}/2 - 1)$ 
7: //Predicción del primer elemento con máscara lineal.
8:  $\vec{v}' = \vec{v}[v_1 \dots v_{\text{orden}}]$ 
9:  $f_2 = \vec{m}_2 \times \vec{v}'$ 
10: si  $|f[v_1 \dots v_4]| \leq |f[v_2 \dots v_5]|$  then
11:    $\vec{v}' = \vec{v}[v_1 \dots v_4]$ 
12:    $f_2 = \vec{m}_1 \times \vec{v}'$ 
13: else
14:    $\vec{v}' = \vec{v}[v_2 \dots v_5]$ 
15:    $f_2 = \vec{m}_2 \times \vec{v}'$ 
16: fin si
17: for  $i = 6$  hasta  $2 * n - 6$  a incrementos de 2 do
18:   //Predicción del primer elemento con máscara lineal.
19:    $\epsilon = 10^{-6}$ 
20:    $e = v_{\frac{i}{2}-1} - v_{\frac{i}{2}-2}$ 
21:    $e_2 = v_{\frac{i}{2}} - v_{\frac{i}{2}-1}$ 
22:    $e_3 = v_{\frac{i}{2}+1} - v_{\frac{i}{2}}$ 
23:    $e_4 = v_{\frac{i}{2}+2} - v_{\frac{i}{2}+1}$ 
24:    $e_5 = v_{\frac{i}{2}+3} - v_{\frac{i}{2}+2}$ 
25:    $p_{\text{izquierdo}} = \frac{1}{2}((e_2 - e_1)^2 + (e_3 - e_2)^2 + (e_3 - 2e_2 + e_1)^2)$ 
26:    $p_{\text{derecho}} = \frac{1}{2}((e_3 - e_2)^2 + (e_4 - e_3)^2 + (e_4 - 2e_3 + e_2)^2)$ 
27:    $p_{\text{centro}} = \frac{1}{2}((e_4 - e_3)^2 + (e_5 - e_4)^2 + (e_5 - 2e_4 + e_3)^2)$ 
28:   //Numeradores de los pesos
29:    $nw_{izq} =$ 

```

Algoritmo 16 WENO Zoom Parte 2/2

```

30: //Pesos
31: sum = nwizq + nwcentro + nwder
32: wl =  $\frac{nw_{izq}}{sum}$ 
33: wr =  $\frac{nw_{der}}{sum}$ 
34: wc =  $\frac{nw_{centro}}{sum}$ 
35: fin for
36: // Suma de los polinomios multiplicados por sus pesos.
37:  $\vec{v}' = [v_{\frac{j}{2}-1}, \dots, v_{\frac{j}{2}+2}]$ 
38:  $\vec{q}_c = \vec{m}_1 \times \vec{v}'$ 
39:  $\vec{v}' = [v_{\frac{j}{2}-2}, \dots, v_{\frac{j}{2}+1}]$ 
40:  $\vec{q}_l = \vec{m}_2 \times \vec{v}'$ 
41:  $\vec{v}' = [v_{\frac{j}{2}}, \dots, v_{\frac{j}{2}+3}]$ 
42:  $\vec{q}_d = \vec{m}_3 \times \vec{v}'$ 
43:  $f_j = w_l q_l + w_c q_c + w_r q_d$ 
44: //Predicción del penúltimo elemento.
45: si  $|f[v_{n-3}, \dots, v_n]| \leq |f[v_{n-4}, \dots, v_{n-1}]|$  then
46:    $\vec{v}' = [v_{n-3}, \dots, v_n]$ 
47:    $f_{2n-4} = \vec{m}_1 \times \vec{v}'$ 
48: else
49:    $\vec{v}' = [v_{n-4}, \dots, v_{n-1}]$ 
50:    $f_{2n-4} = \vec{m}_3 \times \vec{v}'$ 
51: fin si
52: //Predicción del penúltimo elemento.
53:  $\vec{v}' = [v_{n-3}, \dots, v_n]$ 
54:  $f_{2n-4} = \vec{m}_3 \times \vec{v}'$ 

```

3.3.4. Interpolación Racional

Como se verá en esta sección, la interpolación racional posee multitud de semejanzas con la interpolación WENO, vista en secciones anteriores.

De manera formal y de forma análoga a cómo se ha explicado en los casos anteriores, consideraremos $X = \{x_i\}_{i=0}^N$, $x_i = ih$, $Nh = 1$, una partición de nodos igualmente espaciados en el intervalo $[0, 1]$, y $f_i = f(x_i)$ los valores que toma una función dada $f(x)$ en los nodos de dicha partición. En [17] se propone emplear como aproximación del valor de la función en $f(x_{i-1} + \frac{h}{2}) = f_{i-\frac{1}{2}}$ una media ponderada entre f_{i-1} y f_i .

$$\hat{f}_{i-\frac{1}{2}} = w_{i-1}f_{i-1} + w_i f_i, \text{ con } w_{i-1} + w_i = 1, \quad (3.16)$$

siendo $\hat{f}_{i-\frac{1}{2}}$ la aproximación obtenida.

Los valores para los pesos pueden ser definidos de la siguiente manera:

$$w_{i-1}^1 = \frac{1 + \alpha(f_i - f_{i+1})^2}{2 + \alpha((f_{i-2} - f_{i-1})^2 + (f_i - f_{i+1})^2)}, \quad (3.17)$$

$$w_i^1 = \frac{1 + \alpha(f_{i-2} - f_{i-1})^2}{2 + \alpha((f_{i-2} - f_{i-1})^2 + (f_i - f_{i+1})^2)}, \quad (3.18)$$

$$(3.19)$$

$$w_{i-1}^2 = \frac{1 + \alpha(f_{i-1} - f_{i+1})^2}{2 + \alpha((f_{i-2} - f_i)^2 + (f_{i-1} - f_{i+1})^2)}, \quad (3.20)$$

$$w_i^2 = \frac{1 + \alpha(f_{i-2} - f_i)^2}{2 + \alpha((f_{i-2} - f_i)^2 + (f_{i-1} - f_{i+1})^2)}, \quad (3.21)$$

$$(3.22)$$

$$w_i^3 = \frac{1 + \alpha(f_{i-1} - f_{i+1})^2 + (f_i - f_{i+1})^2}{2 + \alpha(\sum_{k=0}^1 ((f_{i-2} - f_{i-k})^2 + (f_{i-k} - f_{i+1})^2))}, \quad (3.23)$$

$$w_{i-1}^3 = \frac{1 + \alpha(f_{i-2} - f_i)^2 + (f_{i-2} - f_{i-1})^2}{2 + \alpha(\sum_{k=0}^1 ((f_{i-2} - f_{i-k})^2 + (f_{i-k} - f_{i+1})^2))}, \quad (3.24)$$

$$(3.25)$$

donde α es un parámetro variable. Si tomamos $\alpha = 0$ lo que tenemos es la media ponderada entre f_{i-1} y f_i , esto es:

$$\hat{f}_{i-\frac{1}{2}} = \omega_{i-1}f_{i-1} + \omega_i f_i = \frac{1}{2}f_{i-1} + \frac{1}{2}f_i, k = 1, 2, 3. \quad (3.26)$$

Cabe destacar que si f es simétrica respecto de $x_{i-\frac{1}{2}}$ la predicción obtenida coincidirá con el valor medio, es decir con la expresión (3.26), ya que se verificarán las siguientes relaciones:

$$\begin{aligned} f(x_{i-\frac{1}{2}} + h) &= f(x_{i-\frac{1}{2}} - h), \\ f(x_{i-\frac{1}{2}} + h) - f(x_{i-\frac{1}{2}}) &= f(x_{i-\frac{1}{2}}) - f(x_{i-\frac{1}{2}} - h). \end{aligned}$$

y, por tanto,

$$\begin{aligned} |f_{i-2} - f_{i-1}| &= |f_i - f_{i+1}|, \\ |f_{i-1} - f_{i+1}| &= |f_{i-2} - f_i|. \end{aligned} \tag{3.27}$$

Por todo lo expuesto, para cualquier definición de los pesos de la forma definida en (3.17), (3.20) y en (3.23), podemos considerar la expresión (3.16) como una interpolación WENO de orden 2, ya que empleamos dos nodos para alcanzar el orden óptimo. Además los pesos empleados aproximan a los pesos óptimos de la misma manera que aquellos utilizados en la interpolación WENO. Destacar que en caso de que exista alguna discontinuidad en el intervalo $[x_{i-1}, x_i]$, la aproximación dada por \hat{p}_{i-1}^1 es del orden del salto, esto es, $\hat{p}_{i-1}^1 - f_{i-\frac{1}{2}} = O(|f|)$, y el error cometido en dicha continuidad es $w_{i-1}^k f_{i-1} - f_{i-\frac{1}{2}} = O(|f|)$.

Algoritmo

Propuesta de algoritmo siguiendo el método de reconstrucción Racional.

Algoritmo 17 Interpolación Racional

Entrada: niv , $orden$, im niv - Niveles de zoom. $orden$ - Orden del polinomio interpolatorio. im - Imagen original sobre la cual aplicar el zoom.**Salida:** b b - Imagen interpolada.

```

1:  $[n\ m] = size(a)$ 
2: //Bucle para los niveles de zoom.
3: for  $k = 1$  hasta  $niv$  do
4:    $n\_filas = 2 * n - 1$ 
5:    $n\_columnas = 2 * m - 1$ 
6:    $b = zeros(n\_filas, n\_columnas)$ 
7:    $b(1 : 2 : n\_filas, 1 : 2 : n\_columnas) = a(1 \dots n, 1 \dots m)$ 
8:   // Predicción de las columnas.
9:   for  $j = 1$  hasta  $n\_columnas$  a incrementos de 2 do
10:     $b(1 \dots n\_filas, j) = pph\_racional(b(1:2:n\_filas, j), n, orden)'$ 
11:   fin for
12:   // Predicción de las filas.
13:   for  $j = 1$  hasta  $n\_filas$  do
14:     $b(i, 1 \dots n\_columnas) = pph\_racional(b(i, 1:2:n\_filas), n, orden)$ 
15:   fin for
16:   //Actualización de las variables.
17:    $n = n\_filas; m = n\_columnas$ 
18:    $a = b$ 
19: fin for

```

Algoritmo 18 Racional Zoom

Entrada: $v, n, orden$ \vec{v} - Vector de datos. n - Longitud del vector \vec{v} . $orden$ - Orden del polinomio interpolatorio.**Salida:** f \vec{f} - Vector con los valores interpolados.

```

1:  $\vec{f} = \text{zeros}(1, 2 * n - 1)$ 
2:  $\vec{f}(1 : 2 : 2 * n - 1) = \vec{v}(1 \dots n)$ 
3: //Obtenemos las máscaras correspondientes a cada uno de los stencils posibles
   a elegir.
4:  $m_1 = \text{getMask}(orden/2, orden/2)$ 
5:  $m_2 = \text{getMask}(orden/2 - 1, orden/2 + 1)$ 
6:  $m_3 = \text{getMask}(orden/2 + 1, orden/2 - 1)$ 
7: //Predicción del primer elemento con máscara lineal.
8:  $\vec{v}' = \vec{v}[v_1 \dots v_{orden}]$ 
9:  $f_2 = \vec{m}_2 \times \vec{v}'$ 
10: si  $|f[v_1 \dots v_4]| \leq |f[v_2 \dots v_5]|$  then
11:    $\vec{v}' = \vec{v}[v_1 \dots v_4]$ 
12:    $f_2 = \vec{m}_1 \times \vec{v}'$ 
13: else
14:    $\vec{v}' = \vec{v}[v_2 \dots v_5]$ 
15:    $f_2 = \vec{m}_2 \times \vec{v}'$ 
16: fin si
17: for  $i = 6$  hasta  $2 * n - 6$  a incrementos de 2 do
18:   //Predicción del primer elemento con máscara lineal.
19:    $e = v_{\frac{i}{2}-1} - v_{\frac{i}{2}-2}$ 
20:    $e_2 = v_{\frac{i}{2}} - v_{\frac{i}{2}-1}$ 
21:    $\vec{v}' = \vec{v}[v_n, v_{n-1}, v_{n-2}, v_{n-3}]$ 
22:    $f_2 = \vec{m}_2 \times \vec{v}'$ 
23:    $w_0 = \frac{1 + \alpha e_1^2}{2 + \alpha e_2^2 + e_1^2}$ 
24:    $w_1 = \frac{1 + \alpha e_2^2}{2 + \alpha e_2^2 + e_1^2}$ 
25:    $f_j = w_0 v_{\frac{i}{2}} + w_1 v_{\frac{i}{2}+1}$ 
26: fin for
27: //Predicción del último elemento.

```

3.3.5. Interpolación PPH

En esta sección se describe un esquema de interpolación a trozos polinomial denominada PPH (*Piecewise Polynomial Harmonic*). Esta técnica de reconstrucción posee varias características deseables como son:

Por un lado, cada polinomio esta constituido por un *stencil* centrado de cuatro puntos. En segundo lugar, en las regiones suaves el la técnica produce el mismo resultado que en al utilizar algún método lineal, como Lagrange. Por último, la exactitud se ve reducida en regiones cerca de las singularidades pero sigue siendo mejor que en el caso lineal.

A continuación se describe el operador de reconstrucción PPH de modo análogo al explicado para otras técnicas detalladas anteriormente.

Sea $I_k^P(x, f^k)$ el operador de reconstrucción PPH y $x \in \mathbb{R}$, tomemos j tal que $x \in [x_{j-1}^k, x_j^k]$. Entonces $I_k^P(x, f^k) = \tilde{P}_j(x, f^k) = f_j^k$, donde $\tilde{P}_j(x, f^k)$ es un polinomio formado a partir de los datos centrados, $\{f_{j-2}^k, f_{j-1}^k, f_j^k, f_{j+1}^k\}$, y tal que $\tilde{P}_j(x_{j-1}^k, f^k) = f_{j-1}^k$ y $\tilde{P}_j(x_j^k, f^k) = f_j^k$.

Se dispone del conjunto de puntos $\{f_{j-2}^k, f_{j-1}^k, f_j^k, f_{j+1}^k\}$ y se quiere realizar la predicción del punto medio, $f_{j-1/2}^k$. Como deducimos de lo anteriormente comentado, si la función no contiene singularidades en el intervalo $[x_{j-2}^k, x_{j+1}^k]$, bastaría con una interpolación centrada para proporcionar una buena aproximación. Pero como hemos visto, cuando la señal muestra singularidades, dicha aproximación pierde exactitud. A continuación se discutirá la modificación propuesta cuando se detecta una singularidad en $[x_j^k, x_{j+1}^k]$.

Supongamos que la diferencia dividida $f[x_{j-1}, x_j, x_{j+1}]$ es mayor o igual que $f[x_{j-2}, x_{j-1}, x_j^k]$ en valor absoluto. Esto indica la posible presencia de una singularidad en un punto $x_d \in [x_j^k, x_{j+1}^k]$. Se considera el trozo polinomial para $[x_{j-1}^k, x_j^k]$ escrito como,

$$P_j(x) = a_0 + a_1(x - x_{j-\frac{1}{2}}) + a_2(x - x_{j-\frac{1}{2}})^2 + a_3(x - x_{j-\frac{1}{2}})^3. \quad (3.28)$$

Para un esquema lineal centrado las cuatro condiciones de interpolación en los puntos x_{j-2}, x_{j-1}, x_j y x_{j+1} son

$$\begin{cases} a_0 - a_1\frac{3}{2}h + a_2(\frac{3}{2}h)^2 - a_3(\frac{3}{2}h)^3 = f_{j-2}, \\ a_0 - a_1\frac{1}{2}h + a_2(\frac{1}{2}h)^2 - a_3(\frac{1}{2}h)^3 = f_{j-1}, \\ a_0 - a_1\frac{1}{2}h + a_2(\frac{1}{2}h)^2 - a_3(\frac{1}{2}h)^3 = f_j, \\ a_0 - a_1\frac{3}{2}h + a_2(\frac{3}{2}h)^2 - a_3(\frac{3}{2}h)^3 = f_{j+1}. \end{cases}$$

Despejando, obtenemos que $a_1 = \frac{f_{j-2} - 27f_{j-1} + 27f_j - f_{j+1}}{24h}$. De este modo, el sistema

anterior es equivalente a

$$\begin{cases} a_0 - a_1 \frac{3}{2}h + a_2 (\frac{3}{2}h)^2 - a_3 (\frac{3}{2}h)^3 = f_{j-2}, \\ a_0 - a_1 \frac{1}{2}h + a_2 (\frac{1}{2}h)^2 - a_3 (\frac{1}{2}h)^3 = f_{j-1}, \\ a_0 - a_1 \frac{1}{2}h + a_2 (\frac{1}{2}h)^2 - a_3 (\frac{1}{2}h)^3 = f_j, \\ a_1 = \frac{f_{j-2} - 27f_{j-1} + 27f_j - f_{j-1}}{24h}. \end{cases}$$

Si introducimos los siguientes cambios de variable $e_{j-\frac{3}{2}} = f[x_{j-2}, x_{j-1}]$, $e_{j-\frac{1}{2}} = f[x_{j-1}, x_j]$, $e_{j+\frac{1}{2}} = f[x_j, x_{j+1}]$, $D_{j-1} = f[x_{j-2}, x_{j-1}, x_j]$ y $D_j = f[x_{j-1}, x_j, x_{j+1}]$; después de una serie de manipulaciones llegamos a la expresión:

$$a_1 = \frac{e_{j-\frac{1}{2}} + 13e_{j-\frac{3}{2}}}{12} - \frac{1}{12} \frac{D_{j-1} + D_j}{2} h.$$

Se puede observar, que en presencia de una discontinuidad de salto en $[x_j, x_{j-1}]$, $a_1 = O(\frac{1}{h})$, ya que $D_j = O(\frac{1}{h^2})$. Este comportamiento es debido a la mala aproximación de la reconstrucción en presencia de discontinuidades. Por ello, se sustituye la media aritmética por la armónica, y se obtiene la nueva expresión para a_1 modificada:

$$\tilde{a}_1 = \frac{e_{j-\frac{1}{2}} + 13e_{j-\frac{3}{2}}}{12} - \frac{1}{12} \frac{2D_{j-1} + D_j}{2D_{j-1} + D_j} h. \quad (3.29)$$

La media armónica consigue adaptarse mejor a la presencia de singularidades porque cuando $|D_{j-1}|$ es $O(1)$ $|D_j|$ es $O(\frac{1}{h^2})$, la media armónica permanece siendo $O(1)$ y, en consecuencia, $\tilde{a}_1 = O(1)$. Cabe destacar también que en las regiones suaves $a_1 - \tilde{a}_1 = O(h^3)$, ya que la diferencia entre la media armónica y la aritmética original es $O(h^2)$. Como resultado la interpolación es de cuarto orden, y $f_{j-\frac{1}{2}} - \hat{P}_j(x_j - \frac{1}{2}) = O(h^4)$ La reconstrucción empleada entonces es de cuarto orden.

$$a_1 = \frac{e_{j-\frac{1}{2}} + 13e_{j-\frac{3}{2}}}{12}.$$

Se tiene entonces $a_1 - \tilde{a}_1 = O(h)$ Aunque adaptada a las singularidades, el grado de exactitud se ha reducido a dos.

De esta manera, el operador reconstrucción PPH estará constituido por la ecuación (3.29) así como por los nuevos operadores \tilde{a}_0 , \tilde{a}_1 , \tilde{a}_2 \tilde{a}_3 si El algoritmo queda como sigue:

$$\begin{cases} \text{Hacer } k = 1 \dots, L \\ f_{2j}^k = f_j^{k-1} \\ f_{2j-1}^k = \begin{cases} \frac{f_{j-1}^k + f_j^k}{2} - \frac{1}{4} \frac{Df_{j-1}^k + Df_j^k}{Df_{j-1}^k + Df_j^k}, & \text{si } Df_{j-1}^k Df_j^k > 0 \\ \frac{f_{j-1}^k + f_j^k}{2}, & \text{en otro caso.} \end{cases} \end{cases}$$

Algoritmo

Algoritmo propuesto para la reconstrucción mediante la técnica PPH descrita en la sección anterior.

Algoritmo 19 Interpolación PPH

Entrada: *niv*, *orden*, *im*, *met_particion**niv* - Niveles de zoom.*orden* - Orden del polinomio interpolatorio.*im* - Imagen original sobre la cual aplicar el zoom.**Salida:** *b**b* - Imagen interpolada.

```

1:  $[n \ m] = \text{size}(a)$ 
2: //Bucle para los niveles de zoom.
3: for  $k = 1$  hasta  $niv$  do
4:    $n\_filas = 2 * n - 1$ 
5:    $n\_columnas = 2 * m - 1$ 
6:    $b = \text{zeros}(n\_filas, n\_columnas)$ 
7:    $b(1 : 2 : n\_filas, 1 : 2 : n\_columnas) = a(1 \dots n, 1 \dots m)$ 
8:   // Predicción de las columnas.
9:   for  $j = 1$  hasta  $n\_columnas$  a incrementos de 2 do
10:     $b(1 \dots n\_filas, j) = \text{pph\_zoom}(b(1:2:n\_filas, j), n, orden, met\_particion)'$ 
11:   fin for
12:   // Predicción de las filas.
13:   for  $j = 1$  hasta  $n\_filas$  do
14:     $b(i, 1 \dots n\_columnas) = \text{pph\_zoom}(b(i, 1:2:n\_filas), n, orden, met\_particion)$ 
15:   fin for
16:   //Actualización de las variables.
17:    $n = n\_filas; m = n\_columnas$ 
18:    $a = b$ 
19: fin for

```

Algoritmo 20 PPH Zoom

Entrada: v , n , $orden$ \vec{v} - Vector de datos. n - Longitud del vector \vec{v} . $orden$ - Orden del polinomio interpolatorio.**Salida:** f \vec{f} - Vector con los valores interpolados.

```

1:  $\vec{f} = \text{zeros}(1, 2 * n - 1)$ 
2:  $\vec{f}(1 : 2 : 2 * n - 1) = \vec{v}(1 \dots n)$ 
3: //Obtenemos la máscara correspondientes a cada uno de los stencils posibles
   a elegir.
4:  $m_2 = \text{getMask}(orden/2 - 1, orden/2 + 1)$ 
5: //Predicción del primer elemento con máscara lineal.
6:  $\vec{v}' = \vec{v}[v_1 \dots v_{orden}]$ 
7:  $f_2 = \vec{m}_2 \times \vec{v}'$ 
8: for  $i = 4$  hasta  $2 * n - 4$  a incrementos de 2 do
9:    $p_1 = v_{\frac{i}{2}-1} - 2v_{\frac{i}{2}} + v_{\frac{i}{2}}^i + 1$ 
10:   $p_2 = v_{\frac{i}{2}} - 2v_{\frac{i}{2}+1} + v_{\frac{i}{2}}^i + 2$ 
11:   $aux = \frac{v_{\frac{i}{2}} + v_{\frac{i}{2}+1}}{2}$ 
12:  si  $p_1 * p_2 > 0$  then
13:     $f_j = aux - \frac{(p_1 * p_2)}{4(p_1 + p_2)}$ 
14:  else
15:     $f_j = aux$ 
16:  fin si
17: fin for
18: //Predicción del último elemento.
19:  $\vec{v}' = \vec{v}[v_n, v_{n-1}, v_{n-2}, v_{n-3}]$ 
20:  $f_2 = \vec{m}_2 \times \vec{v}'$ 

```

Capítulo 4

Experimentos y resultados

En este capítulo se pretende evaluar el grado de calidad que es posible obtener mediante la aplicación de los algoritmos anteriormente vistos. Para ello se comenzará con una breve descripción de aquellos indicadores útiles y medibles que son frecuentemente utilizados en el estudio de imágenes digitales. Acto seguido se detallará el procedimiento por el cual éstos serán medidos y evaluados.

Por último se presentará las imágenes correspondientes a dos niveles de resolución con el fin de poder apreciar visualmente la calidad obtenida, y las tablas con los resultados del experimento para los restantes niveles. Para finalizar, se muestran una serie de gráficas con las que se pretende obtener conclusiones acerca del experimento.

4.1. Indicadores medibles para la evaluación de la calidad

4.1.1. Introducción

Los siguientes indicadores de calidad en imágenes digitales que a continuación se presentan, son frecuentes en el ámbito de la manipulación y el procesamiento de imágenes. A continuación se detalla, para cada uno de ellos, una breve descripción y la motivación de su uso para alguno de ellos. Para facilitar la descripción formal se ha considerado que se tienen dos imágenes \mathcal{A} y \mathcal{A}' definidas de la siguiente manera: $\mathcal{A} := (a_{i,j})_{M \times N}$ y $\mathcal{A}' := (a'_{i,j})_{M \times N}$, siendo \mathcal{A}' una aproximación a \mathcal{A} .

4.1.2. Error Cuadrático Medio

El Error Cuadrático Medio (ECM) o *Mean Squared Error*, (*MSE*), es una medida del cuadrado del error entre dos imágenes; en el caso que nos ocupa entre

las imágenes \mathcal{A} y \mathcal{A}' . Este error mide el grado en que una imagen difiere con respecto a otra, y viene dado por la expresión siguiente:

$$ECM = \frac{1}{MN} \sum_{i=1}^{M-1} \sum_{j=1}^{N-1} |(a_{i,j}) - (a'_{i,j})|^2 \quad (4.1)$$

En el registro de imágenes, es habitual hacer uso de esta medida para cuantificar el error que se produce entre el vector de movimiento real de la imagen objetivo y de la de referencia, y el vector de movimiento estimado en el procedimiento de registro. La presencia de este error se debe a que a veces la estimación no se calcula con una precisión suficiente. Otra medida directamente derivada del ECM es la raíz cuadrática media, o *Root Mean Square*, (*RMS*), calculada como la raíz cuadrada del ECM, de la siguiente manera:

$$RMS = \sqrt{ECM} = \sqrt{\sum_{i=1}^{M-1} \sum_{j=1}^{N-1} |(a_{i,j}) - (a'_{i,j})|^2} \quad (4.2)$$

4.1.3. Relación señal a ruido de pico (PSNR)

La relación PSNR (*Peak to Signal Noise Ratio*) es utilizada para definir la relación entre la máxima energía posible de una señal y el ruido que afecta a su representación fidedigna. Debido a que muchas señales tienen un gran rango dinámico, el PSNR se expresa generalmente en escala logarítmica, utilizando como unidad el decibelio. Cabe recordar que un aumento de 20 dB corresponde a un decrecimiento de una décima parte en la diferencia *RMS* entre las dos imágenes. El uso más habitual del PSNR es como medida cuantitativa de la calidad de la reconstrucción en el ámbito de la compresión de imágenes.

Esta medida se define como:

$$PSNR = 20 \log \frac{b}{\sqrt{ECM}}. \quad (4.3)$$

donde b es el mayor valor posible de la señal, y *RMS* es la raíz cuadrática media.

Para una imagen en formato RGB, la definición del PSNR es la misma, pero el ECM se calcula como la media aritmética de los ECM de los tres colores (R, G y B). Los valores típicos que adopta este parámetro están entre 30 y 50 dB , siendo mayor cuanto mejor es la codificación. El comité MPEG emplea un valor umbral informal de $0,5 \text{ dB}$ en el incremento del PSNR para decidir si se incluye una determinada mejora en un algoritmo de codificación, ya que se considera que este aumento del PSNR es apreciable visualmente.

4.1.4. Correlación cruzada normalizada

El Coeficiente de Correlación Cruzada o *Normalized Cross Correlation*, (*NK*), es una de las medidas de similitud más frecuentemente utilizada. Ésta se calcula entre parejas de bloques pertenecientes a la imagen de referencia y a la imagen objetivo, con el propósito de encontrar el máximo entre dicha medida. Aquél bloque con el que se consigue el máximo es el que determina la correspondencia finalmente establecida.

Este coeficiente permite el alineamiento con precisión de imágenes que han sido trasladadas entre sí, aunque también es posible su aplicación entre imágenes que han sufrido rotaciones leves o escalados.

Como desventajas, se suelen citar entre otras, el elevado coste computacional que requiere, aunque es inferior al de otras medidas frecuentes; y la excesiva planicidad de los máximos de similitud detectados, debido a la autosimilitud de las imágenes.

$$NK = \frac{\sum_{i=1}^M \sum_{j=1}^N (a_{i,j}) \cdot (a'_{i,j})}{\sum_{i=1}^M \sum_{j=1}^N (a_{i,j})^2}.$$

4.1.5. Diferencia media

La Diferencia Media o *Average Difference*, (*AD*) no es más que la media de las diferencias obtenidas entre píxeles de ambas imágenes.

Formalmente

$$AD = \frac{\sum_{i=1}^M \sum_{j=1}^N (a_{i,j}) - (a'_{i,j})}{MN}.$$

4.1.6. Diferencia Máxima

La Diferencia Máxima o *Maximum Difference*, (*MD*) es la máxima diferencia existente entre píxeles de ambas imágenes.

$$MD = \text{Max} (| (a_{i,j}) - (a'_{i,j}) |).$$

4.1.7. Error Absoluto Medio

El Error Normalizado Medio (*ENM*), o *Normalized Absolute Error*, (*NAE*) de define como la media aritmética de los errores absolutos cometidos.

$$NAE = \frac{\sum_{i=1}^M \sum_{j=1}^N |(a_{i,j}) - (a'_{i,j})|}{\sum_{i=1}^M \sum_{j=1}^N |(a_{i,j})|}.$$

4.2. Experimentos realizados

4.2.1. Introducción

En esta sección se plantea el uso de diversos tipos de imágenes para evaluar el grado de optimalidad de las técnicas anteriormente descritas. Además de las imágenes en formato DICOM¹, se han utilizado imágenes de otra naturaleza con el fin de que los resultados sean representativos de otros tipos de imagen, y de esta manera puedan ser utilizados como referencia en futuros estudios

El *software* utilizado para la codificación de las funciones y la realización de las funciones ha sido `Matlab R2010b` por presentar las funcionalidades necesarias para llevar a cabo con eficiencia esta serie de experimentos.

4.2.2. Descripción y procedimientos

Descripción de las imágenes y del experimento

A continuación se describen las diez imágenes involucradas en los experimentos realizados. Dada la imposibilidad de poder abarcar un gran número de imágenes distintas, se ha tenido que llevar a cabo una estricta selección de la muestra.

Por un lado, en el experimento han sido incluidas dos imágenes en formato DICOM denominadas `Dicom1` y `Dicom2`, por corresponder al tipo de imágenes a las que va dirigido este estudio. `Lena` y `Geo`, se ha tomado como representativas del tipo de imágenes utilizadas cotidianamente, como son las fotografías etc. Las restantes imágenes corresponden a imágenes médicas de distinto tipo: se han incluido dos de TAC (`Tac1` y `Tac2`), dos de PET (`PET1` y `Pet2`), y dos imágenes híbridas de PET y tomografía computerizada (`Pet-Tc1` y `Pet-Tc2`).

Todas las imágenes pertenecientes a la citada muestra están codificadas en escala de grises, es decir, que para la representación de las diversas tonalidades de gris se emplean 8 bits por píxel, dando como resultado $2^8 = 256$ tonalidades de gris existentes en la imagen, y codificadas mediante valores comprendidos en el rango $[0, 255]$.

`Dicom1` y `Dicom2`

Las imágenes a la que se les ha denominado `DicomX` (ver figura 4.1a y 4.1b) son imágenes de en formato DICOM, extraídas de una gammagrafía obtenida en el diagnóstico del neuroblastoma.

¹En las imágenes en formato DICOM expuestas ha sido eliminada cualquier posible dato o información sensible de proporcionar información acerca de la identidad del paciente, lugar de toma de la imagen, etc; de acuerdo con lo dispuesto en la Ley Orgánica 15/1999 de 13 de diciembre de Protección de Datos de Carácter Personal.

La imagen inicial consta de cuatro regiones que muestran las diferentes zonas que son objeto del estudio: la parte anterior del cuerpo, la parte posterior del mismo, y la partes pélvica y abdominal en detalle, tanto anterior como posterior.

Después del procesado de esta gammagrafía, se ha obtenido la imagen *Dicom1* que muestra la parte posterior del cuerpo del paciente y abarca 344×174 píxeles; y la imagen *Dicom2* que muestra su parte anterior mediante una ventana de 344×174 .

Lena

La imagen *Lena.jpg* (ver figura 4.1c) es una imagen de 512×512 píxeles. Es la típica imagen utilizada como referencia en estudios relacionados con imágenes digitales y a ello se debe su inclusión en este estudio.

Geo

La imagen *Geo.jpg* (ver figura 4.1d) es una imagen de 1024×1024 píxeles. Corresponde a una imagen de satélite del planeta Tierra. Dada la geometría del globo y las tonalidades que los efectos meteorológicos producen en su representación, se ha decidido incluirla en este experimento

Tac1

La imagen *Tac.jpg* (ver figura 4.2a) es una imagen médica de dimensiones 1228×908 píxeles. Corresponde a una imagen de TAC cerebral de un paciente de 75 años, ingresado por síndrome motor con hemipresia derecha leve. El corte supratentorial presenta dos lagunas o infartos lacunares; una talámica izquierda (mayor) y otra derecha puntiforme.

Tac2

La imagen *Tac2.jpg* (ver figura 4.2b) es una imagen también de TAC, de 1800×1284 píxeles. Concretamente se trata de un corte superior de la zona abdominal.

Pet-Tc1

La imagen *Pet-Tc1.jpg* (ver figura 4.2e) es una imagen de dimensiones 1800×1588 . Ésta es de tipo híbrido a partir de una imagen PET y una de Tomografía computerizada.

Pet-Tc2

La imagen `Pet-Ct2.jpg` (ver figura 4.2f) es una imagen de tamaño 1200×797 píxeles, y es también una imagen híbrida, fruto de la misma combinación de técnicas que la anterior. Corresponde concretamente a una paciente de 38 años que ingresó tras sufrir dolores en la cadera izquierda y a la que previamente se le había detectado un carcinoma en las células renales.

Pet1

La imagen `Pet1.jpg` (ver figura 4.2e) es una imagen de dimensiones 430×430 . Es una imagen de PET de la que no se dispone más información.

Pet2

La imagen `Pet2.jpg` (ver figura 4.2f) es una imagen de 400×541 píxeles, y corresponde a una imagen coronal tomografía por Emisión de Positrones (PET) de cuerpo completo. El paciente es un varón de 76 años en estudio por molestias articulares.



(a)

(b)



(c)



(d)

Figura 4.1: Imágenes anterior (a) y posterior (b) de un paciente, denominadas respectivamente Dicom1 y Dicom2. Imágenes fotográficas Lena.jpg (a) y Geo.jpg (b).

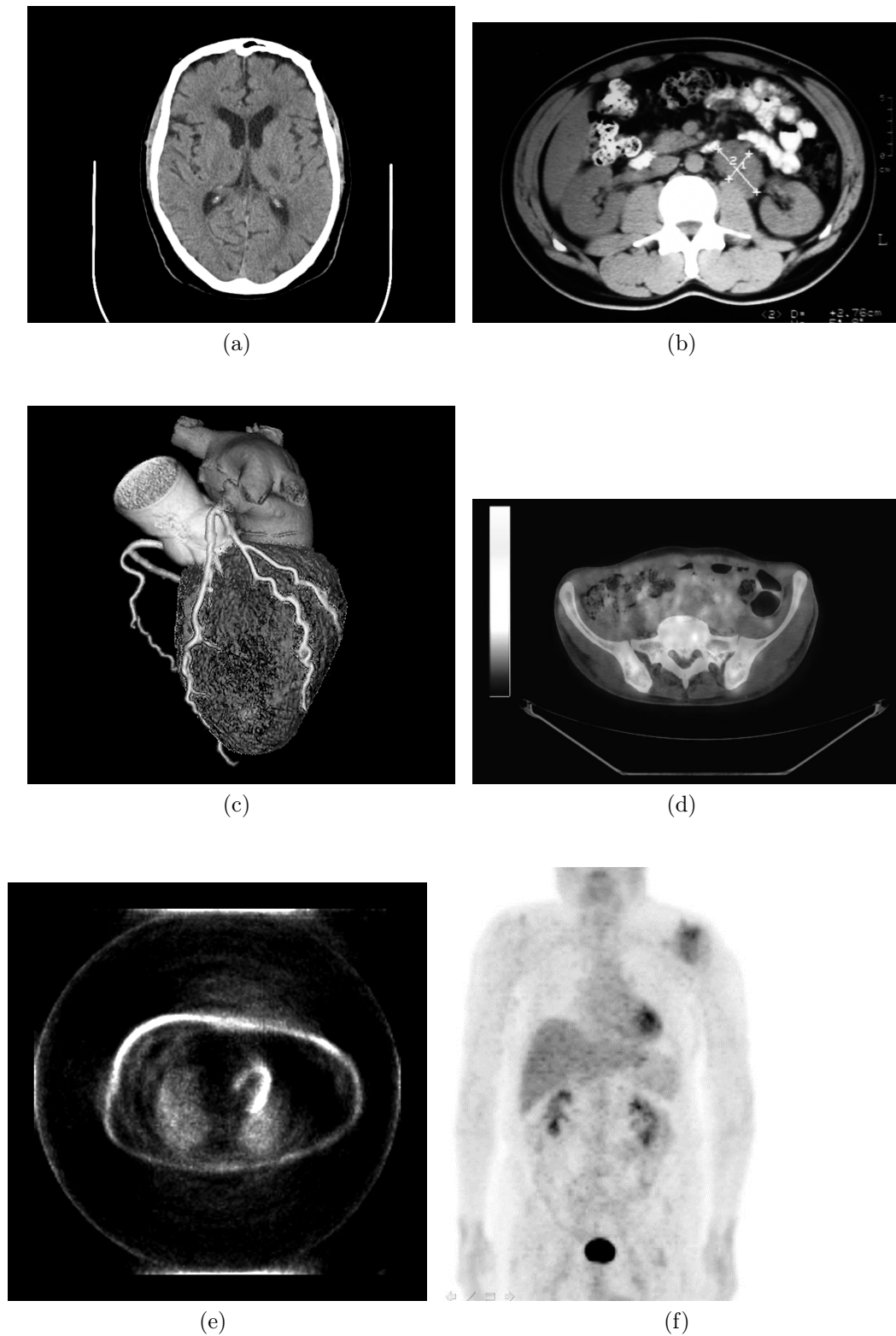


Figura 4.2: Resto de imágenes utilizadas en los experimentos. Lena.jpg (a), Tac1.jpg (b), Tac2.jpg (c), Pet-Tc1.jpg (d), Pet-Ct2.jpg (e), Pet1.jpg (f) y Pet2.jpg (g).

Procedimiento realizado

A continuación se explica el procedimiento que se ha seguido durante para llevar a cabo el experimento. En la sección (4.3) se encuentran los resultados que se han obtenido para las distintas técnicas, descritos mediante los indicadores de calidad de imágenes digitales, de los cuales ya se ha hablado anteriormente.

Partimos de una imagen en forma matricial que denotaremos formalmente como $A := (a_{i,j})_{m \times n}$, donde cada $a_{i,j} \in [0, \dots, 255]$, por estar codificadas en escala de grises. Seguidamente se enuncian los pasos efectuados.

1. En primer lugar, se ha obtenido A' a partir de A . A' representa una reducción de la matriz original y queda definida como $A' := (a'_{i,j})_{\frac{m}{k} \times \frac{n}{k}}$; entiendo k como el factor de reducción de la imagen inicial. Los elementos de A' , por tanto, mantienen la equivalencia con los de A como sigue: $A' := (a'_{i,j}) = (a_{\frac{i}{k}, \frac{j}{k}})$ siendo $k = 2l$ y $l = 1, \dots, z$. Por motivos de coste computacional y de factibilidad en el momento de realización de las pruebas, se ha decidido acotar los valores de l , incluyendo la restricción $z \leq 8$.
2. Seguidamente, si se define una técnica de interpolación mediante una función genérica denotada por $\mathcal{I} : \mathbb{R}^y \rightarrow \mathbb{R}^y$ se tiene que, el resultado de aplicar \mathcal{I} sobre la matriz anterior A' es $\mathcal{I}(A') = \tilde{A}$. Resulta obvio que en cada caso el coste computacional y la calidad de la nueva matriz obtenida variará según la técnica, tal y como se desprende de las operaciones y ponderaciones que cada una de éstas lleva a cabo.
Ésta última matriz, \tilde{A} , representa una aproximación de la imagen inicial, A .
3. En tercer lugar, una vez disponibles ambas matrices A y \tilde{A} , son comparadas mediante el indicador PSNR, que proporciona el nivel de similitud medido en decibelios.

Este procedimiento se ha repetido para cada valores de $k = 1 \dots 4$ y para cada una de las técnicas expuestas en el capítulo 3.

4.3. Resultados

A continuación se presentan los resultados obtenidos para cada factor de zoom y técnica.

Por un lado se muestra la tabla con los datos obtenidos mediante la evaluación de los indicadores comentados para cada técnica, y por otro, se muestran las imágenes obtenidas mediante la interpolación con uno y dos niveles de *zoom* respectivamente, es decir, las imágenes obtenidas a partir de la imagen original reducida a la mitad y a la cuarta parte, respectivamente.

En la columna de la izquierda de las siguientes tablas se encuentra el nombre de las técnicas estudiadas en el tercer capítulo de esta memoria. A modo de recordatorio.

ENO jerárquico ó **ENO_h** Interpolación ENO mediante selección jerárquica del *stencil* utilizando cuatro puntos (3.3.1).

ENO no jerárquico ó **ENO** Interpolación ENO mediante selección jerárquica del *stencil* utilizando cuatro puntos (3.3.1).

WENO Interpolación WENO *stencil* utilizando cuatro puntos (3.3.3).

ENO-SR Interpolación ENO-SR basada en la técnica ENO con selección jerárquica del *stencil* (3.3.2).

RACIONAL Interpolación utilizando cuatro puntos en la selección del *stencil* y un $\alpha = 0,5$ (3.3.4).

PPH Interpolación PPH (3.3.5).

Aquellas siglas o abreviaturas utilizadas en las columnas de cada tabla corresponden a los indicadores de calidad de imagen mencionados con anterioridad, y son los que a continuación se enumeran.

MSE Error Cuadrático Medio o *Mean Squared Error* (4.1.2).

PSNR Relación Señal a Ruido de Pico, o *Peak Signal to Noise Ratio* (4.1.3).

NCC Coeficiente de Correlación Normalizado, o *Normalized Cross Correlation* (4.1.4).

AD Diferencia Media, o *Average Difference* (4.1.5).

MD Diferencia Máxima, o *Maximum Difference* (4.1.6).

NAE Error Absoluto Normalizado, o *Normalized Absolute Error* (4.1.7).

En la sección ?? se exponen las conclusiones acerca de los resultados observados.

4.3.1. Resultados con la imagen *Dicom1*

Técnica	N	MSE	PSNR	NCC	AD	MD	NAE
ENO no jerárquico	1	120,6907	27,3141	0,9145	0,2300	102	0,3874
	2	141,1701	26,6334	0,8866	0,4619	128	0,4161
	3	204,7422	25,0186	0,8501	0,7106	155	0,4746
	4	275,9669	23,7222	0,7376	1,8971	182	0,5529
ENO jerárquico	1	134,8196	26,8310	0,9115	0,1981	102	0,4132
	2	154,7419	26,2347	0,8911	0,3765	128	0,4337
	3	217,0438	24,7653	0,8533	0,7455	149	0,4887
	4	295,4249	23,4249	0,7386	1,7789	180	0,5662
WENO	1	136,6975	26,7732	0,8979	0,4118	121	0,4050
	2	153,0548	26,2823	0,8476	0,8541	167	0,459
	3	208,0133	24,9499	0,7485	2,0822	191	0,4722
	4	353,2384	22,6624	0,5326	4,4272	229	0,5935
ENO-SR	1	136,9803	26,7642	0,9057	0,3680	105	0,4164
	2	157,5326	26,1571	0,8824	0,5962	128	0,4372
	3	219,9318	24,7079	0,8463	0,8533	151	0,4922
	4	307,1795	23,2569	0,7319	1,661	181	0,5840
RACIONAL	1	115,6464	27,3316	0,9020	0,5802	104	0,3818
	2	139,1557	26,6859	0,8586	1,0636	143	0,4034
	3	193,8547	25,2565	0,8034	1,5401	154	0,491
	4	318,4464	23,1004	0,6780	2,6364	189	0,5843
PPH	1	115,6464	27,4976	0,910	0,0660	101	0,3811
	2	138,1151	26,7284	0,8937	0,2510	123	0,4116
	3	196,3408	25,2007	0,8611	0,5099	151	0,4675
	4	291,5029	23,4844	0,7389	1,7228	178	0,5635

Tabla 4.1: Resultados numéricos para la imagen *Dicom1* mediante el uso de las diferentes técnicas con diferentes factores de *zoom*.

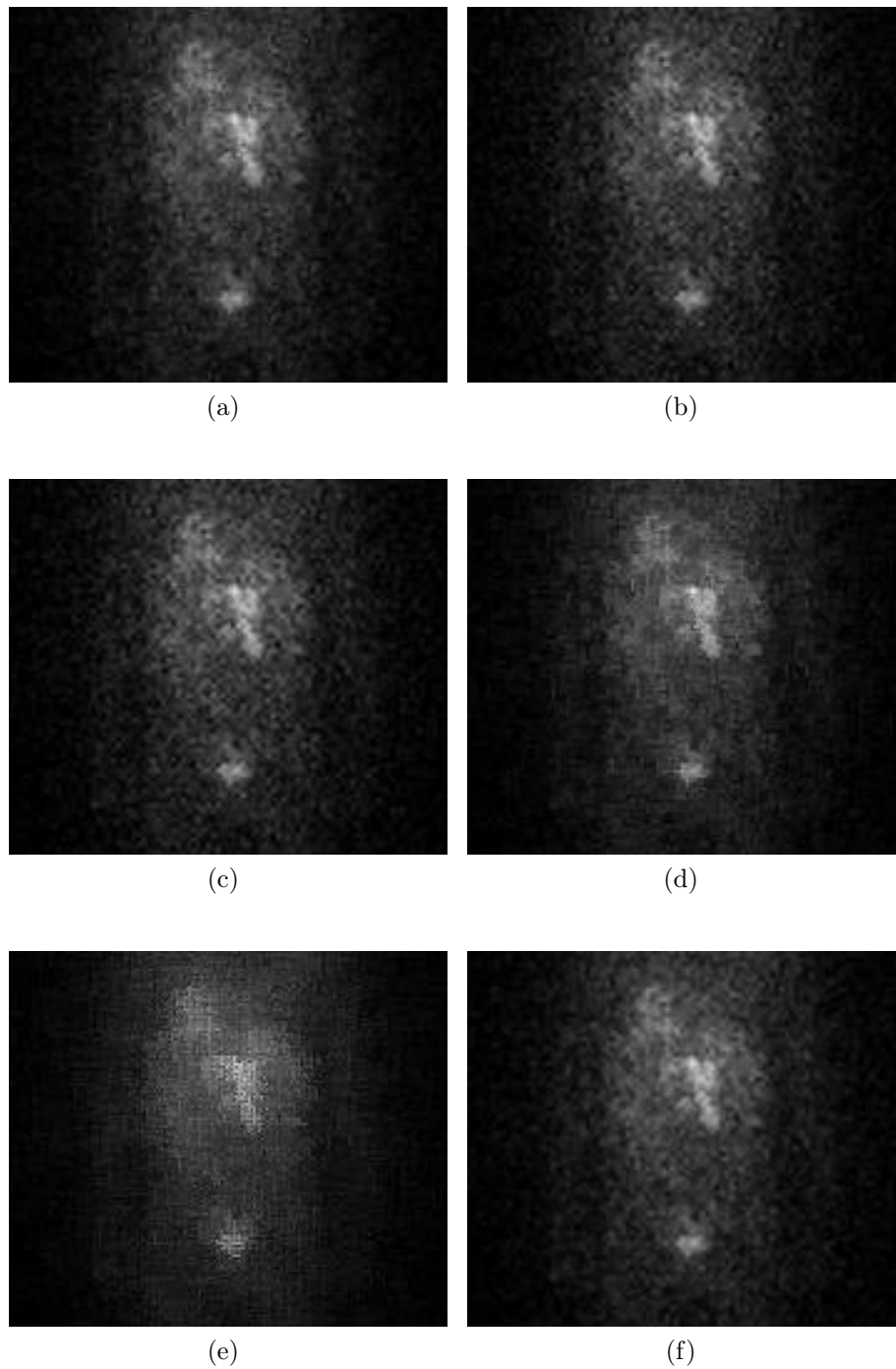


Figura 4.3: Imágenes *Dicom1* con un nivel de zoom ($\times 2$) acotadas por una ventana de 175×151 píxeles. Reconstrucción con ENO no jerárquico (*a*), con ENO jerárquico (*b*), con ENO-SR (*c*), con WENO (*d*), con Racional (*e*), y con la técnica PPH (*f*).

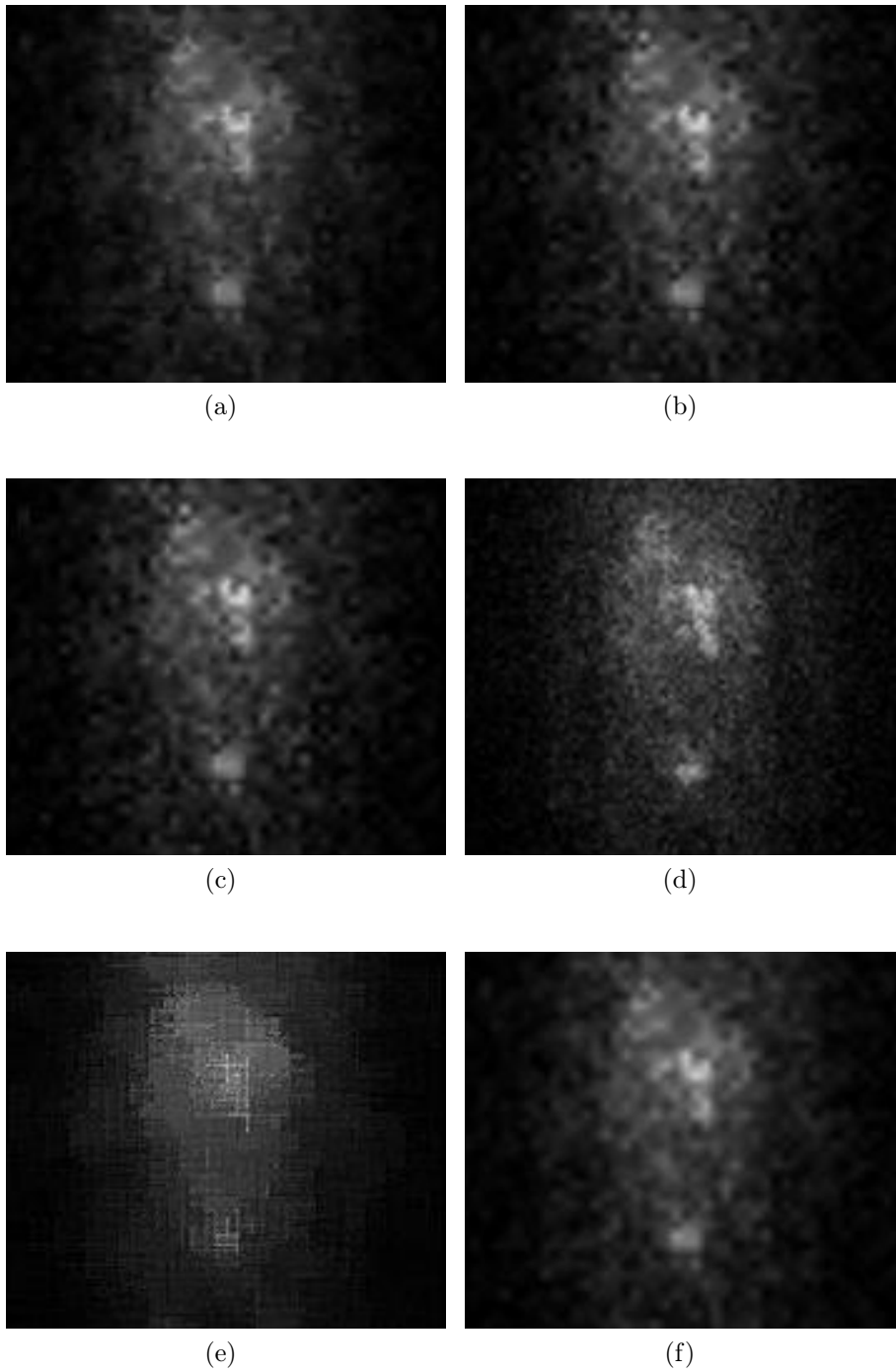


Figura 4.4: Imágenes Dicom1 con un nivel de 2 (x4) acotadas por una ventana de 175×151 píxeles. Reconstrucción con ENO no jerárquico (a), con ENO jérárquico (b), con ENO-SR (c), con WENO (d), con Racional (e), y con la técnica PPH (f).

4.3.2. Resultados con la imagen *Dicom2*

Técnica	N	MSE	PSNR	NCC	AD	MD	NAE
ENO no jerárquico	1	113,1933	27,5926	0,9304	0,0602	117	0,3693
	2	148,6390	26,4095	0,9043	0,0103	140	0,4133
	3	217,7340	24,7515	0,8450	0,1768	171	0,4803
	4	335,6572	22,8718	0,7044	1,1989	179	0,6042
ENO jerárquico	1	126,0167	27,1265	0,9283	0,00217	117	0,3942
	2	158,3730	26,1340	0,9108	-0,0824	140	0,4286
	3	231,6568	24,4824	0,8547	0,0439	187	0,4926
	4	350,8093	22,6801	0,7155	0,9933	177	0,6179
WENO	1	130,4950	26,9749	0,9106	0,2102	141	0,3897
	2	167,1330	25,9002	0,8567	0,3394	172	0,4181
	3	240,1387	24,3262	0,7464	1,3369	176	0,4849
	4	382,2118	22,3078	0,5660	2,6737	195	0,6401
ENO-SR	1	128,007	27,058	0,9227	0,1791	117	0,3977
	2	161,3301	26,0536	0,9048	0,1283	140	0,4317
	3	235,4351	24,4121	0,8472	0,2164	184	0,4973
	4	361,3813	22,551	0,7048	1,0844	177	0,6321
RACIONAL	1	112,9673	27,6013	0,9187	0,3563	117	0,3651
	2	147,5731	26,4407	0,8840	0,5129	155	0,4019
	3	207,6204	24,9581	0,7930	1,0537	171	0,430
	4	335,5185	22,8736	0,6203	2,4094	185	0,6066
PPH	1	108,5177	27,7758	-0,1370	-0,1370	117	0,3638
	2	144,9833	26,5176	0,9135	-0,2287	144	0,4091
	3	220,6592	24,6936	0,8538	-0,0245	174	0,4784
	4	352,1596	22,6634	0,7211	0,6919	179	0,6186

Tabla 4.2:

Resultados numéricos para la imagen *Dicom2* mediante el uso de las diferentes técnicas con diferentes factores de *zoom*.

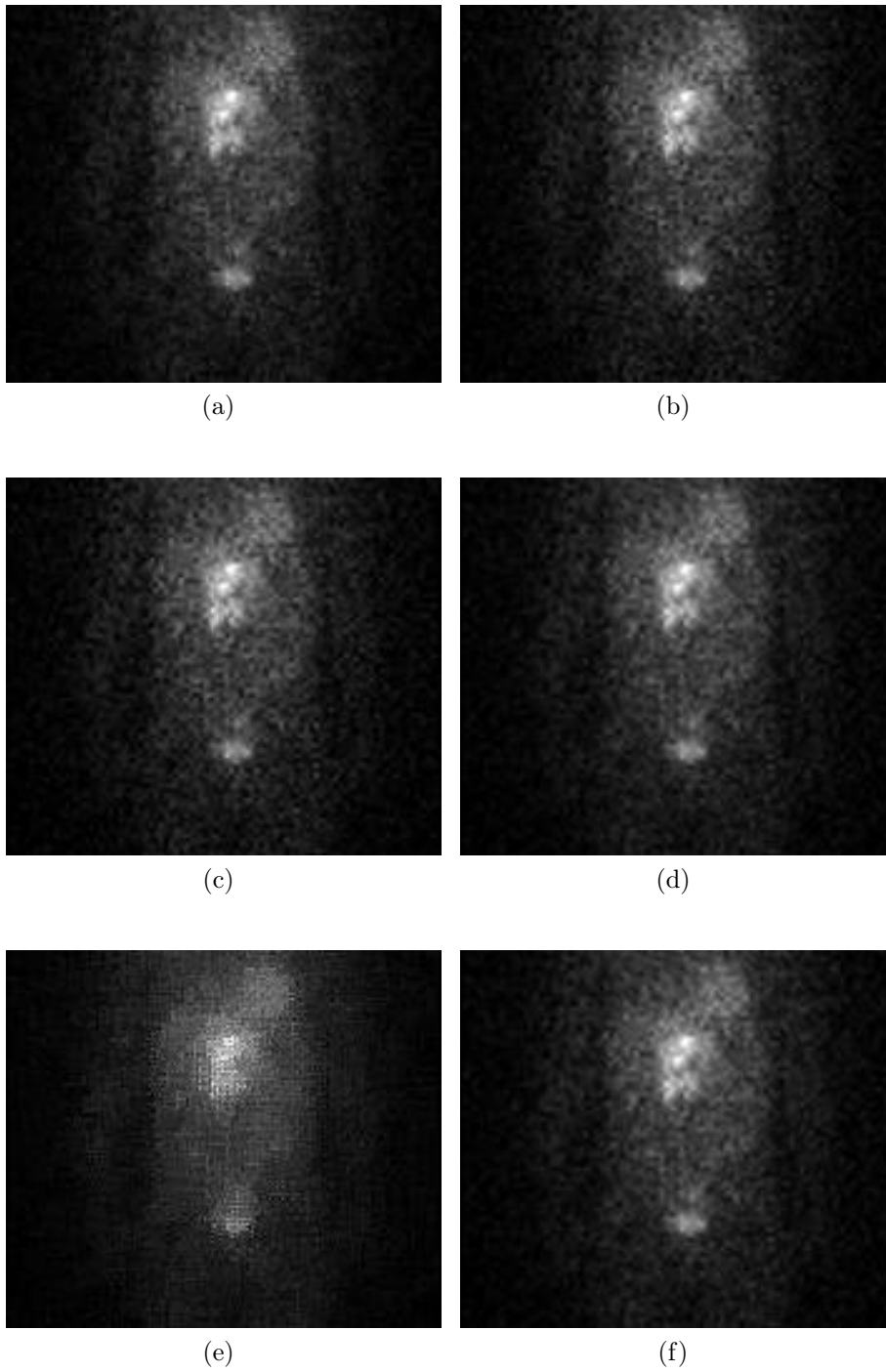


Figura 4.5: Imágenes Dicom2 con un nivel de zoom (x2) acotadas por una ventana de 174×151 píxeles. Reconstrucción con ENO no jerárquico (a), con ENO jerárquico (b), con ENO-SR (c), con WENO (d), con Racional (e), y con la técnica PPH (f).

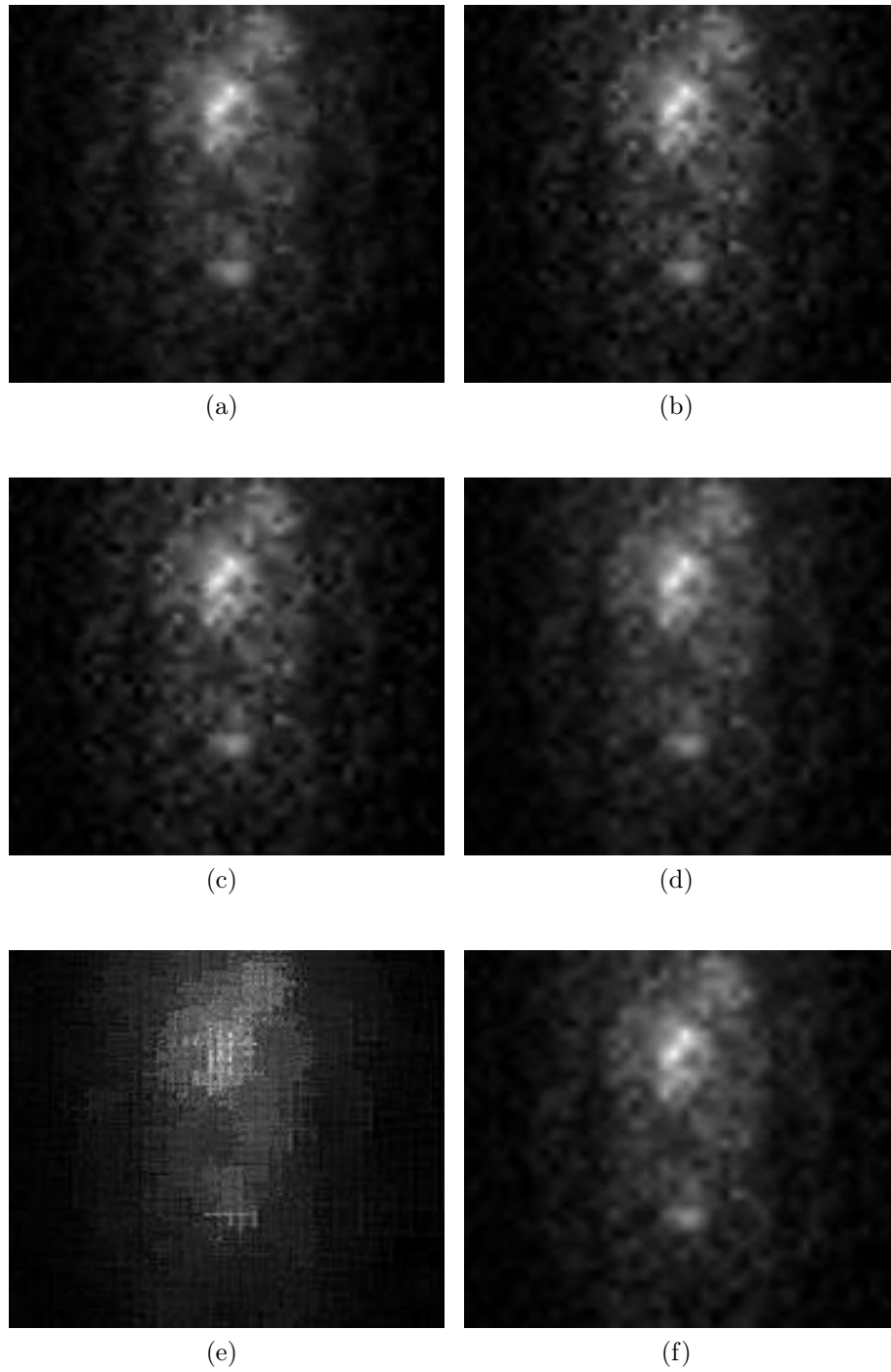


Figura 4.6: Imágenes Dicom2 con un nivel 2 (x4) de *zoom* acotadas por una ventana de 174×151 píxeles. Reconstrucción con ENO no jerárquico (*a*), con ENO jerárquico (*b*), con ENO-SR (*c*), con WENO (*d*), con Racional (*e*), y con la técnica PPH (*f*).

4.3.3. Resultados con la imagen *Lena*

Técnica	N	MSE	PSNR	NCC	AD	MD	NAE
ENO no jerárquico	1	61,7359	30,2254	0,9954	0,3050	189	0,0342
	2	283,7605	23,6013	0,9836	0,5736	189	0,0349
	3	712,4067	19,6035	0,9602	1,1324	216,076	0,1493
	4	1410	16,6035	0,9228	1,7535	214,9427	0,2349
ENO jerárquico	1	60,9001	30,2846	0,9956	6,2969	189	0,0391
	2	287,0612	23,5511	0,9842	0,5570	189	0,0780
	3	744,4507	19,4124	0,9615	1,0174	216,0762	0,1535
	4	1501	16,3656	0,9252	1,4454	216,5927	0,2422
WENO	1	157,5510	26,1565	1,0077	-0,2058	189	0,0636
	2	455,4448	21,5464	1,0132	2,9024	202	0,1189
	3	955,0030	18,3308	1,0143	5,3136	193	0,1850
	4	1770	15,7553	0,9974	-7,5572	191	0,2714
ENO-SR	1	61,1501	30,2668	1,0077	-1,2058	189	0,0636
	2	289,1822	23,5191	0,9839	0,5939	189	0,0882
	3	744,3638	19,4130	0,9608	1,1265	196	0,1544
	4	1513,7	16,3305	0,9246	1,4993	198	0,2456
RACIONAL	1	71,0149	29,6173	0,9968	0,2633	189	0,0353
	2	330,3484	22,9411	0,9848	0,7057	194	0,0872
	3	788,6687	19,1619	0,9634	1,1424	195	0,1484
	4	1410	16,3871	0,9308	1,5220	191	0,273
PPH	1	61,2942	30,2566	0,9961	0,2147	189	0,0341
	2	280,2441	23,6013	0,9840	0,4817	189	0,0863
	3	704,4169	19,6525	0,9607	0,9674	195	0,1491
	4	1397,2	16,678	0,9231	1,5848	191	0,2338

Tabla 4.3: Resultados numéricos para la imagen *Lena.jpg* mediante el uso de las diferentes técnicas con diferentes factores de *zoom*.

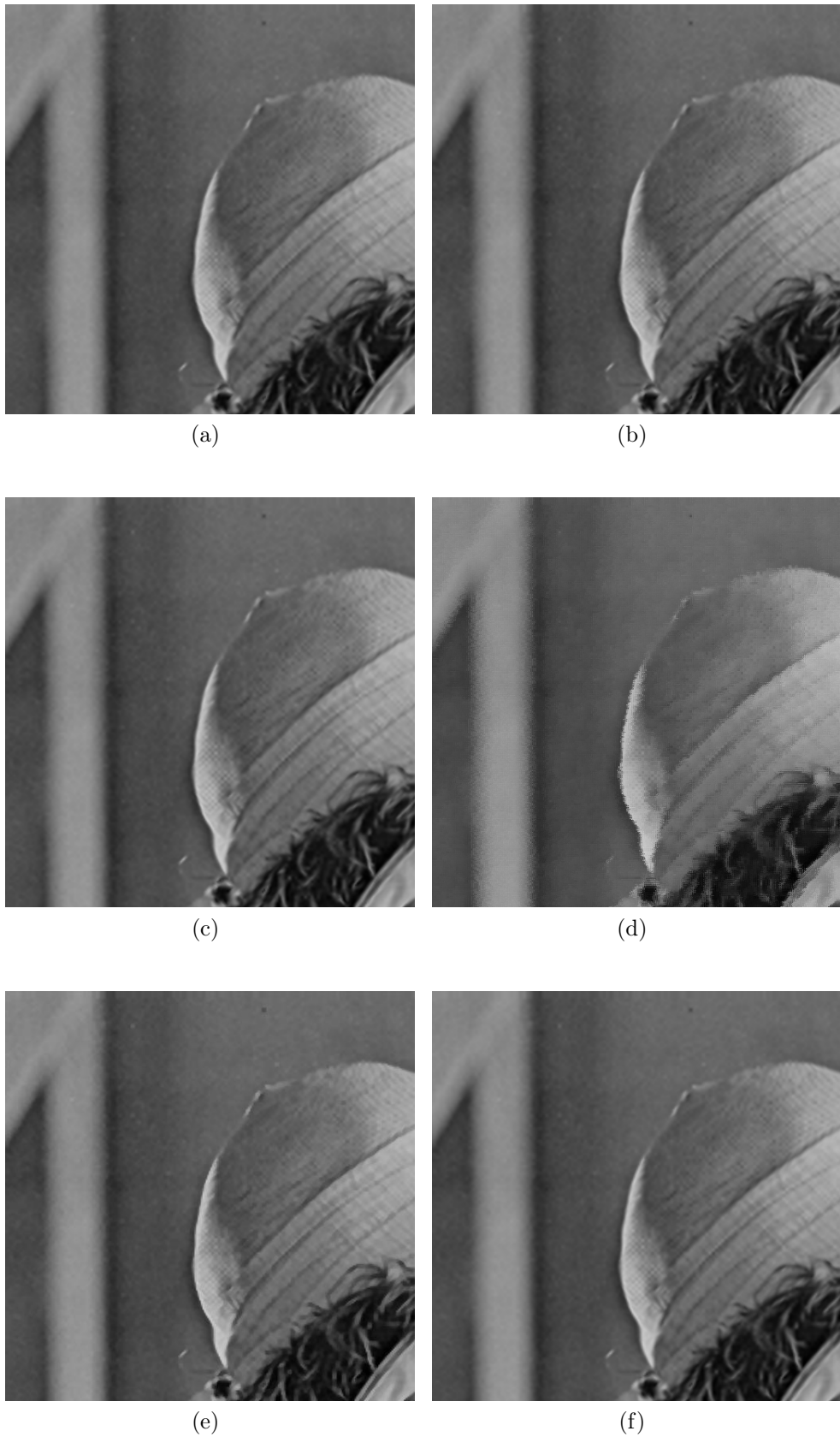


Figura 4.7: Imágenes *Lena.jpg* con un nivel de zoom (x2) acotadas por una ventana de 256×256 píxeles. Reconstrucción con ENO no jerárquico (a), con ENO jerárquico (b), con ENO-SR (c), con WENO (d), con Racional (e), y con la técnica PPH (f).

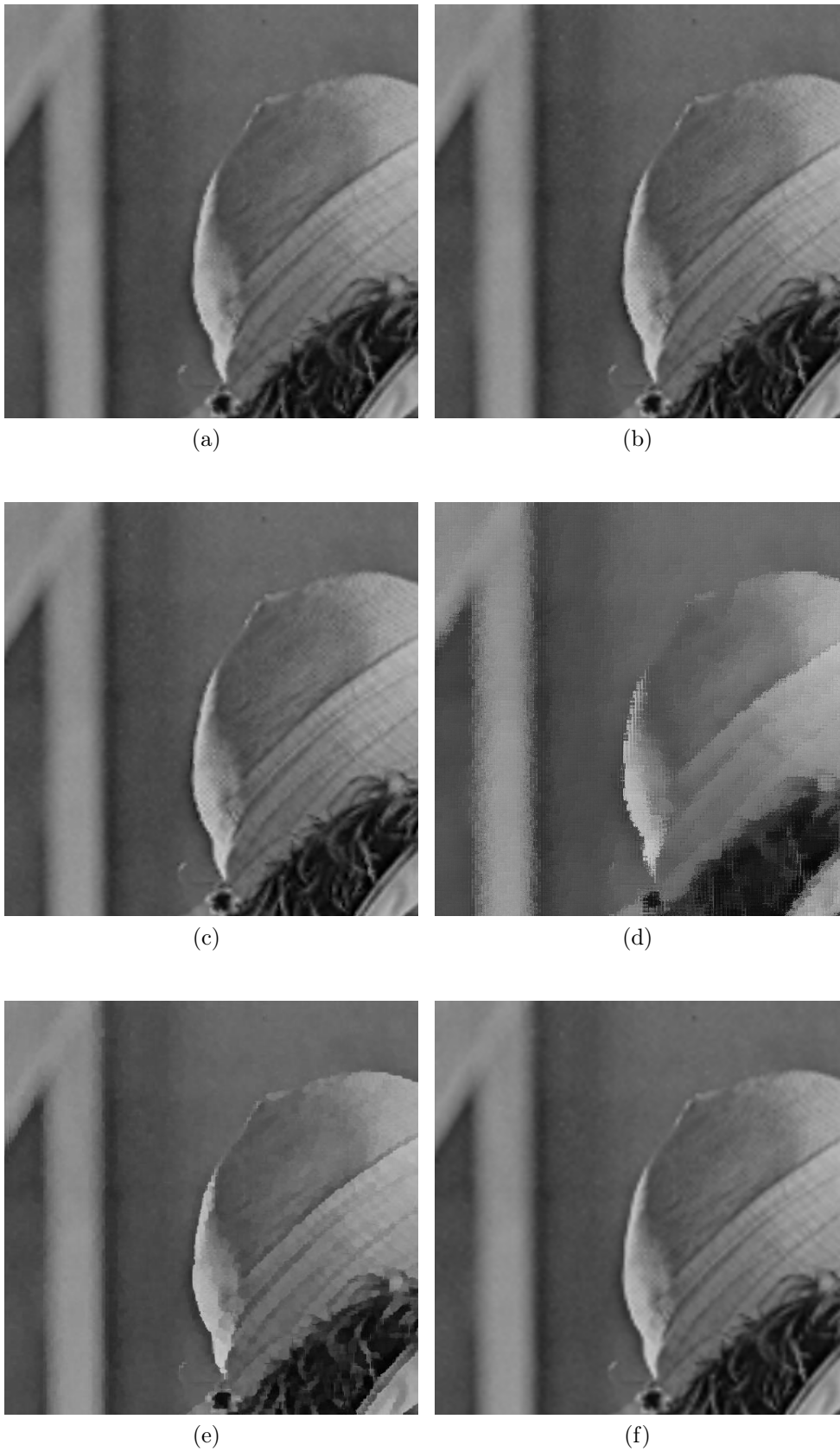


Figura 4.8: Imágenes *Lena.jpg* con un factor de zoom 2 (x4) acotadas por una ventana de 255×255 píxeles. Reconstrucción con ENO no jerárquico (a), con ENO jérárquico (b), con ENO-SR (c), con WENO (d), con Racional (e), y con la técnica PPH (f).

4.3.4. Resultados con la imagen *Geo*

Técnica	N	MSE	PSNR	NCC	AD	MD	NAE
ENO no jerárquico	1	331,5985	22,9247	0,9683	0,9686	225	0,1646
	2	661,5391	19,9252	0,9362	0,3100	241	0,2454
	3	1014,2	18,0696	0,9005	0,5461	244	0,3217
	4	1512,8	16,3071	0,8052	0,8797	251	0,4167
ENO jerárquico	1	389,1436	22,2297	0,9672	0,0356	225	0,1791
	2	2738,9232	19,4448	0,9363	0,2214	241	0,2596
	3	1123,5	17,6552	0,908	0,4567	247	0,3382
	4	1664,3	15,9710	0,8523	0,110	253	0,4317
WENO	1	508,75	21,0658	0,9664	-0,0097	248	0,2057
	2	817,8155	19,0043	0,9399	-0,0496	245	0,2722
	3	1704	17,4473	0,9054	0,3249	242	0,3409
	4	1512,8	16,307	0,8052	0,8797	251	0,4167
ENO-SR	1	396,4578	28,1488	0,9640	0,2660	225	0,1815
	2	752,5579	19,3654	0,9315	0,5757	243	0,2627
	3	1143,1	17,5499	0,8944	0,9151	250	0,3420
	4	1673,3	15,8952	0,8454	1,2764	254	0,4367
RACIONAL	1	364,0667	22,5190	0,9646	0,4856	225	0,1676
	2	696,6248	19,7008	0,9224	1,2421	239	0,3159
	3	1055,6	17,8960	0,8883	2,1221	247	0,3159
	4	1585	16,1294	0,8388	2,9108	247	0,04087
PPH	1	315,1721	23,1453	0,9693	-0,0574	225	0,1619
	2	650,2568	20,0000	0,9372	0,0734	243	0,2448
	3	1008,2	18,0955	0,9015	0,2115	243	0,3228
	4	1,4819	16,4225	0,8521	0,5361	241	0,4135

Tabla 4.4: Resultados numéricos para la imagen *Geo.jpg* mediante el uso de las diferentes técnicas con diferentes factores de *zoom*.

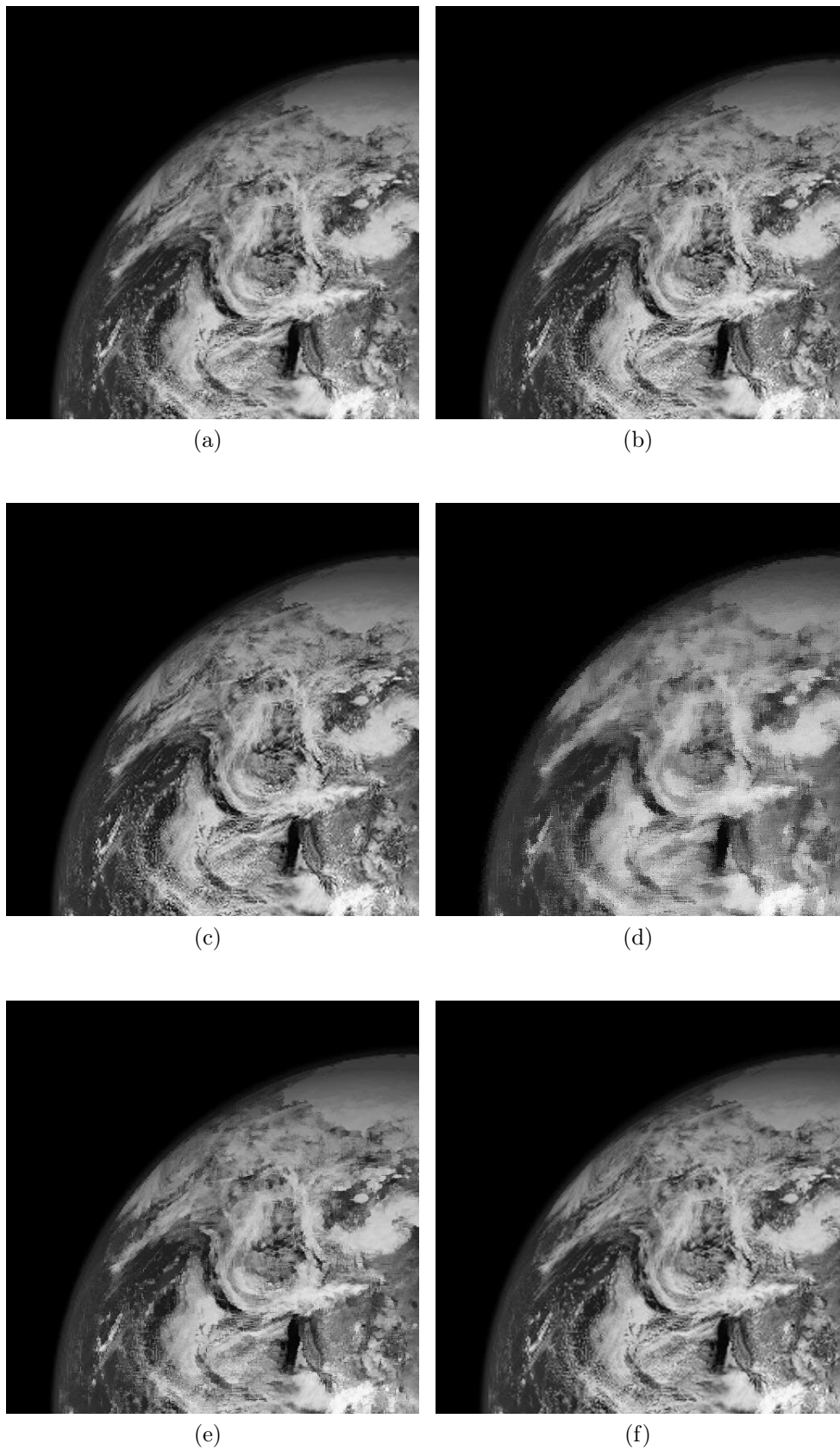


Figura 4.9: Imágenes *Geo.jpg* con un nivel de zoom (x2) acotadas por una ventana de 255×255 píxeles. Reconstrucción con ENO no jerárquico (a), con ENO jerárquico (b), con ENO-SR (c), con WENO (d), con Racional (e), y con la técnica PPH (f).

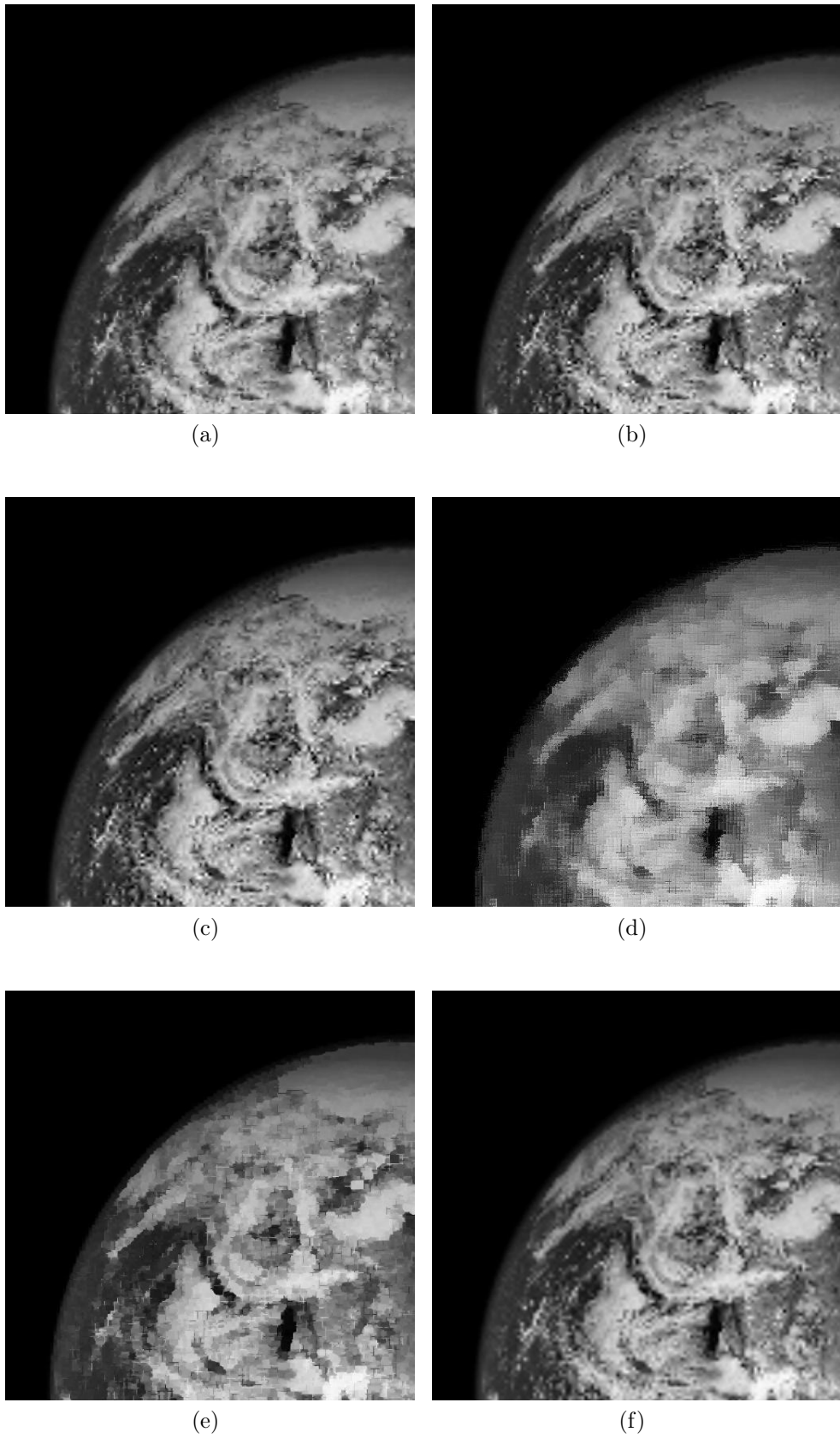


Figura 4.10: Imágenes *Geo.jpg* con un factor de zoom 2 (x4) acotadas por una ventana de 255×255 píxeles. Reconstrucción con ENO no jerárquico (a), con ENO jérárquico (b), con ENO-SR (c), con WENO (d), con Racional (e), y con la técnica PPH (f).

4.3.5. Resultados con la imagen *Tac1*

Técnica	N	MSE	PSNR	NCC	AD	MD	NAE
ENO no jerárquico	1	100,1344	28,1250	0,9904	0,050	236	0,0579
	2	534,6369	20,8502	0,9545	-0,0894	213,9609	0,1494
	3	1249,8	17,1625	0,8871	0,0206	245,9943	0,2663
	4	2377	14,3698	0,8034	-0,8391	214,8542	0,4303
ENO jerárquico	1	88,1348	28,6793	0,9920	0,0048	236	0,0558
	2	497,0134	21,1671	0,9562	0,0279	255	0,1460
	3	1258	17,1519	0,0881	0,0089	255	0,2661
	4	2371	14,4809	0,7880	0,1804	255	0,4256
WENO	1	260,5409	23,9720	0,9875	-0,3029	255	0,0924
	2	830,4847	18,9375	0,9375	-0,7333	255	0,1827
	3	1390	16,0990	0,8964	-0,9266	255	0,2899
	4	2481,1	14,1843	0,7547	1,5493	255	0,4179
ENO-SR	1	340,2642	24,8521	0,9875	-0,3079	255	0,0924
	2	710,36	19,5354	0,9327	-0,6258	255	0,0,1652
	3	1590	17,0064	0,8424	-0,4158	255	0,3654
	4	2651,1	14,0192	0,7547	1,5493	255	0,4179
RACIONAL	1	136,8917	26,7670	0,9946	-8,60e005	255	0,0602
	2	686,1479	19,7666	0,9569	0,1003	255	0,1504
	3	1447	16,5243	0,8974	0,5206	255	0,2519
	4	1454	14,3547	0,8223	0,9210	255	0,3691
PPH	1	99,5180	28,1518	0,9997	-0,01354	236	0,575
	2	524,6542	20,321	0,9530	-0,0665	255	0,1422
	3	1281	17,0532	0,8844	-0,2060	255	0,2702
	4	2434	14,2668	0,7864	-0,7832	255	0,4411

Tabla 4.5: Resultados numéricos para la imagen *Tac1.jpg* mediante el uso de las diferentes técnicas con diferentes factores de *zoom*.

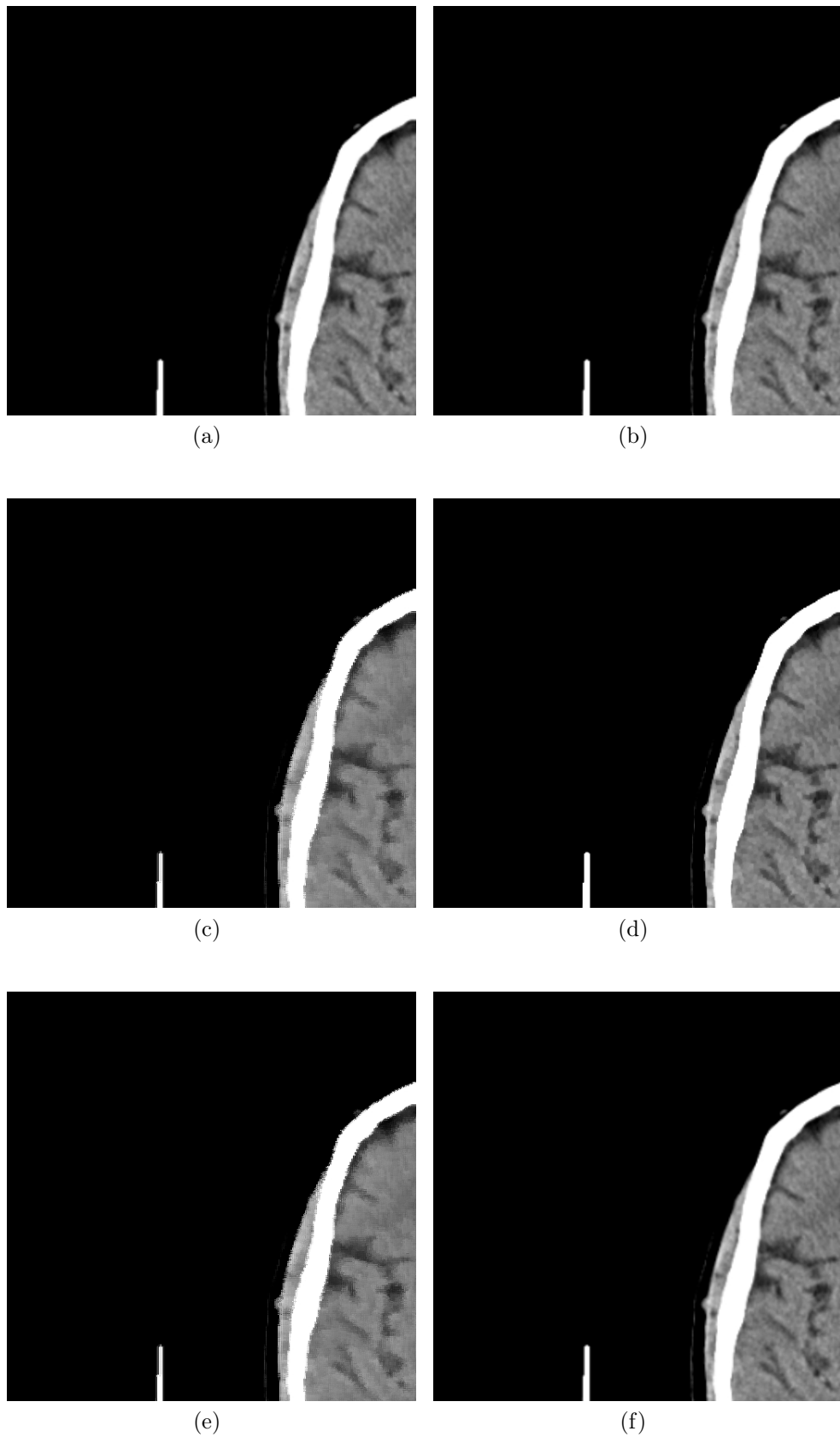


Figura 4.11: Imágenes Tac1.jpg con un 1 nivel de zoom (x2) acotadas por una ventana de 512×542 píxeles. Reconstrucción con ENO no jerárquico (a), con ENO jerárquico (b), con ENO-SR (c), con WENO (d), con Racional (e), y con la técnica PPH (f).

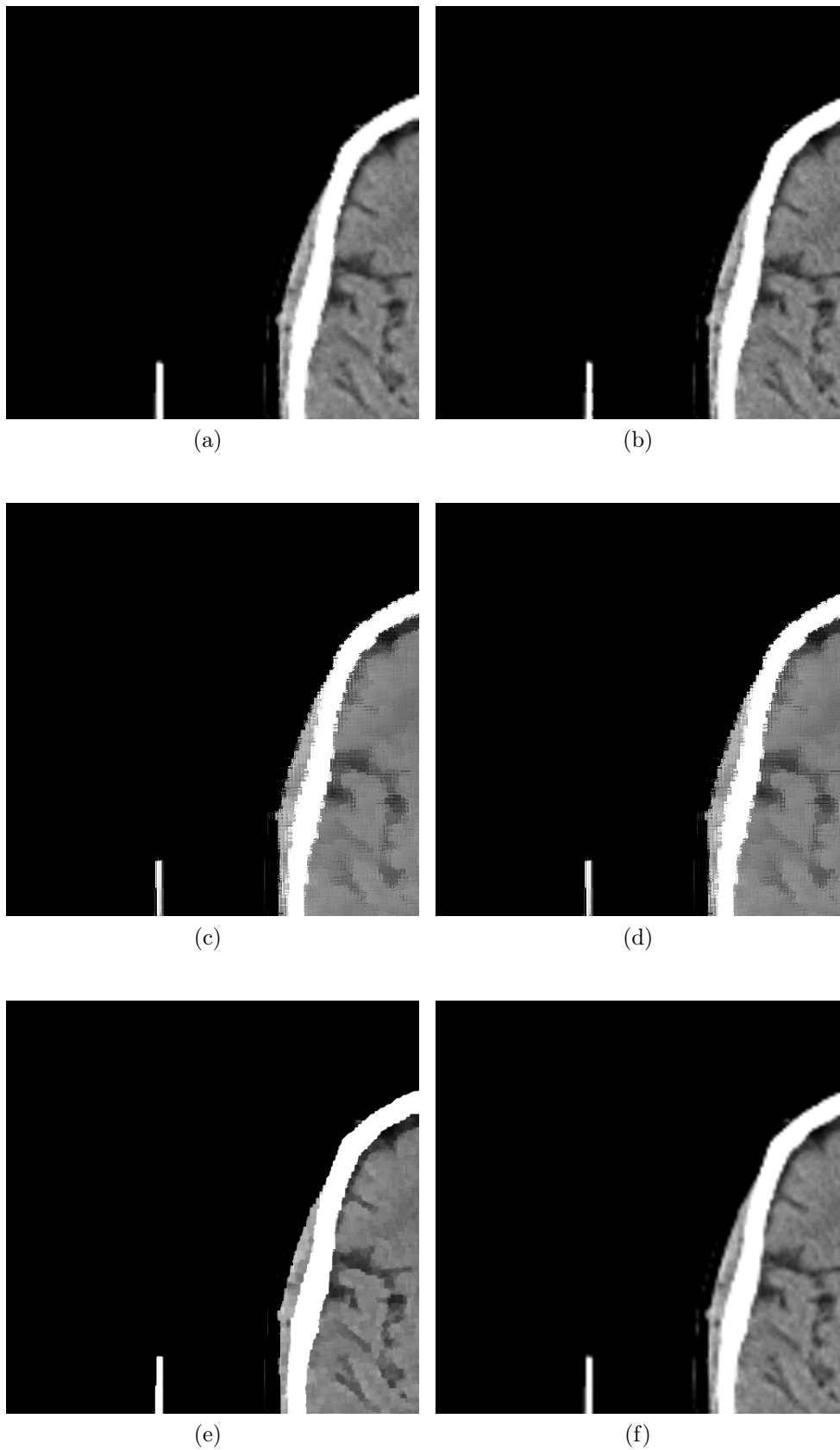


Figura 4.12: Imágenes Tac1.jpg con un factor de zoom 2 (x4) acotadas por una ventana de 512×542 píxeles. Reconstrucción con ENO no jerárquico (a), con ENO jerárquico (b), con ENO-SR (c), con WENO (d), con Racional (e), y con la técnica PPH (f).

4.3.6. Resultados con la imagen *Tac2*

Técnica	N	MSE	PSNR	NCC	AD	MD	NAE
ENO no jerárquico	1	39,3236	32,1843	0,9977	00,88	183	0,0455
	2	255,7191	24,0532	0,9854	0,0401	195	0,1146
	3	844,9424	24,0532	0,9490	0,2619	243	0,2150
	4	1717	15,7816	0,8931	0,5474	249	0,3436
ENO jerárquico	1	38,3645	32,2915	0,9979	0,0047	183	0,0456
	2	251,9034	24,1185	0,4864	0,0336	195	0,1140
	3	852,4802	18,8240	0,9504	0,3015	243	0,2190
	4	1830	15,4913	0,8940	0,4101	255	0,354
WENO	1	154,4015	26,2443	1,602	-0,3554	221	0,0831
	2	528,6831	20,8998	0,9914	-0,3181	246	0,1593
	3	1250	17,1587	0,9648	-0,3181	246	0,1593
	4	2870	14,9356	0,9302	-2,1281	225	0,3710
ENO-SR	1	38,5038	32,2758	0,9978	0,0108	183	0,0457
	2	252,2411	26,1122	0,9864	0,0355	195	0,1142
	3	654,9497	20,8114	0,95	0,3769	210	0,2198
	4	1840	16,4653	0,8914	6,6097	250	0,3563
RACIONAL	1	47,8495	31,3324	0,9993	0,0384	185	0,0473
	2	331,3535	22,9279	0,9892	0,2011	238	0,1191
	3	1043	17,9757	1,3236	0,9494	243	0,2223
	4	1844	15,3568	0,9080	1,7185	247	0,3277
PPH	1	38,7738	32,2954	0,9981	-0,0609	183	0,0453
	2	253,8404	26,0852	0,9854	-0,0182	195	0,1144
	3	838,55	20,8955	0,9493	0,0624	243	0,2193
	4	1748,6	16,7164	0,8406	0,528	249	0,3493

Tabla 4.6: Resultados numéricos para la imagen *Tac2.jpg* mediante el uso de las diferentes técnicas con diferentes factores de *zoom*.

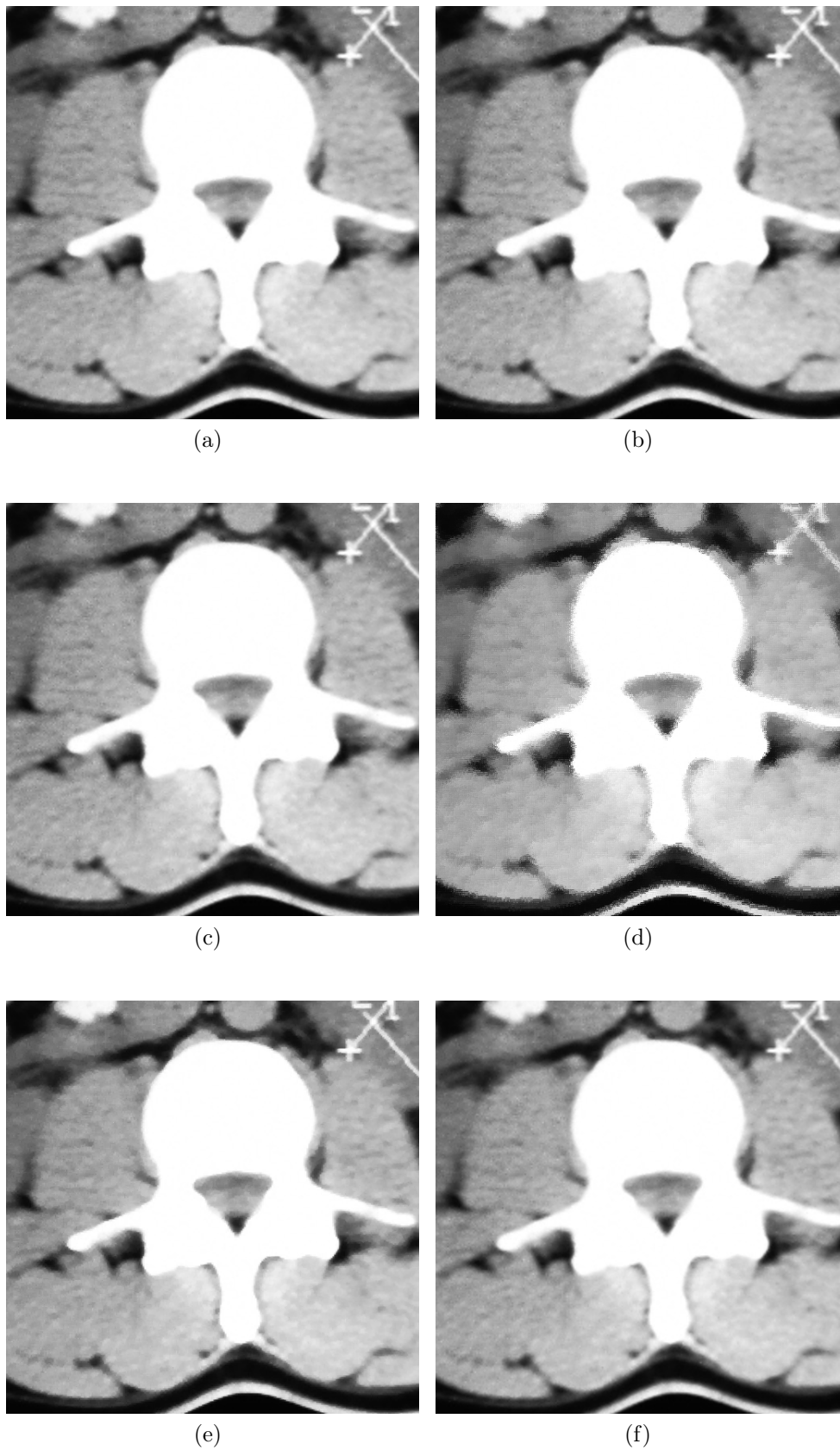


Figura 4.13: Imágenes Tac2.jpg con un 1 nivel de zoom (x2) acotadas por una ventana de 512×512 píxeles. Reconstrucción con ENO no jerárquico (a), con ENO jérárquico (b), con ENO-SR (c), con WENO (d), con Racional (e), y con la técnica PPH (f).

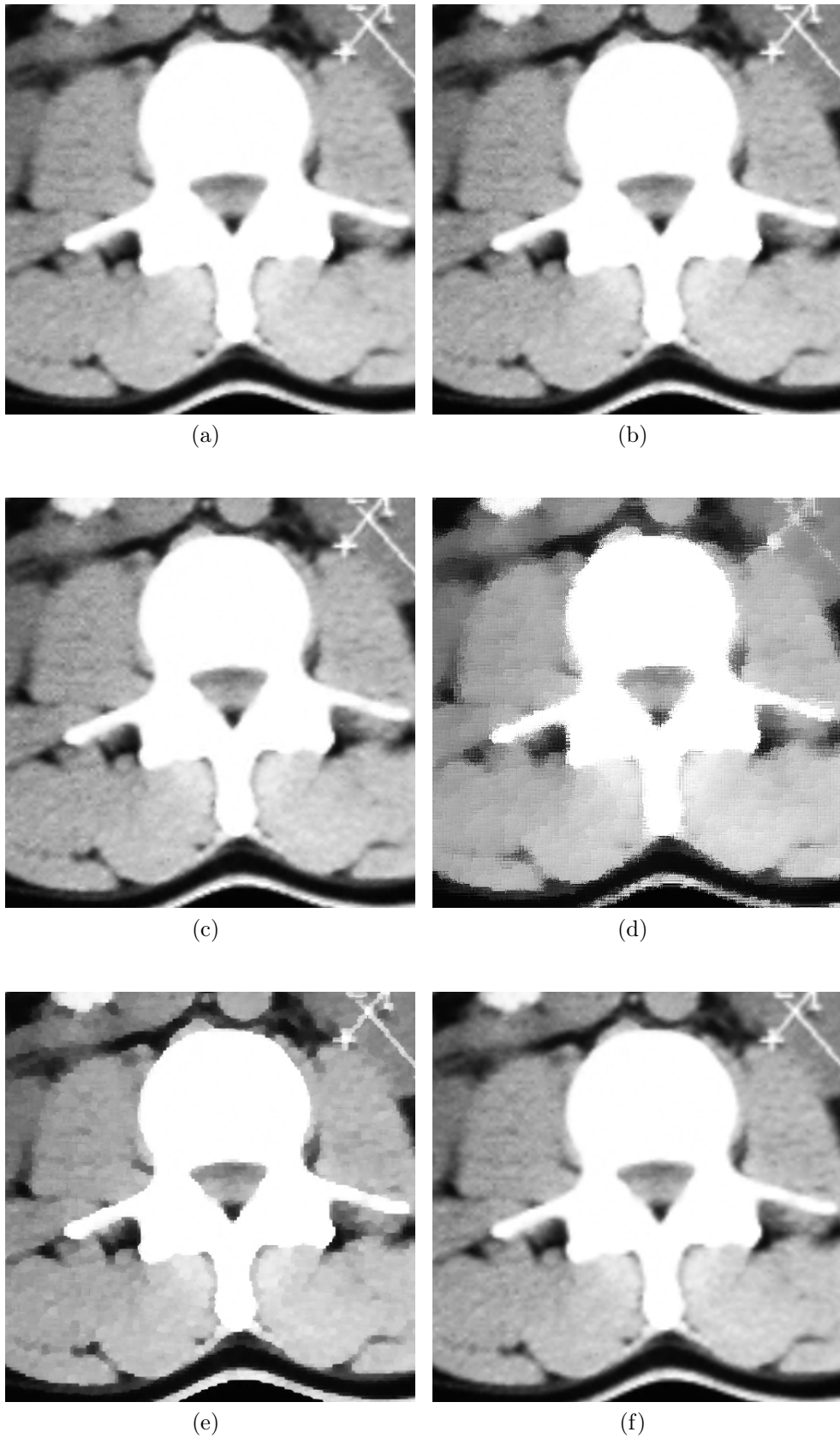


Figura 4.14: Imágenes Tac2.jpg con un factor de zoom 2 (x4) acotadas por una ventana de 512×512 píxeles. Reconstrucción con ENO no jerárquico (a), con ENO jerárquico (b), con ENO-SR (c), con WENO (d), con Racional (e), y con la técnica PPH (f).

4.3.7. Resultados con la imagen *Pet-Tc1*

Técnica	N	MSE	PSNR	NCC	AD	MD	NAE
ENO no jerárquico	1	44,9343	31,6050	0,9939	0,0013	142	0,0783
	2	224,1021	24,6263	0,9683	0,01184	238	0,1810
	3	4740,2337	21,3709	0,9330	-0,0259	238	0,2713
	4	781,2310	19,2030	0,87779	0,3312	245	0,3544
ENO jerárquico	1	44,3473	31,6221	0,9943	$-0,000125 * 10^{-5}$	150	0,0780
	2	244,3708	24,2503	0,9681	-0,0080	238	0,1882
	3	518,0333	20,9872	0,9311	0,03377	248	0,845
	4	837,5324	18,9008	0,881	0,2912	246	0,3726
WENO	1	174,4752	25,7135	0,9927	-0,5268	237	0,1475
	2	368,7084	22,4640	0,9804	-0,09805	243	0,2286
	3	587,9350	20,4375	0,9486	-,9213	249	0,2977
	4	894,1853	18,6165	0,8838	-0,0439	252	0,3805
ENO-SR	1	44,5924	31,6382	0,9945	-0,80	150	0,0782
	2	247,1329	24,2019	0,9673	0,0394	239	0,1894
	3	524,3784	20,9344	0,9282	0,1813	247	0,3758
	4	848,7539	18,8430	0,8746	0,5349	247	0,3758
RACIONAL	1	56,4816	30,6117	0,9961	0,0441	203	0,0836
	2	257,9580	24,0153	0,9706	0,1312	239	0,1872
	3	514,5613	21,0164	0,9343	0,2211	244	0,0705
	4	831,2260	18,9336	0,8525	1,9148	255	0,355
PPH	1	44,1232	31,6841	0,9940	-0,0247	141	0,0779
	2	221,1880	24,6832	0,9679	-0,0142	238	0,1803
	3	416,7993	21,4209	0,9303	-0,0086	238	0,2699
	4	772,9352	19,2499	0,8804	0,0891	244	0,3585

Tabla 4.7: Resultados numéricos para la imagen *Pet-Tc1.jpg* mediante el uso de las diferentes técnicas con diferentes factores de *zoom*.

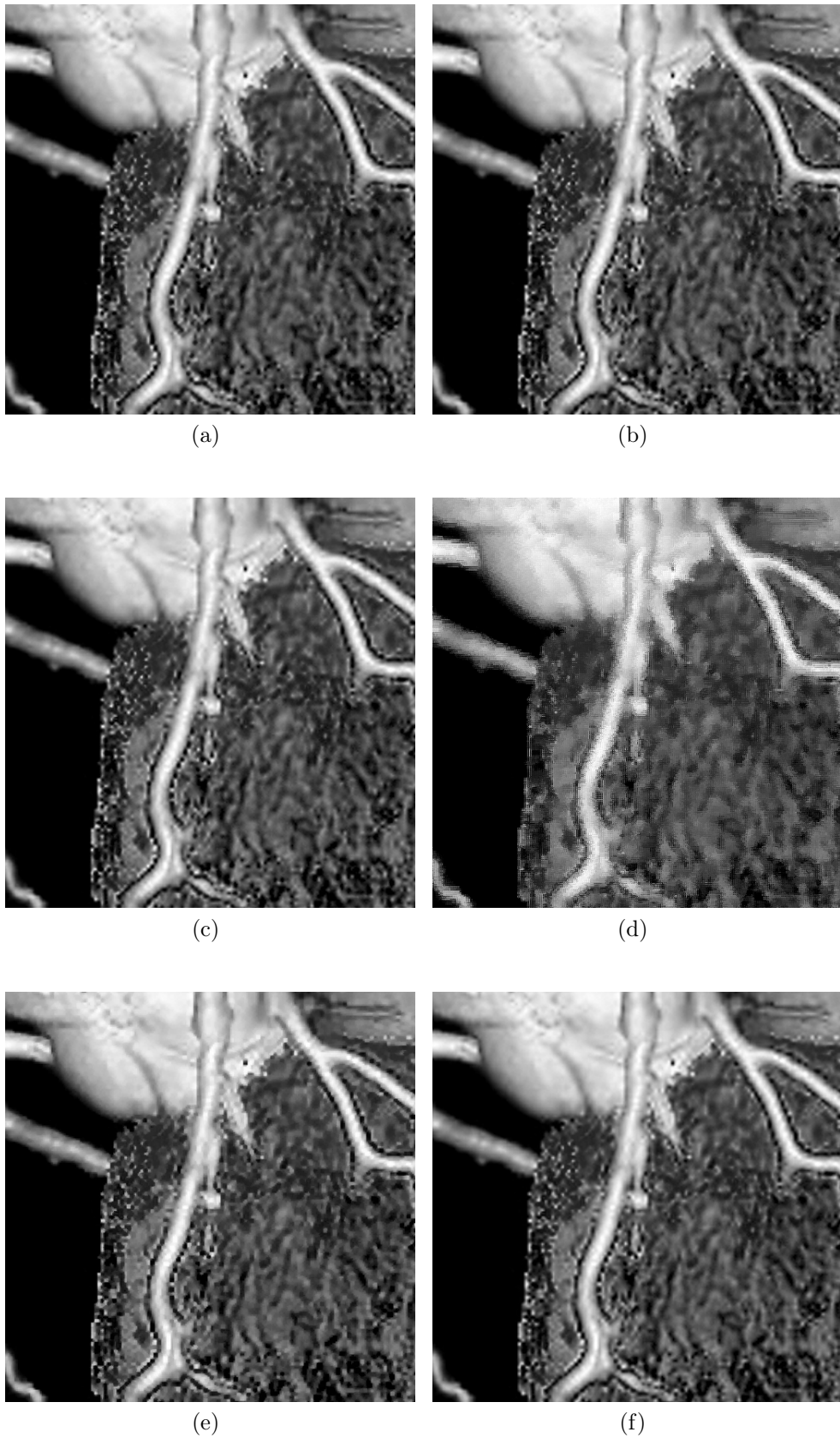


Figura 4.15: Imágenes *Pet1.jpg* con un nivel de zoom ($\times 2$) acotadas por una ventana de 961×512 píxeles. Reconstrucción con ENO no jerárquico (*a*), con ENO jerárquico (*b*), con ENO-SR (*c*), con WENO (*d*), con Racional (*e*), y con la técnica PPH (*f*).

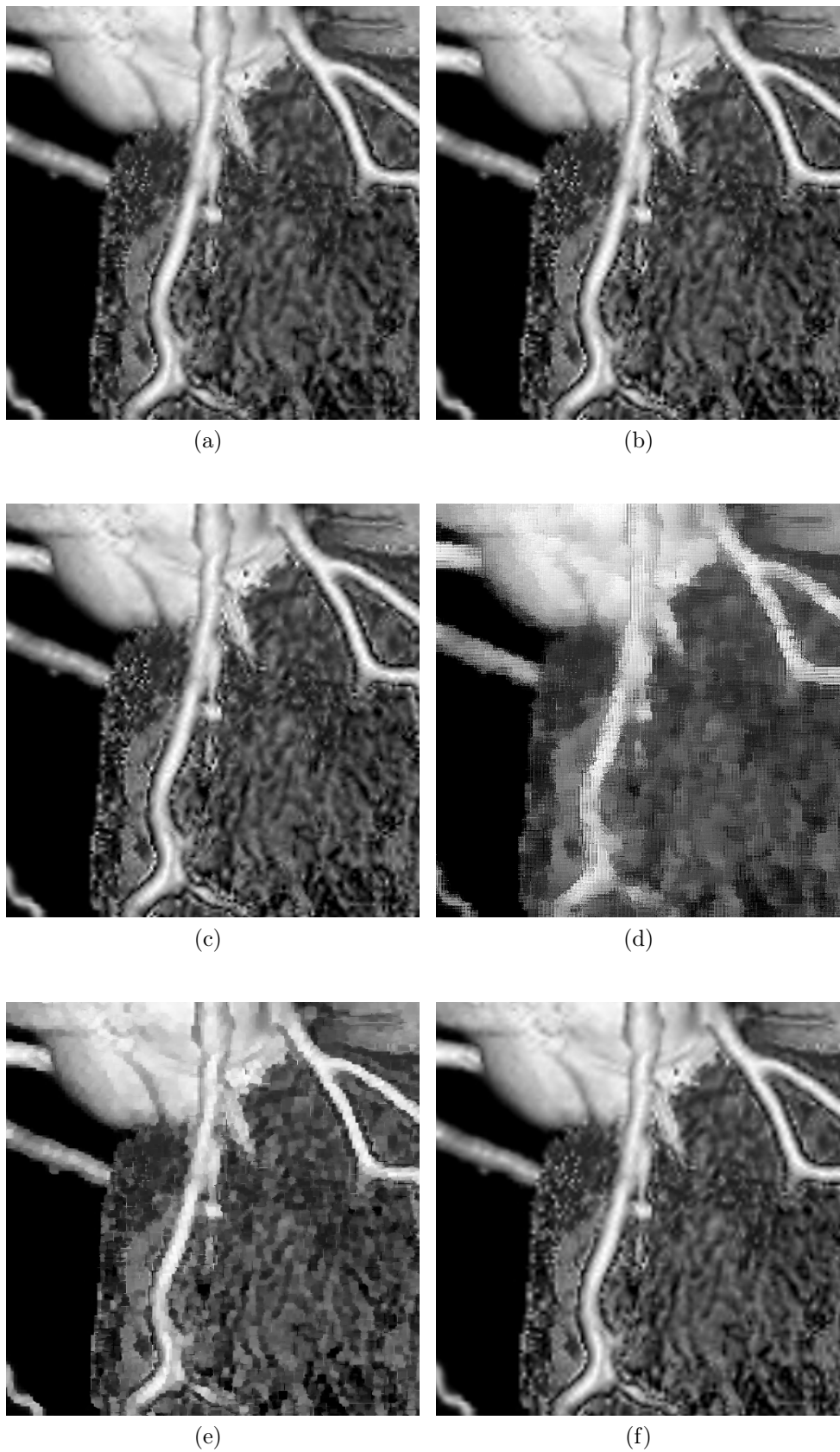


Figura 4.16: Imágenes *Pet-Tc1.jpg* con un factor de zoom 2 (x4) acotadas por una ventana de 961×512 píxeles. Reconstrucción con ENO no jerárquico (a), con ENO jerárquico (b), con ENO-SR (c), con WENO (d), con Racional (e), y con la técnica PPH (f).

4.3.8. Resultados con la imagen *Pet-Tc2*

Técnica	N	MSE	PSNR	NCC	AD	MD	NAE
ENO no jerárquico	1	32,1185	38,0633	0,9960	0,0059	109	0,0393
	2	170,4518	25,8141	0,9770	0,2311	214	0,0988
	3	398,0473	25,8148	0,9508	0,3511	245	0,1837
	4	983,8310	18,2016	0,8582	1,5086	250	0,3245
ENO jerárquico	1	27,8125	33,6884	0,9970	0,0285	101	0,0384
	2	159,330	26,1077	6,9801	0,1955	214	0,0973
	3	400,5415	22,1043	0,9531	0,3151	245	0,1856
	4	962,0801	18,2987	0,8732	1,1633	250	0,3242
WENO	1	83,3407	28,6677	1,0072	-0,4396	180	0,0677
	2	206,8424	27,9744	0,9837	0,2481	214	0,0998
	3	618,6745	20,2162	0,9444	0,7119	250	0,1934
	4	1345	16,8409	0,8633	1,5738	250	0,354
ENO-SR	1	27,9109	33,6731	,09971	0,0255	101	0,0386
	2	159,9576	26,0908	0,9799	0,2067	214	0,0978
	3	405,7442	22,0483	0,9519	0,3626	245	0,1870
	4	1006,2	18,1040	0,8666	1,2478	250	0,3304
RACIONAL	1	88,3704	28,6677	1,0072	-0,4396	180	0,0677
	2	206,8428	24,9744	0,9837	0,2481	214	0,0998
	3	618,6745	20,2162	0,9444	0,7119	250	0,1934
	4	1345,8	16,8409	0,8633	1,57,38	250	0,3254
PPH	1	30,0271	33,3557	0,9971	-0,0182	85	0,0390
	2	169,2704	25,84,50	,09789	,1069	214	0,0988
	3	419,3260	21,9053	0,9509	0,0496	245	0,1877
	4	1001,2	18,1255	0,8647	0,8665	250	0,3296

Tabla 4.8: Resultados numéricos para la imagen *Pet-Tc2.jpg* mediante el uso de las diferentes técnicas con diferentes factores de *zoom*.

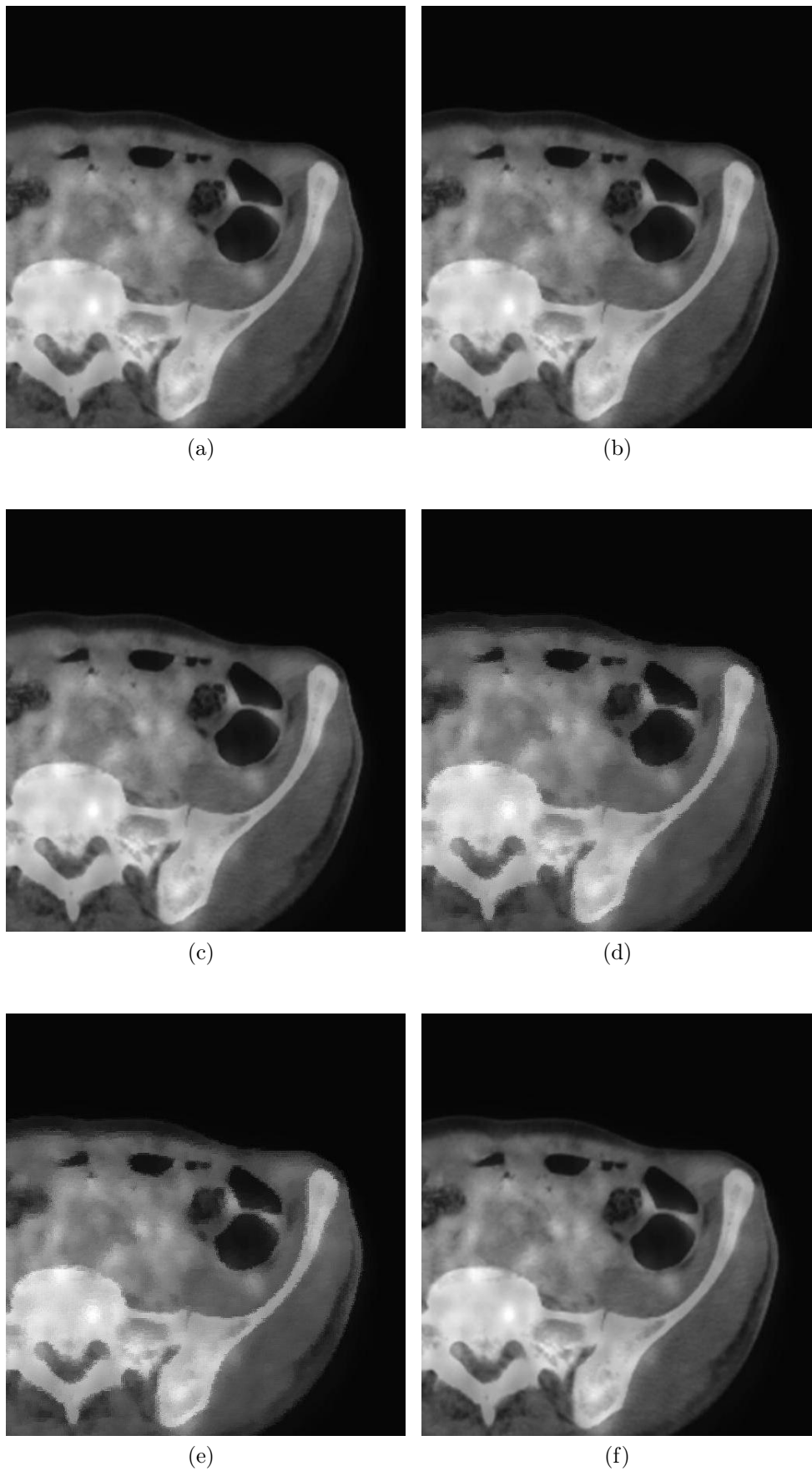


Figura 4.17: Imágenes Pet-Ct2.jpg con un nivel de zoom (x2) acotadas por una ventana de 516×542 píxeles. Reconstrucción con ENO no jerárquico (a), con ENO jerárquico (b), con ENO-SR (c), con WENO (d), con Racional (e), y con la técnica PPH (f).

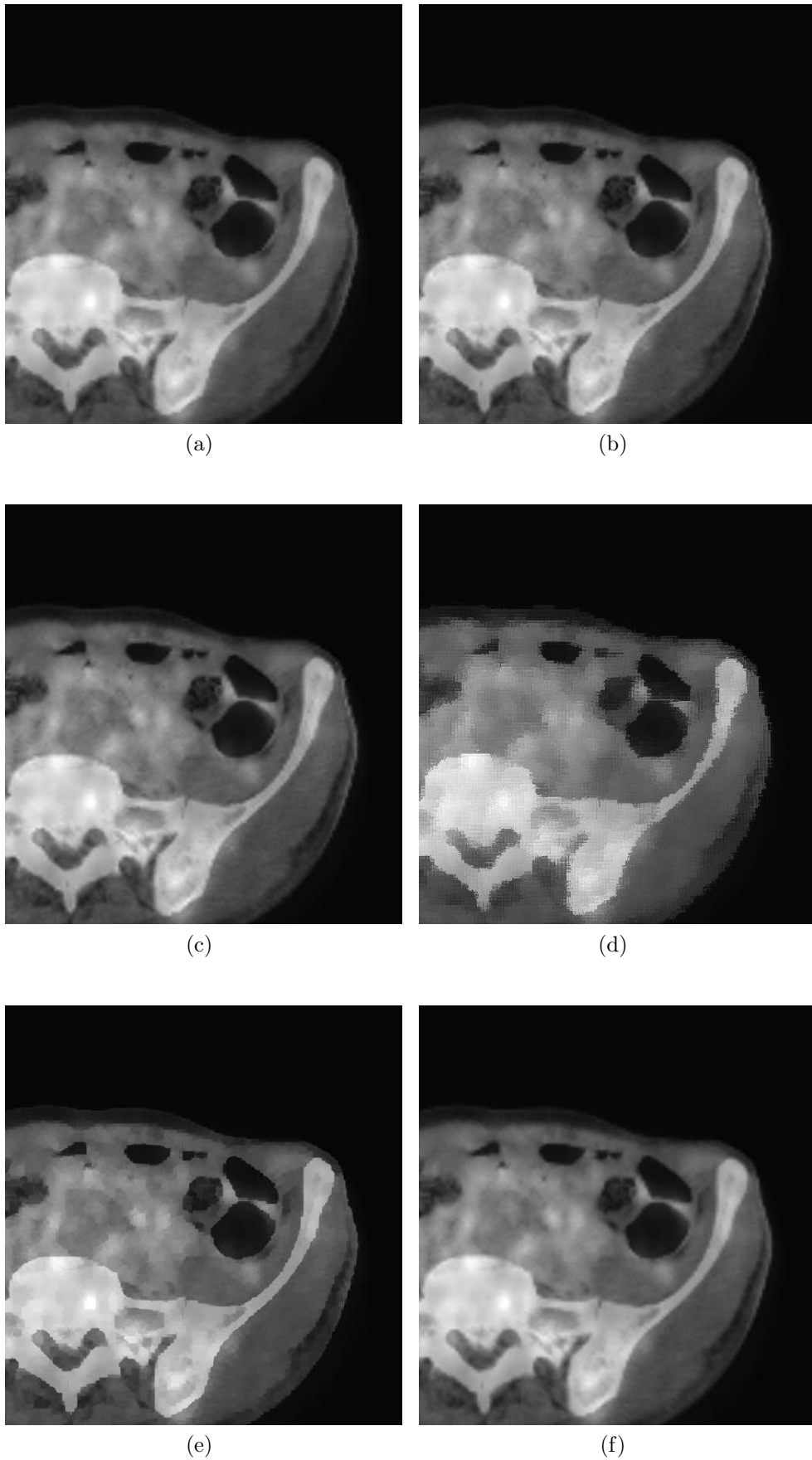


Figura 4.18: Imágenes *Pet-Ct2.jpg* con un factor de zoom 2 (x4). Reconstrucción con ENO no jerárquico (a), con ENO jerárquico (b), con ENO-SR (c), con WENO (d), con Racional (e), y con la técnica PPH (f).

4.3.9. Resultados con la imagen *Pet1*

Técnica	N	MSE	PSNR	NCC	AD	MD	NAE
ENO no jerárquico	1	251,8877	24,1187	0,9306	0,0585	255	0,2444
	2	740,6606	19,4346	0,7822	0,4978	255	0,5137
	3	120,4	17,3305	0,5941	0,5341	255	1,5484
	4	2072,6	14,9636	0,3961	0,0265	255	0,1320
ENO jerárquico	1	239,6264	24,3355	0,9318	0,1152	255	0,2460
	2	734,4960	19,7409	0,7874	0,4650	255	0,5162
	3	1223,9	17,2533	0,5916	1,5453	255	0,7568
	4	2124,7	14,8578	0,4020	-0,4020	255	1,1416
WENO	1	451,3236	21,5859	0,8571	1,3639	255	0,3543
	2	937,8203	18,4096	0,6013	4,4571	255	0,5726
	3	1390,6	15,2723	0,3690	8,5724	255	1,0293
	4	2100	05,2723	0,2100	8,1692	255	1,0293
ENO-SR	1	240,1396	24,3262	0,9313	0,416	255	0,2469
	2	735,6175	19,4643	0,7859	0,5404	255	0,7669
	3	1225,6	17,1422	0,5877	1,5142	255	0,7669
	4	2106,7	24,8149	0,3991	0,3621	255	1,1311
RACIONAL	1	248,0244	23,5972	0,9220	0,6031	255	0,2482
	2	769,0261	19,2714	0,7611	2,1585	255	0,4986
	3	1236,7	17,2082	0,4864	6,1446	255	06910
	4	1924,4	15,2879	0,2692	7,7486	255	0,9739
PPH	1	237,5153	24,3262	0,9313	0,1416	255	0,2469
	2	735,6175	19,4643	0,7822	0,1978	255	0,2137
	3	1255,6	17,1422	0,5941	1,5484	255	0,7472
	4	2106,7	14,8949	0,3991	0,3621	255	1,1311

Tabla 4.9: Resultados numéricos para la imagen *Pet1.jpg* mediante el uso de las diferentes técnicas con diferentes factores de *zoom*.

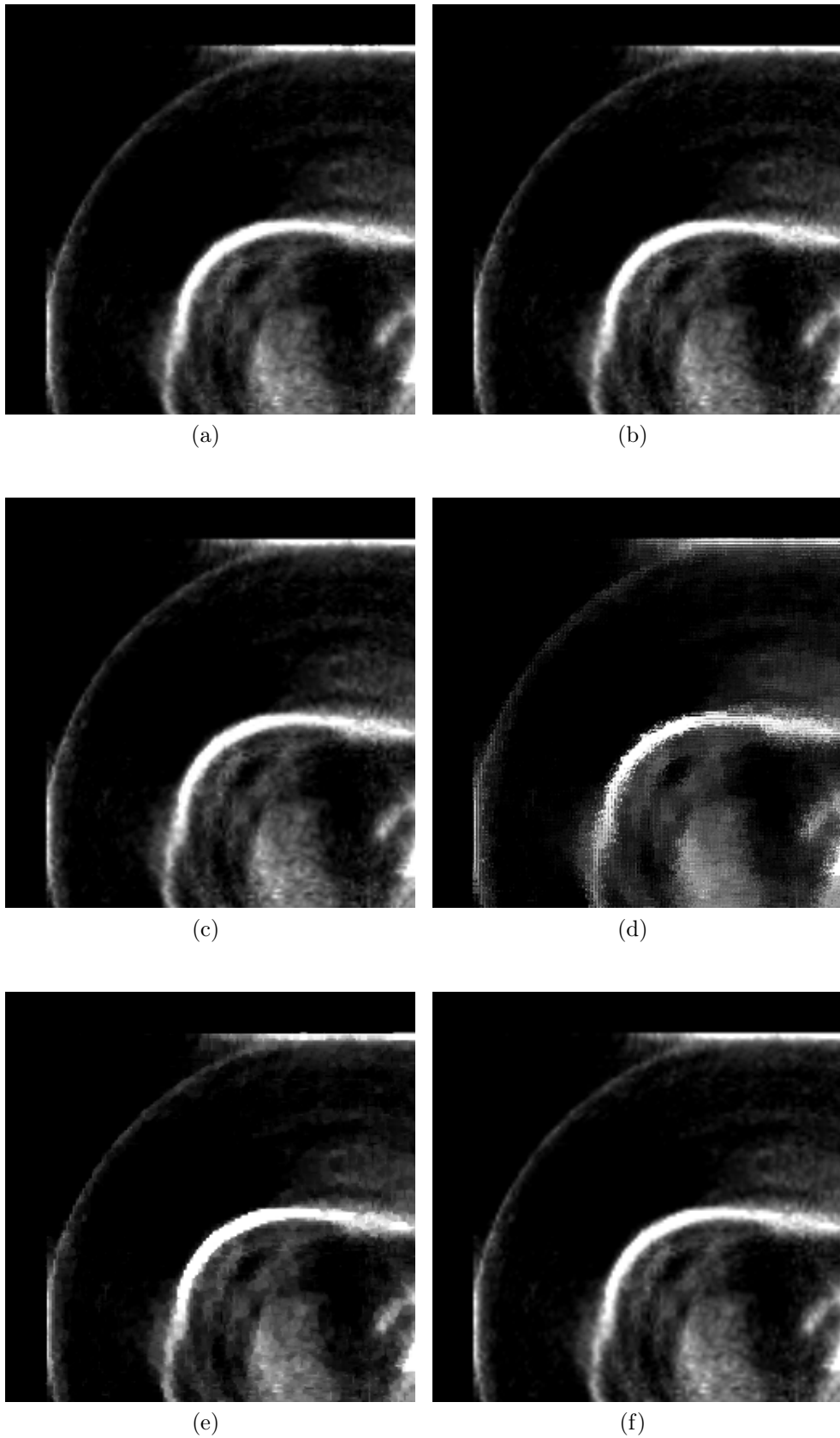


Figura 4.19: Imágenes *Pet1.jpg* con un nivel de zoom ($\times 2$) acotadas por una ventana de 255×255 píxeles. Reconstrucción con ENO no jerárquico (*a*), con ENO jerárquico (*b*), con ENO-SR (*c*), con WENO (*d*), con Racional (*e*), y con la técnica PPH (*f*).

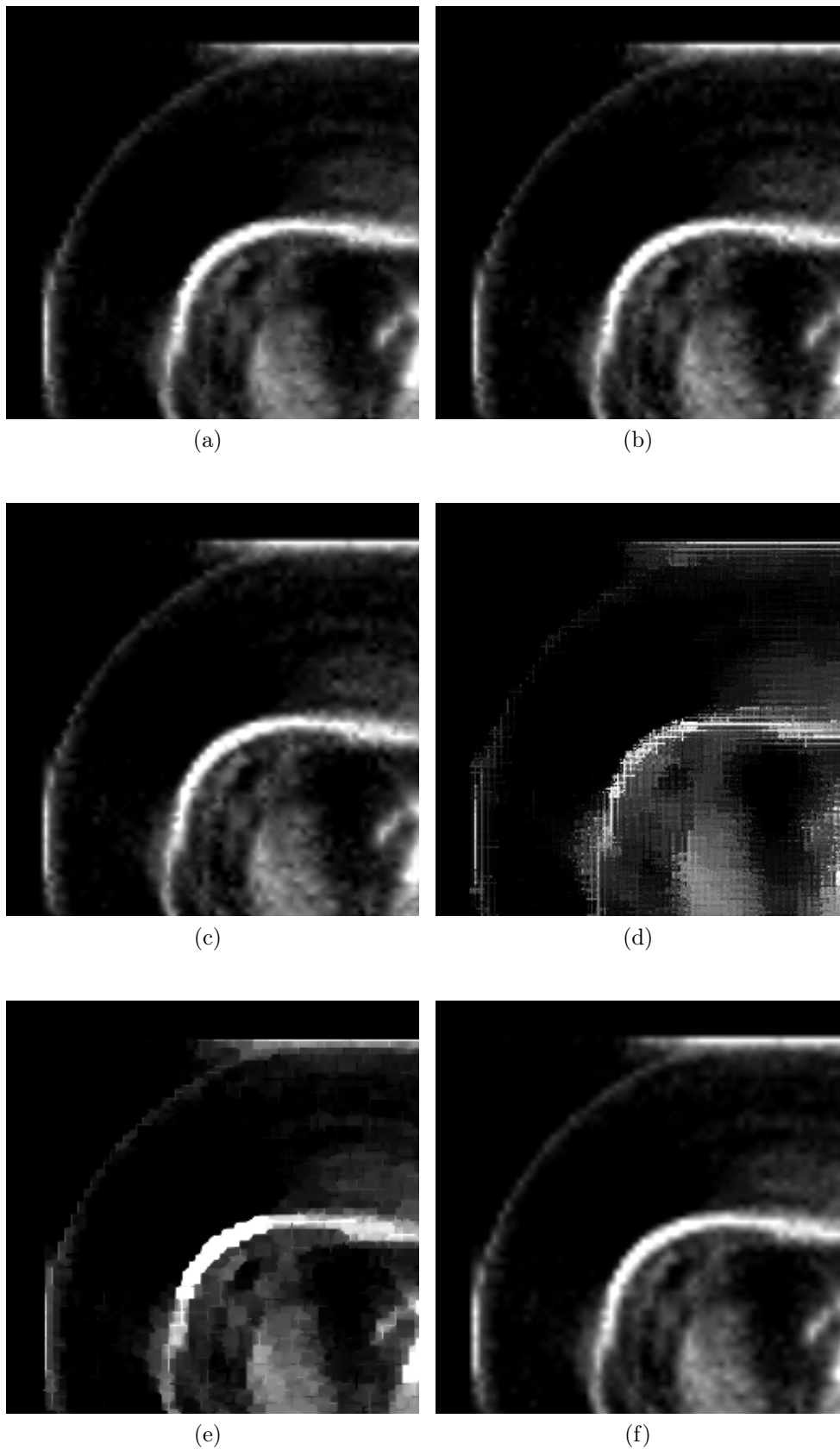


Figura 4.20: Imágenes `Pet1.jpg` con un factor de zoom 2 (x4) acotadas por una ventana de 255×255 píxeles. Reconstrucción con ENO no jerárquico (a), con ENO jerárquico (b), con ENO-SR (c), con WENO (d), con Racional (e), y con la técnica PPH (f).

4.3.10. Resultados con la imagen *Pet2*

Técnica	N	MSE	PSNR	NCC	AD	MD	NAE
ENO no jerárquico	1	178,0352	25,6257	0,9962	0,0471	255	0,0132
	2	441,1912	21,6845	0,9905	1,4865	255	0,0339
	3	1159,9	17,4864	0,9746	3,9229	255	0,1215
	4	2718	13,7884	0,9408	9,3452	255	0,1215
ENO jerárquico	1	177,7876	25,6318	0,9962	0,6483	255	0,0133
	2	439,1984	21,7042	0,9903	1,4999	255	0,0341
	3	1153,2	17,5117	0,9747	3,9870	255	0,0665
	4	2540,9	13,7519	0,9411	9,3063	255	0,1230
WENO	1	248,6434	24,1750	1,0096	-2,571	255	0,0277
	2	590,2008	20,4208	1,090	-5,2726	255	0,0520
	3	1219,0	17,2709	1,0282	-8,6821	255	0,0837
	4	2191,1	14,8692	1,0327	-11,7271	255	0,1248
ENO-SR	1	177,8455	25,6304	0,9962	0,6459	255	0,0133
	2	459,7052	21,6991	0,9904	1,5135	255	0,0337
	3	1173,2	17,4370	0,9737	4,1459	255	0,0666
	4	2765,6	13,7129	0,9389	9,7851	255	0,1227
RACIONAL	1	182,5956	25,5159	0,9966	0,5809	255	0,0139
	2	448,7212	21,6110	0,9929	0,5809	25	0,0139
	3	1168,0	17,2709	0,9806	1,0145	255	0,0337
	4	2553,9	13,8920	0,9527	6,4235	255	0,1169
PPH	1	177,9268	25,6284	0,9965	0,5957	255	0,0133
	2	439,8465	21,6991	0,9904	1,5135	255	0,0337
	3	1173,2	17,4370	0,9737	4,1459	255	0,0666
	4	2765,6	13,7129	0,9389	9,7851	255	0,1227

Tabla 4.10: Resultados numéricos para la imagen *Pet2.jpg* mediante el uso de las diferentes técnicas con diferentes factores de *zoom*.

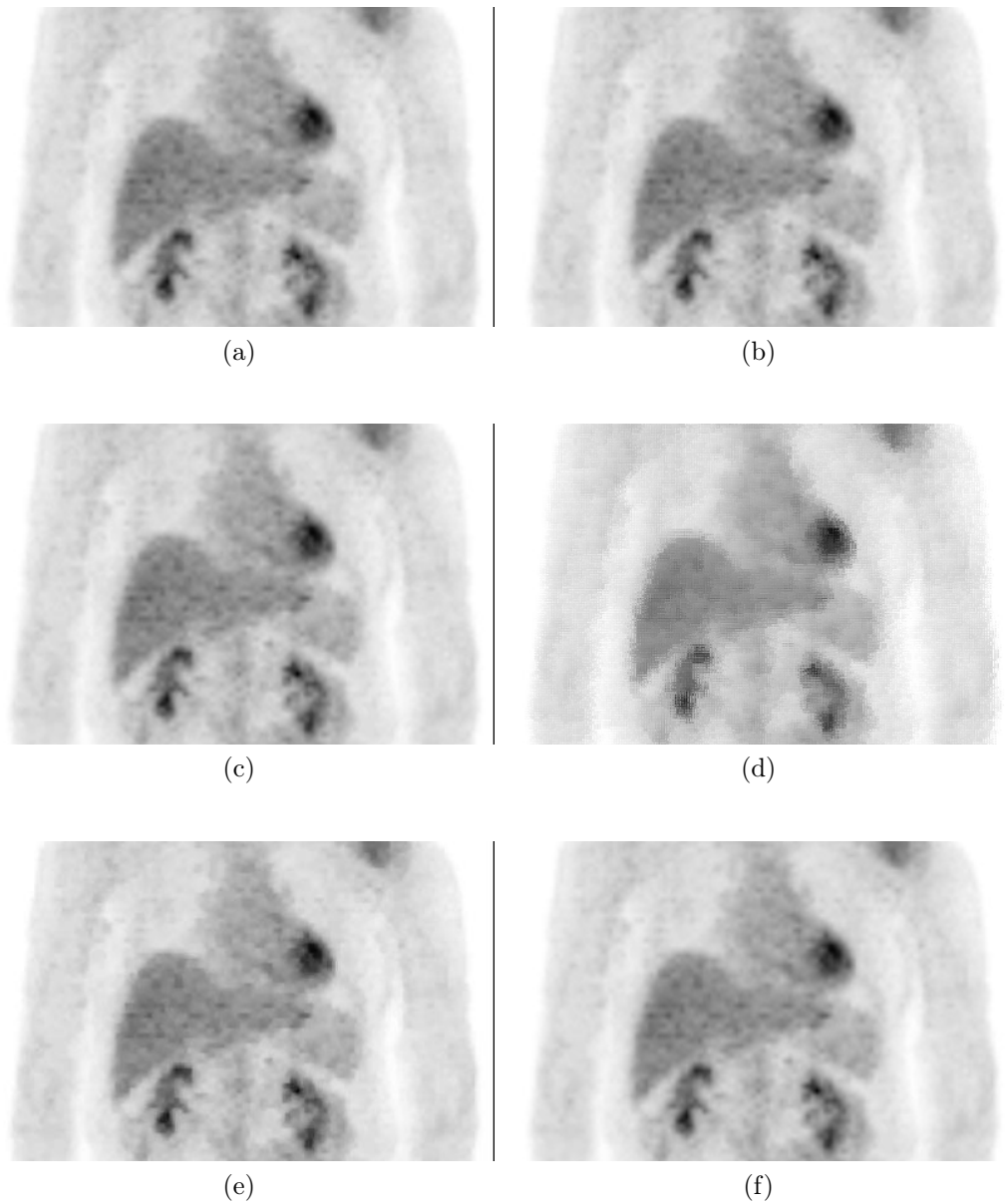


Figura 4.21: Imágenes *Pet2.jpg* con un nivel de zoom ($\times 2$) acotadas por una ventana de 400×256 píxeles. Reconstrucción con ENO no jerárquico (*a*), con ENO jerárquico (*b*), con ENO-SR (*c*), con WENO (*d*), con Racional (*e*), y con la técnica PPH (*f*).

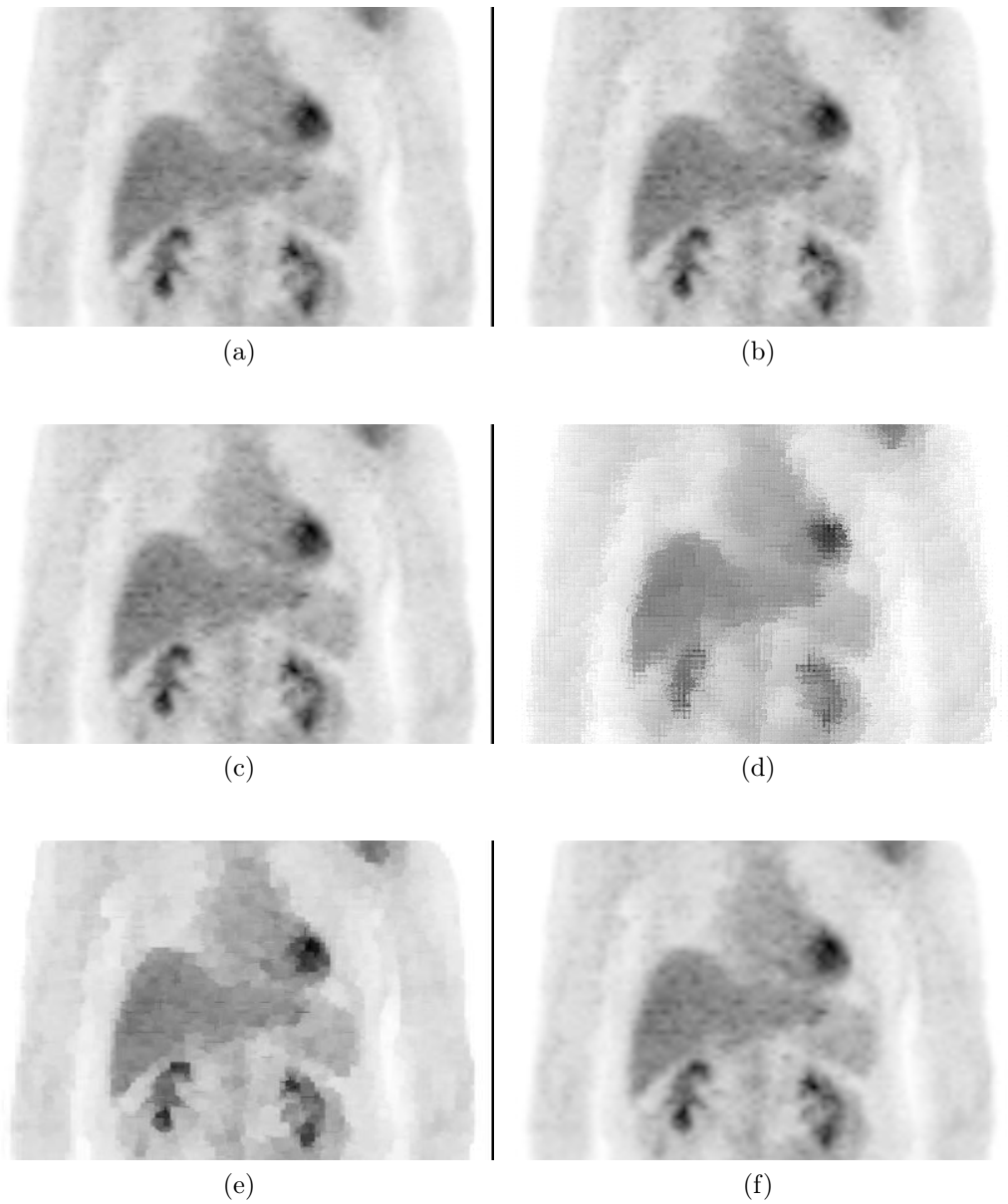


Figura 4.22: Imágenes *Pet2.jpg* con un factor de zoom 2 (x4) acotadas por una ventana de 400×256 píxeles. Reconstrucción con ENO no jerárquico (*a*), con ENO jerárquico (*b*), con ENO-SR (*c*), con WENO (*d*), con Racional (*e*), y con la técnica PPH (*f*).

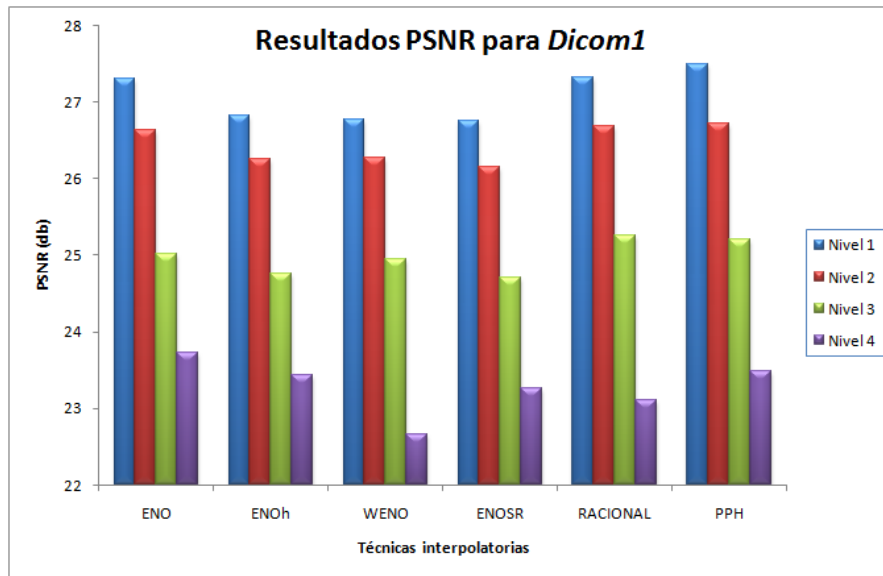


Figura 4.23: Gráfico que muestra los valores del PSNR obtenidos para cada técnica y para cada nivel, aplicados sobre la imagen Dicom1.

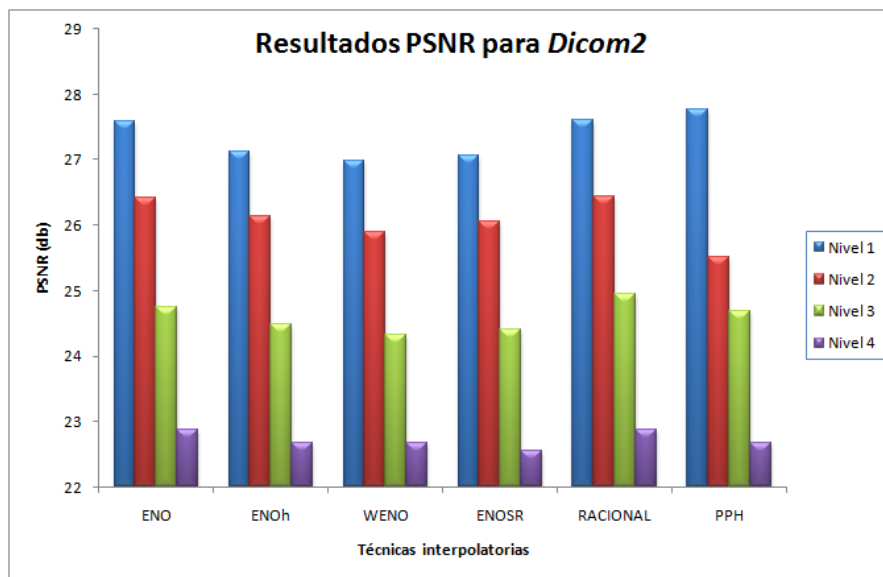


Figura 4.24: Gráfico que muestra los valores del PSNR obtenidos para cada técnica y para cada nivel, aplicados sobre la imagen Dicom2.

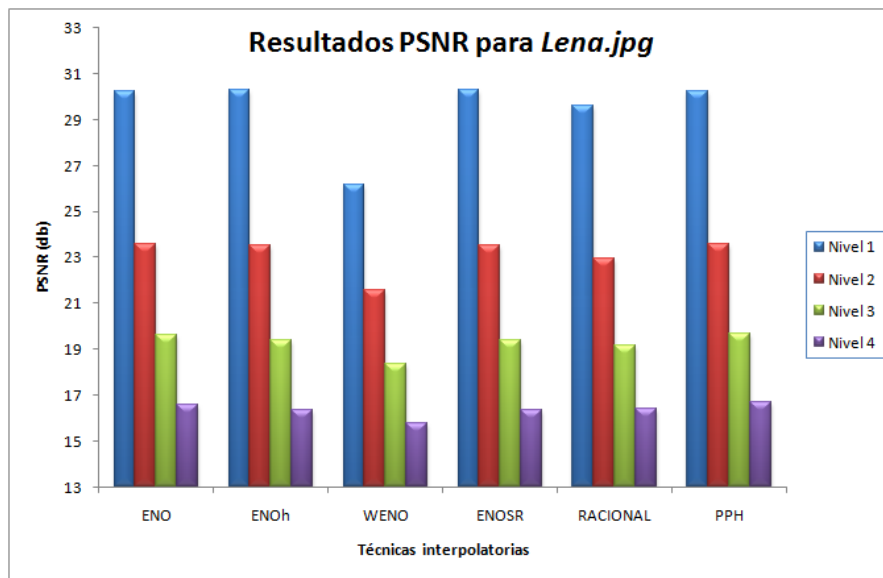


Figura 4.25: Gráfico que muestra los valores del PSNR obtenidos para cada técnica y para cada nivel, aplicados sobre la imagen *Lena.jpg*.

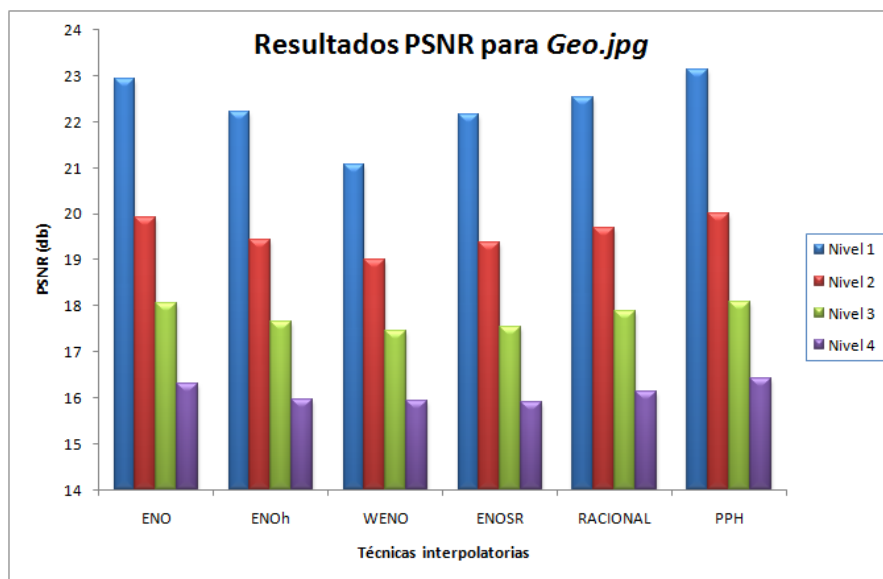


Figura 4.26: Gráfico que muestra los valores del PSNR obtenidos para cada técnica y para cada nivel, aplicados sobre la imagen *Geo.jpg*.

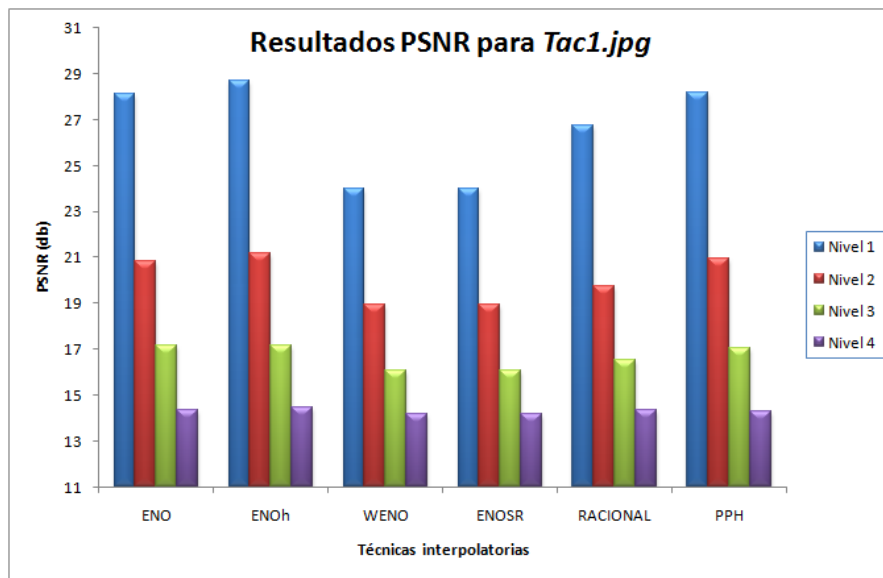


Figura 4.27: Gráfico que muestra los valores del PSNR obtenidos para cada técnica y para cada nivel, aplicados sobre la imagen Tac1.jpg.

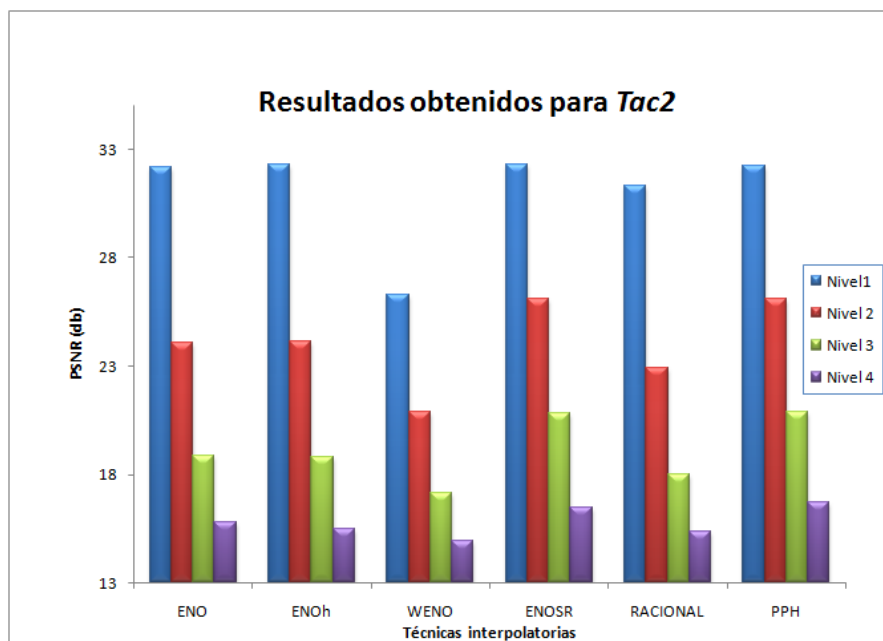


Figura 4.28: Gráfico que muestra los valores del PSNR obtenidos para cada técnica y para cada nivel, aplicados sobre la imagen Tac2.jpg.

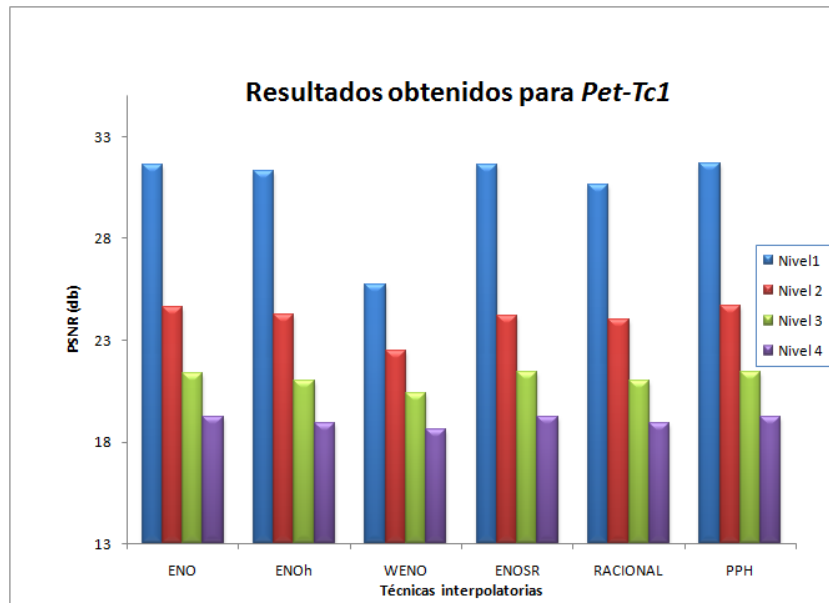


Figura 4.29: Gráfico que muestra los valores del PSNR obtenidos para cada técnica y para cada nivel, aplicados sobre la imagen *Pet-Tc1.jpg*.

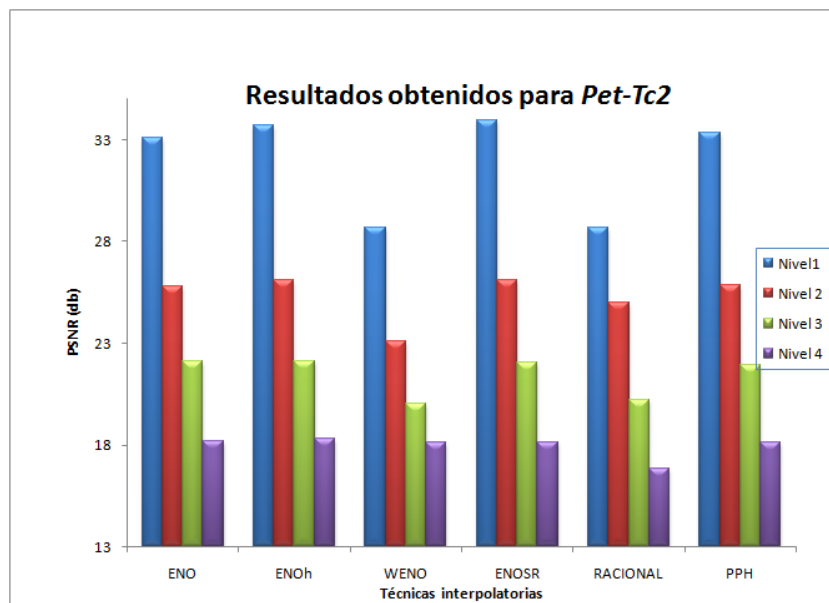


Figura 4.30: Gráfico que muestra los valores del PSNR obtenidos para cada técnica y para cada nivel, aplicados sobre la imagen *Pet-Tc2.jpg*.

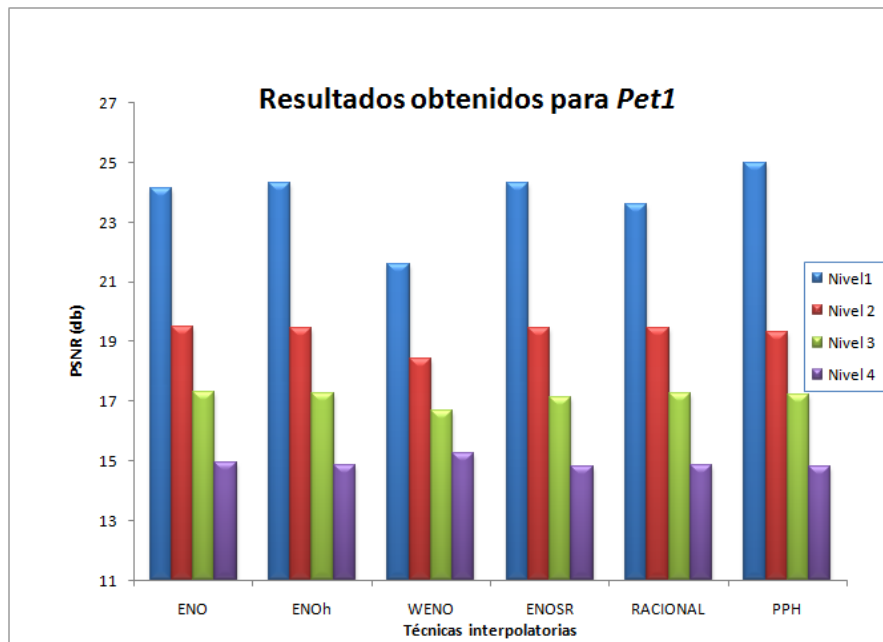


Figura 4.31: Gráfico que muestra los valores del PSNR obtenidos para cada técnica y para cada nivel, aplicados sobre la imagen *Pet1.jpg*.

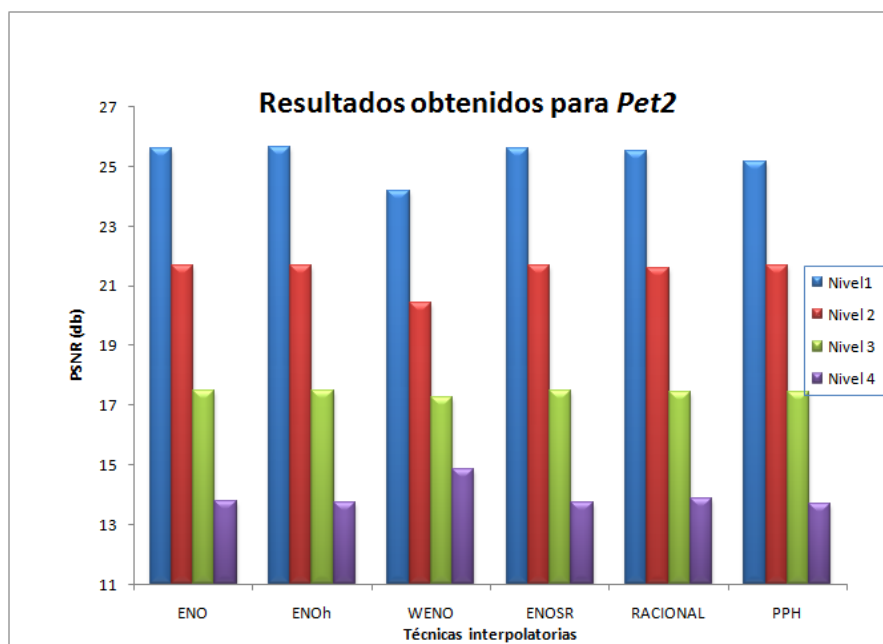


Figura 4.32: Gráfico que muestra los valores del PSNR obtenidos para cada técnica y para cada nivel, aplicados sobre la imagen *Pet2.jpg*.

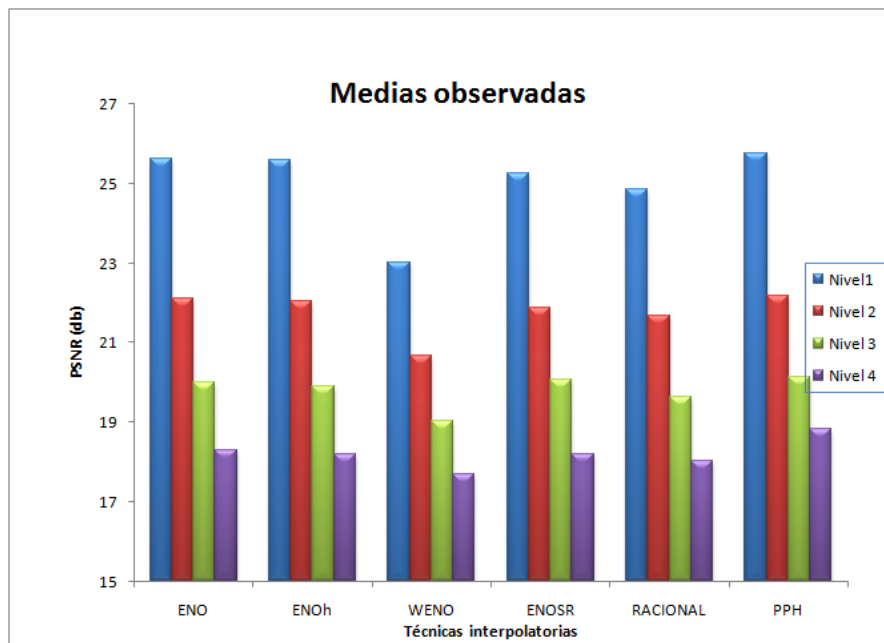


Figura 4.33: Gráficos de medias obtenidas para cada técnica en cada nivel.

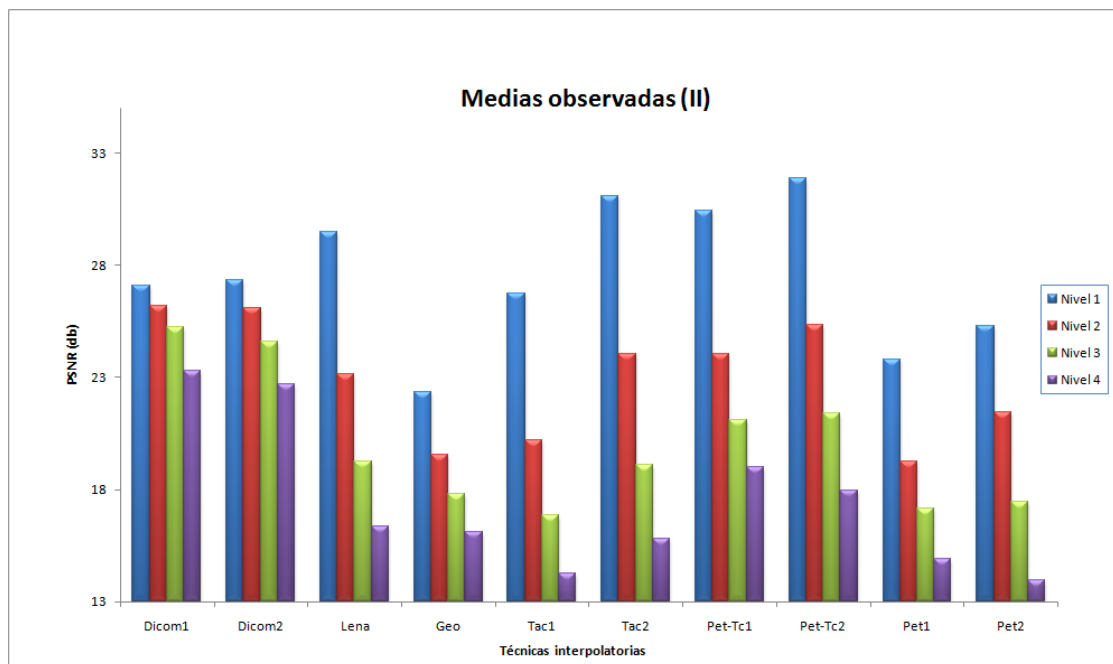


Figura 4.34: Gráficos de medias obtenidas para cada imagen en cada nivel.

	Niv.	ENO	ENOh	WENO	ENOSR	RAC	PPH
Dicom1	1	27,3141	26,8310	26,7732	26,7642	27,3316	27,4976
	2	26,6334	26,2647	26,2823	26,1571	26,6959	26,7284
	3	25,0186	24,7653	24,9499	24,7079	25,2565	25,2007
	4	23,7222	23,4349	22,6624	23,2569	23,1004	23,4844
Dicom2	1	27,5926	27,1265	26,9749	27,0586	27,6013	27,7758
	2	26,4095	26,1340	25,9002	26,0536	26,7407	25,5176
	3	24,7515	24,4824	24,3262	24,4121	24,9581	24,6936
	4	22,8718	22,6801	22,6634	22,5511	22,8736	22,6634
Lena	1	30,2254	30,2846	26,1565	30,2668	29,6117	30,2566
	2	23,6013	23,5511	21,5464	23,5191	22,9411	23,6013
	3	19,6035	19,4124	18,3308	19,4130	19,1619	19,6525
	4	16,6035	16,3656	15,7553	16,3305	16,3871	16,6780
Geo	1	22,9247	22,2297	21,0658	22,1488	22,5190	23,1453
	2	19,9252	19,4448	19,0043	19,3654	19,7008	20,0000
	3	18,0696	17,6552	17,4473	17,5499	17,8960	18,0955
	4	16,3071	15,9710	15,9357	15,8952	16,1294	16,4225
Tac1	1	28,125	28,6793	23,972	24,852	26,7670	28,1518
	2	20,8502	21,1671	18,9375	19,5344	19,7666	20,9321
	3	17,1625	17,1519	16,099	17,006	16,5243	17,0532
	4	14,3698	14,4804	14,1893	14,0192	14,3547	14,2668
Tac2	1	32,1843	32,2915	26,2553	32,2758	31,3324	32,2454
	2	24,0532	24,1185	20,8988	26,1122	22,9274	26,0852
	3	18,8625	18,824	17,1587	20,8114	17,9754	20,8955
	4	15,7816	15,4973	14,9356	16,4653	15,3568	16,7164
Pet-Tc1	1	31,6050	31,3221	25,7135	31,6382	30,6117	31,6841
	2	24,6263	24,2503	22,4640	24,2019	24,0153	24,6832
	3	21,3709	20,9872	20,4175	21,4209	21,0164	21,4209
	4	19,2030	18,9008	18,6165	19,2494	18,9336	19,2494
Pet-Tc2	1	33,0633	33,6884	28,6677	33,9731	28,6677	33,3557
	2	25,8148	26,1077	23,1062	26,0908	24,9744	25,845
	3	22,1315	22,1043	20,0143	22,0483	20,2162	21,9053
	4	18,2016	18,2987	18,1040	18,1040	16,8409	18,1255
Pet1	1	24,1187	24,3355	21,5859	24,3262	23,5972	24,9739
	2	19,4946	19,4709	18,4096	19,4643	19,4709	19,3097
	3	17,3303	17,2533	16,6807	17,1422	17,2533	17,244
	4	14,9656	14,8575	15,2723	14,8149	14,8578	14,8117
Pet2	1	25,6257	25,6318	24,175	25,6304	25,5191	25,1684
	2	21,6845	21,7042	20,4208	21,697	21,6110	21,6991
	3	17,4864	17,5117	17,2709	17,4925	17,4562	17,437
	4	13,7884	13,7519	14,8692	13,7267	13,8920	13,7129

Tabla 4.11: Tabla que resume los valores de PSNR obtenido para las distintas imágenes agrupados por tipo de técnica.

Tabla de medias y desviaciones por técnica							
Nivel	\bar{x}_i, σ_i^2	ENO	ENO _h	WENO	ENOSR	RACIONAL	PPH
1	\bar{x}_1	25,7361	25,6925	23,2079	25,4120	25,0108	25,8169
	σ_1^2	5,9184	6,0309	4,5562	5,8133	5,4680	5,8018
2	\bar{x}_2	22,3857	22,3043	21,0075	22,1795	22,0134	22,4327
	σ_2^2	3,8146	3,8441	3,4851	3,7390	3,8045	3,7456
3	\bar{x}_3	20,3125	20,1801	19,4280	20,4305	19,5424	20,4060
	σ_3^2	3,4152	3,4402	3,2180	3,5103	3,6781	3,3880
4	\bar{x}_4	18,5086	18,3912	18,0670	18,4025	18,6184	18,7667
	σ_4^2	18,5086	18,3912	18,0670	18,4025	18,6184	18,7667

Tabla 4.12: Media y desviación estándar de los valores del PSNR de todas las imágenes obtenidos para cada técnica y nivel de *zoom*.

Tabla de medias y desviaciones por imagen (I)							
Nivel	\bar{x}_i, σ_i^2	Dicom1	Dicom2	Lena	Geo	Tac1	Tac2
1	\bar{x}_1	27,0853	27,3550	29,4669	22,3389	26,7579	31,0975
	σ_1^2	0,3311	0,3402	1,6424	0,7341	1,9441	2,4006
2	\bar{x}_2	26,2188	26,0759	23,1267	19,5734	20,1980	24,0326
	σ_2^2	0,7679	0,3437	0,8144	0,3757	0,9076	1,9793
3	\bar{x}_3	25,2247	24,6040	19,2624	17,7856	16,8329	19,0879
	σ_3^2	0,4899	0,2383	0,4882	0,2742	0,4296	1,5045
4	\bar{x}_4	23,2769	22,7172	16,3533	16,1102	14,2800	15,7922
	σ_4^2	0,3676	0,1289	0,3248	0,2157	0,1614	0,6808

Tabla de medias y desviaciones por imagen (II)					
Nivel	\bar{x}_i, σ_i^2	Pet-Tc1	Pet-Tc2	Pet1	Pet2
1	\bar{x}_1	30,4291	31,9027	23,8229	25,2917
	σ_1^2	2,3447	2,5244	1,1818	0,5755
2	\bar{x}_2	24,0402	25,3232	19,2700	21,4694
	σ_2^2	0,8139	1,1624	0,4268	0,5149
3	\bar{x}_3	21,1056	21,4033	17,1506	17,4425
	σ_3^2	0,3911	1,0028	0,2379	0,0882
4	\bar{x}_4	19,0255	17,9458	14,9300	13,9569
	σ_4^2	0,2542	0,5465	0,1767	0,4515

Tabla 4.13: Media y desviación estándar de los valores del PSNR obtenidos para todas las técnicas agrupadas por tipo de imagen y nivel de *zoom*.

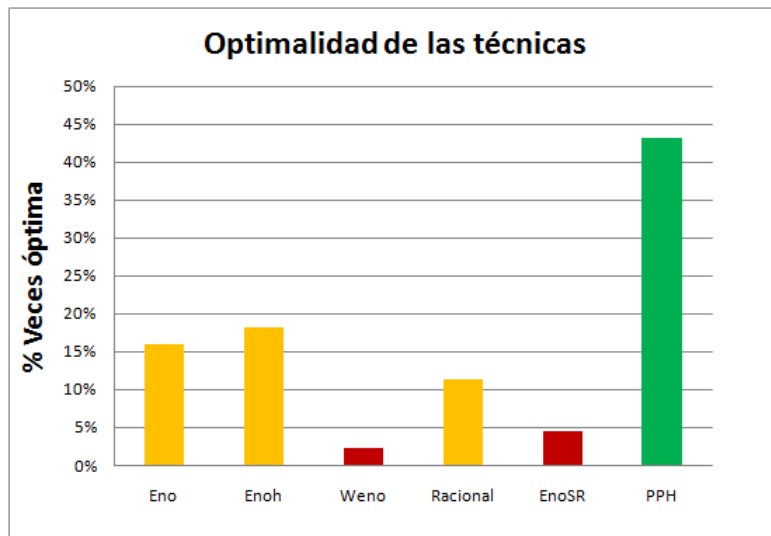


Figura 4.35: Histograma que resume el porcentaje de veces que una técnica consigue mejores resultados que el resto para una configuración ($\langle \text{Imagen} \rangle \langle \text{Técnica} \rangle \langle \text{Nivel} \rangle$). Cada barra aparece de un color diferente y atiende a la división en grupos establecida en base a la calidad obtenida. Aparecen en verde la técnica incluida en el grupo de *alta calidad*, en ámbar las clasificadas en el grupo de *calidad media* y en aquéllas del denominado *grupo de baja calidad*.

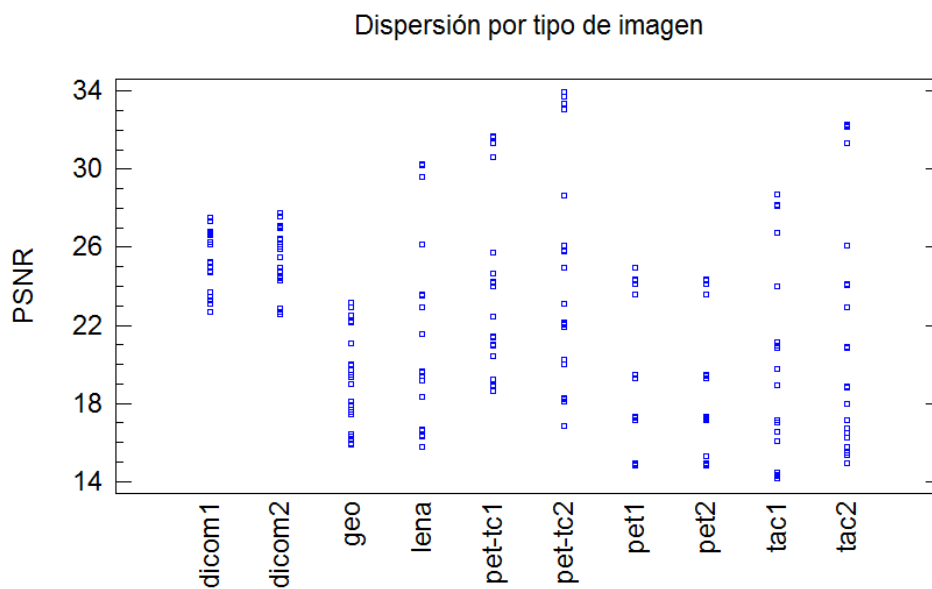


Figura 4.36: Gráfico de dispersión obtenido en función del PSNR obtenidos para cada tipo de imagen.

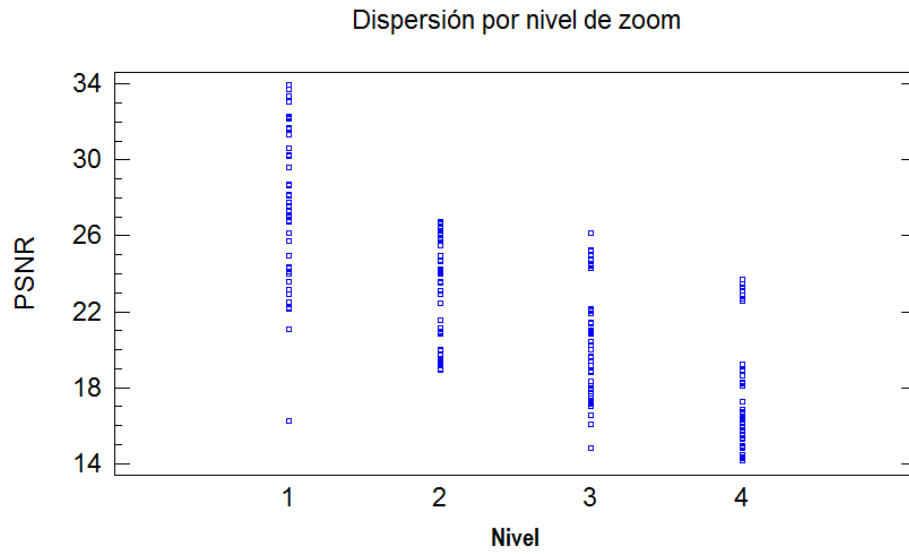


Figura 4.37: Gráfico de dispersión obtenido en función del PSNR obtenidos para cada nivel de *zoom*.

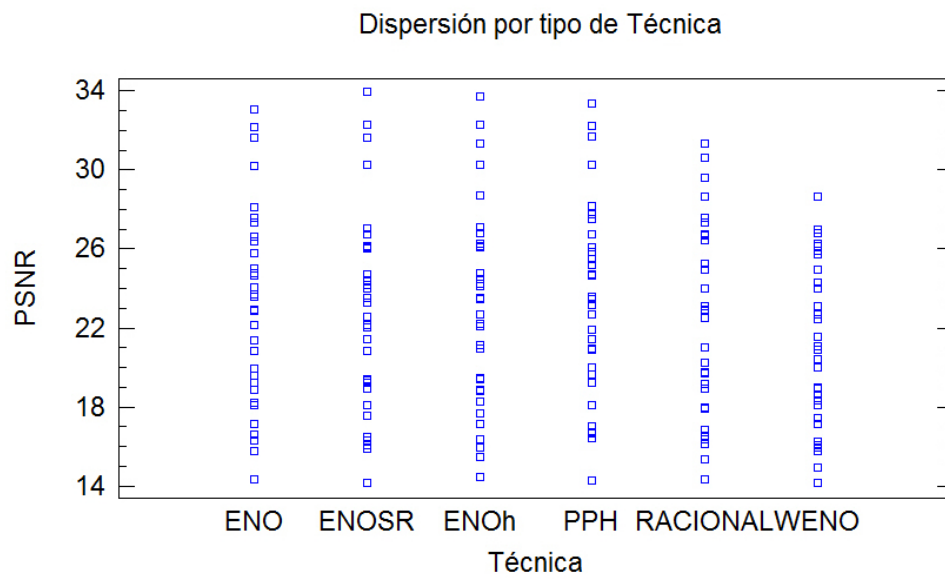


Figura 4.38: Gráfico de dispersión obtenido en función del PSNR obtenidos para cada técnica interpolatoria estudiada.

4.3.11. Análisis de los resultados obtenidos

En esta sección se propone emitir una valoración acerca de los resultados obtenidos en la sección anterior. El análisis parte de las tablas de resultados de PSNR obtenidas por cada técnica y para cada imagen (tablas 4.1, 4.2, 4.3, 4.4, 4.5, 4.6, 4.7, 4.8, 4.9 y la 4.10)), obtenidas en los anteriores experimentos.

En todas ellas han sido medidos los indicadores anteriormente comentados en función de la técnica y nivel utilizado. De ahora en adelante denominaremos *configuración* a la terna $\langle \text{Imagen-Técnica-Nivel} \rangle$, para referirnos a la *técnica* empleada para ampliar una *imagen* con un *nivel* determinado de *zoom*.

En las tablas de datos de cada imagen, se verifica el incremento del valor del PSNR, del Error Cuadrático Medio y el Error Absoluto Normalizado conforme aumenta el nivel de ampliación, aunque este aumento no es proporcional en todos los indicadores.

Con el objetivo de poder observar la evolución del valor de PSNR obtenido en función de *configuraciones* diferentes, se han construido los gráficos 4.23, 4.24, 4.25, 4.26, 4.27, 4.28, , 4.30, 4.31 y 4.32. Todos ellos tratan de resumir la misma información obtenida a partir de imágenes de distinta naturaleza, y presentan en el eje de abscisas las diversas técnicas de reconstrucción empleadas, y en el de ordenadas un rango de PSNR dado. Para cada valor en el eje de abscisas, son mostradas cuatro barras representando los cuatro niveles de ampliación con los que se ha llevado a cabo el experimento.

En primer lugar se muestra el gráfico correspondiente a la imagen *Dicom1*. Las reconstrucciones que mejores resultados obtienen para esta imagen de gammagrafía son, respectivamente la técnica PPH y la técnica ENO no jerárquico. A pesar de esto, los resultados entre distintas técnicas no son notables, ya que las técnicas disminuyen a razón de aproximadamente 1, 2 decibelios cuando se incrementa nivel de *zoom* empleado. Destacar la proximidad en decibelios que existe entre el nivel 1 de resolución con alguna técnica y el nivel 2 correspondiente a la misma técnica.

En segundo lugar el gráfico 4.24 tampoco muestra diferencias significativas entre los resultados de las diferentes técnicas aplicadas sobre la imagen *Dicom2*. Para el primer nivel de ampliación (las diferencias se encuentran por debajo de un decibelio), los óptimos para ésta imagen se consiguen mediante la técnica PPH para el primer nivel y la técnica Racional para los restantes.

En general, con esta imagen, la alternancia entre las distintas técnicas no produce un valor del PSNR medido sustancialmente diferente aunque visualmente unas imágenes parecen ser mucho mejor que otras. Esto puede ser producido por la propia naturaleza de la imagen: por un lado, influirá la resolución inicial de la imagen (debido al procedimiento inicial de reducción de la imagen original); y por otro, los contornos que describen la figura humana en la imagen original no son completamente nítidos, y la frontera entre la imagen humana y el resto no está

completamente delimitada, lo que provoca que la aparición de regiones «difusas» conforme el observador se aleja del centro de la imagen hacia los extremos. Estas podrían ser dos causas de la obtención de imágenes visualmente no aceptables, aún obteniendo valores de PSNR que a priori sí lo parezcan.

Los resultados asociados a la imagen *Lena* pueden ser consultados gráficamente en 4.25. En este caso, mientras que las técnicas las cuales consideramos que han obtenido resultados satisfactorios obtienen alrededor de 30 db para el primer nivel de *zoom* (que respectivamente son ENO jerárquico, ENO no jerárquico y la técnica PPH), aquéllas que les siguen en cuanto a calidad se refiere, es decir (Racional, WENO y ENOSR) obtienen, en el mayor de los casos, alrededor de cuatro decibelios menos para este nivel. Conforme aumenta el nivel de ampliación, estas diferencias entre técnicas van disminuyendo siendo de más o menos dos decibelios, como en las imágenes de gammagrafías anteriormente comentadas. Ésta diferencia considerable que se percibe en niveles bajos de ampliación, puede ser observada analizando las dos reconstrucciones con WENO para el primer y segundo nivel respectivamente, cuya calidad no parece aceptable (ver 4.7d y 4.8d).

En el gráfico 4.26 se presenta la información relativa a los resultados obtenidos en la reconstrucción de la imagen *Geo*. A primera vista se observan valores mucho menores que los obtenidos con la imagen *Lena*, que a priori podrían indicar reconstrucciones de menor calidad. Los mejores resultados son obtenidos por las técnicas PPH y WENO, rebasando la primera a la segunda en todos los niveles, aunque el margen de diferencia oscila alrededor de la décima parte un decibelio. Respecto las demás técnicas que siguen a éstas primeras en calidad, encontramos por un lado la Racional y por otro, la ENO jerárquico, la ENO-SR y la técnica WENO, en este orden respectivo de calidad. Los valores obtenidos para esta técnica y para esta imagen no son en absoluto concluyentes, pues todos los valores se distribuyen de la misma manera, aunque cabe destacar que la calidad que se aprecia visualmente de imágenes obtenidas con WENO (figuras 4.9d y 4.10d) es claramente, y por segunda vez, menor.

Sobre la imagen *Tac1*, cuyos datos son representados gráficamente en 4.27, cabe remarcar que la técnica ENO jerárquica es la que mejores relaciones *Señal a ruido de pico* obtiene tanto para el primer nivel, como para el segundo y el cuarto. Las técnicas que junto a ésta recoge mejores resultados es la PPH; aunque para el tercer nivel el mejor valor obtenido sea con la ENO no jerárquica. Le sigue la Racional por detrás con aproximadamente un salto de tres decibelios, y seguidamente, la reconstrucción WENO una vez más clasificada como aquélla que ha obtenido peores resultados. En el tercer y cuarto nivel, análogamente a lo que se viene observando, los valores de los distintos métodos no lineales convergen en torno a valores similares del indicador PSNR que se está considerando.

A continuación se halla el gráfico 4.28 que presenta los resultados extraídos de

la realización del experimento con *Tac2.jpg*. La reconstrucción ENO jerárquico alcanza el segundo mejor valor de PSNR de todas las pruebas realizadas. ENOSR, ENO no jerárquico y la reconstrucción PPH obtienen niveles aceptables de calidad para el primer nivel, pero a medida que aumentan los niveles de ampliación, la técnica PPH destaca en optimalidad por encima de los primeros. Dado que la diferencia en el primer nivel entre la técnica PPH y la ENOSR es despreciable, podemos afirmar que la primera técnica obtiene resultados óptimos para ésta imagen. Solo queda destacar el «escalón» existente en general, entre el primer y el segundo nivel de *zoom*, que oscila aproximadamente entre seis y ocho decibelios dependiendo de la técnica.

Seguidamente se encuentra la representación gráfica del indicador de calidad PSNR obtenido para la imagen *Pet-Tc1*. En general los resultados son bastante análogos a los obtenidos en la imagen anterior. Los valores óptimos para los cuatro niveles siguen siendo obtenidos por la técnica PPH, aunque se puede verificar que los errores en los que se incurriría si se selecciona cualquier otra técnica a excepción de la WENO, serían despreciables, ya que ambos obtienen resultados significativamente poco diferentes respecto al primero. Otra vez más, la reconstrucción WENO parece obtener peores resultados, aunque a partir del tercer nivel de ampliación la diferencia disminuye, y deja de reflejar distancias significativas respecto a otras técnicas.

Por su parte, para la imagen *Tac2*, las reconstrucciones que a priori se presentan como más aceptables son la interpolación ENOSR, y la ENO; tanto es su selección jerárquica como no jerárquica del *stencil*. Aunque la primera de ellas se presenta óptima para el nivel 1, y la segunda idónea para el tercer nivel de ampliación, la técnica ENO con particionamiento jerárquico muestra un valor más alto de calidad para los niveles segundo y cuarto y, dado que sus niveles para el primer y el tercer nivel no distan significativamente de los presentados por los algoritmos no lineales de interpolación ENOSR y ENO con selección no jerárquica del *stencil*, podemos concluir que la técnica que parece ser estable e idónea para esta imagen en concreto es la ENO con particionamiento jerárquico.

Prosiguiendo con el análisis, en las figuras 4.3.11 y 4.30 se muestran los gráficos para las imágenes *Pet-Tc1* y *Pet-Tc2*, respectivamente. En ellos se aprecia que la distribución del PSNR para ambas imágenes es similar, aún obteniendo dos técnicas interpolatorias como óptimas en cada caso: ENO no jerárquico para la primera imagen y ENO jerárquico para la segunda. Solo cabe destacar aquí, que el método WENO obtiene mejor valor del indicador para los niveles 4 de *zoom* de cada imagen; hecho que no ocurre en ninguna otra *configuración* estudiada. En 4.31 se muestra gráficamente el PSNR obtenido con la imagen *Pet1*. Con esta imagen, la técnica no lineal que obtiene mejores resultados es la ENO no jerárquico, si bien todas las demás obtienen resultados prácticamente igual de aceptables. Cabe

destacar que en esta ocasión y solo para el cuarto nivel de ampliación la técnica WENO obtiene el mejor resultado, aunque en todos los demás niveles se obtienen resultados que oscilan hasta un decibelio por debajo del valor obtenido por los restantes métodos.

En lo referente al último gráfico (figura 4.32 se repite el mismo patrón. La calidad para una reconstrucción dada no difiere significativamente de las restantes, exceptuando el caso WENO, que vuelve a ser la técnica con resultados más bajos. Aunque la técnica PPH aparezca también como óptima par el segundo nivel, la que mejores resultados obtiene (y en principio se propondría utilizar en ésta imagen) sería la ENO con selección jerárquica.

Para proporcionar una visión más compacta y nítida de los datos, se ha construido la tabla 4.11. Ésta tabla solo contiene el indicador PSNR a diferencia de sus anteriores, y permite descubrir a simple vista cuál ha sido la técnica que ha obtenido el valor óptimo en un mayor número de *configuraciones*. Ésta ha sido la *Piecewise Polynomial Harmonic* es la que ha obtenido una media mayor en el mayor número de casos, en contraposición a la WENO, que no ha conseguido superar los resultados calculados por otras técnicas.

La tabla 4.12 y desviaciones típicas para cada técnica, en función de los valores obtenidos para cada imagen en un nivel de *zoom* determinado. Los valores en negrita respresentan valores máximos de cada fila, es decir, los valores máximos de las medias de PSNR obtenido para todas las imágenes, con una técnica y un nivel dado. Ha esto le hemos llamada *valores óptimos*. Obviamente las desviaciones típicas de las técnicas enumeradas que han obtenido mejor calidad en la reconstrucción, son menores que las de las técnicas WENO, ENOSR y RACIONAL, y por consiguiente, las medias de las primeras son mucho mas respresentativas de los valores que se han obtenido, que no las medias obtenidas por éstas últimas técnicas, donde la desviación estándar llega a alcanzar valores alrededor de cuatro. En la misma línea se ha construido la tabla de medias obtenidas para cada nivel e imagen estudiada (figura 4.12) con el objetivo de mostrar si alguna imagen ha obtenido valores significativamente mayores de PSNR respecto a otra, hecho que propiciaría la elección de una u otra técnica en función de la imagen. Pero lo cierto es, que al margen de las imágenes DICOM (en las cuales se obtienen buenos valores del indicador, aunque no son apreciados visualmente) donde los valores están más concentrados, todas disminuyen con magnitud similar, y en los niveles altos de ampliación (el tercero y el cuarto) llegan a alcanzar valores que distan del primer nivel de *zoom* hasta diez decibelios. La diferencia entre los rangos que abarcan las reconstrucciones según la técnica considerada se observa muy bien en los gráficos de dispersión que se explican a continuación.

En la sección anterior aparecen tres diagramas de dispersión, construidos en función de la imagen empleada en la reconstrucción (4.36), en función del nivel de

reconstrucción empleado (4.37) y dependiendo de la técnica utilizada (4.38).

En primer lugar, la primera característica remarcable que se observa en el gráfico 4.36, es que las imágenes *Dicom1*, *Dicom2* y la imagen *Geo* han obtenido valores concentrados en un rango aproximado de [21, 27] para las dos primeras, y en el rango [15, 23] db para la imagen *Geo*. Éste aspecto contrasta con la representación de la dispersión de las imágenes, en las cuales se observa que las diferencias entre los extremos de un rango de valores obtenidos para una misma imagen alcanza valores de hasta aproximadamente diez decibelios. Aquí se hace visible el «escalón» entre el primer y el segundo nivel de resolución, donde los puntos son claramente atribuibles a uno u otro nivel. Conforme aumentan los niveles de ampliación, los valores tienden a converger.

En segundo lugar, atendiendo al gráfico de dispersión en función del nivel de *zoom* (4.37), se observa que aparentemente las amplitudes de los rangos de dispersión son similares para el tercer y cuarto nivel, mientras que el primer nivel abarca más de diez decibelios. Cabe destacar el salto que se produce en el rango de dispersión del nivel 4, donde los valores superiores a veinte decibelios corresponde exclusivamente a las imágenes de gammagrafía, y el resto, que se encuentra por debajo de los diecinueve decibelios; corresponde al valor del PSNR observado en el resto de las imágenes para un nivel cuatro de ampliación.

Por último, si se presta atención a la dispersión de las medias por técnica, (figura 4.38) se concluye que, a grandes rasgos, todas ellas, se distribuyen de manera similar a medida que varían los niveles de *zoom* y la imagen a la cual son aplicadas. El rango de decibelios obtenido está acotado en el rango [13, 5 . . . 35] aunque en la reconstrucción WENO no se alcanzan niveles tan elevados, siendo la cota superior de 29 db.

Llegados a este punto, para afrontar el análisis de las diferentes técnicas de reconstrucción estudiadas de una manera más específica, se propone dividir los métodos de interpolación en tres grupos atendiendo a la condición de «óptimalidad» frente al resto de técnicas, es decir, las técnicas quedan divididas en base a el número de *configuraciones* diferentes en las que ha obtenido un valor de PSNR mayor que las restantes técnicas. En este sentido la técnica PPH se ha clasificado en el grupo de alta calidad, pues ha resultado como óptima en diecinueve ocasiones, un 46 % de las veces. En el grupo de *calidad media* se han incluido las técnicas ENO (con partición tanto jerárquica como no jerárquica del *stencil*) y el método Racional, por resultar aceptables entre el 10 % y el 20 % de los casos, frente a las restantes. Y por último, el grupo de baja calidad, al que se ha representado en rojo y que contiene a WENO y a ENOSR. Ambas técnicas sólo han resultado favorables para cuatro configuraciones, concretamente dos por técnica.

En el histograma ??, el color de cada barra denota dichos grupos de calidad. El verde corresponde al grupo de *alta calidad*, mientras que el ámbar o amarillo y

el rojo se corresponden con los grupos de *calidad media* y *baja calidad*, respectivamente.

Grupo de *alta calidad*

En este grupo ha sido clasificada solamente la interpolación PPH. Esta técnica de reconstrucción ha obtenido los mejores valores de los indicadores aplicados con el objetivo de evaluar la calidad de las imágenes digitales presentadas y, consecuentemente, las mejores reconstrucciones en cuanto a calidad visual se refiere. Por un lado, se observa que la interpolación PPH obtiene los mejores resultados del indicador PSNR en todos los niveles de las imágenes **Geo** y **Pet-Tc1**. También consigue valores óptimos para las imágenes **Lena** y **Tac2** pero solo para los niveles 2, 3 y 4 de *zoom*. Destacar que la diferencia de ésta técnica con respecto a la que si que consigue valores óptimos para los primeros niveles de *zoom* en éstas dos imágenes, 0,3 respecto a la imagen **Lena**, y 0,05 en la imagen **Tac2**.

Resumiendo, si se desprecian estas diferencias (asumiendo el error incurrido) se podría considerar que ésta técnica es la que mejor resultado obtendría con imágenes fotográficas cotidianas (**Lena** y **Geo**) y particularmente con las **Tac2** y **Pet-Tc1**.

Por otro lado, en un nivel menor de calidad se encontraría la técnica ENO con una selección jerárquica del *stencil*. Ésta técnica consigue valores aceptables para las imágenes **Tac1** y **Pet-Tc2**. Obtiene los resultados óptimos para dichas imágenes en todos sus niveles, excepto en el tercero. En la imagen **Pet-Tc2** tampoco consigue el mayor valor en el primer nivel de *zoom*, pero para ése nivel la diferencia con el valor óptimo obtenido por la técnica WENO es de solo 0,29 db. Respecto al tercer nivel, el valor óptimo lo obtiene la técnica ENO jerárquico, con una diferencia la técnica ENO no jerárquico de 0,2106db para **Tac1**; y 0,0272 db para **Pet-Tc2**. En caso de despreciar éstas diferencias, se observa que ésta técnica sería óptima para las imágenes de **Tac1** y **Pet-Tc2**.

Grupo de *calidad media*

La técnica ENO con selección no jerárquica del *stencil* ha obtenido los resultados óptimos para el tercer nivel de las imágenes **tac1** y **Pet-Tc2**; y para el cuarto nivel de *zoom* con la imagen **Dicom1**. En el gráfico que muestra el diagrama de dispersión del PSNR obtenido por cada técnica (figura 4.38), se observa que los datos se distribuyen uniformemente para todos los niveles desde los 14 db hasta los 33 db, prácticamente igual que la técnica ENO mediante selección jerárquica. Además si se atiende al valor de la tabla 4.12, se puede verificar que el valor de la media y de la desviación típica no varía especialmente dentro de la técnica ENO, tanto si se procede a una división jerárquica del *stencil* como si no.

Analizando los resultados, la técnica Racional obtiene datos óptimos para tres

de cuatro niveles con los que se ha experimentado a partir de la imagen *Dicom2*, concretamente para el segundo, tercero y cuarto nivel. Es con la imagen *Dicom2*, la única que muestra resultados favorables para ésta técnica en los citados niveles, aunque cabe destacar que los valores obtenidos respecto a los mismos niveles para las otras técnicas como ENO, ENO jerárquico y PPH no dista más de 1 db. Esto puede ser debido a la difícil adquisición de la imagen que provoca que los contornos aparezcan difusos y la mayor parte de los píxeles de la imagen alberguen valores cercanos o iguales 255. En cualquier caso, que los resultados con la técnica Racional consigan buenos resultados en este caso no puede ser extrapolado a otras imágenes, consideración que si puede ser factible cuando se refiere a la técnica PPH. Basta con visualizar las imágenes correspondientes al segundo nivel de ampliación con cualquiera de las técnicas para que en general, no se consiguen los resultados que deberían. En [17] se proponen una serie de modificaciones a la interpolación Racional que podrían ser considerados en la elección de ésta técnica en algún otro contexto específico.

Grupo de *baja calidad*

En este tercer grupo, han sido clasificadas aquéllas técnicas que menos veces han resultado óptimas para alguna configuración dada o directamente que no han sido óptimas para ninguna de las configuraciones. Las dos técnicas pertenecientes a este grupo son la WENO y la ENO-SR. Resulta curioso sean éstas dos las que han sido agrupadas por obtener una *baja calidad*, ya que ambas se presentaban como modificaciones sobre la técnica ENO y a priori, prometían mejores resultados. En cualquier caso los resultados en este contexto serían orientativos, y se habría de ampliar el estudio para poder extrapolar los resultados otras imágenes utilizadas, o de otra naturaleza.

En primer lugar destacar que la calidad observada en las imágenes que han sido reconstruidas mediante la aplicación de la técnica WENO es peor que el obtenido por las otras imágenes sobre las que se ha realizado el experimento. La diferencia entre el PSNR medio correspondiente a las imágenes reconstruidas con un nivel de ampliación y la media de los valores obtenidos también con un nivel de *zoom* con WENO, es mayor o igual que 4 db en todos los casos; hecho que, atendiendo a los resultados obtenidos es sin duda considerable. En contraposición, conforme se verifican los valores de los niveles siguientes, se observa que las diferencias entre los valores medios obtenidos para éste método de reconstrucción y los obtenidos con los restantes métodos es menor, siendo la máxima de todas ellas 1 db. Mientras para las imágenes *Dicom1* y *Dicom2* los datos podrían resultar aceptables, para el resto de imágenes existen métodos que obtienen mejor calidad; hecho que se puede contrastar de manera visual. Todo esto conlleva a declarar al método de resolución WENO como el menos indicado para llevar a cabo cualquier tipo de reconstrucción

en este contexto.

En segundo lugar, la técnica ENOSR solo ha resultado mejor que las restantes en dos de los casos estudiados. Se trata del cuarto nivel de ampliación en la imagen **Pet-Tc1** y del valor obtenido en el primer nivel de la imagen **Pet-Tc2**. Dado que, por una parte, el valor obtenido en la primera de las imágenes es el mismo que el obtenido por la técnica PPH; y que por otra, el valor para el primer nivel que ha obtenido ésta técnica al ser aplicada sobre la imagen **Pet-Tc2** dista en solo 0,3 del denominado óptimo para ese nivel, puede procederse a descartar ésta técnica como idónea para llevar a cabo una reconstrucción con resultados aceptables.

Por todo lo expuesto y según los resultados obtenidos en este experimento, la técnica PPH se comporta de manera aceptable y obtiene mejores resultados que cualquier otra de las técnicas expuestas en este estudio. Esto ocurre en todos los tipos de imágenes empleadas, por lo que en esta memoria se considera que es la técnica que se debería utilizar para implementar un zoom digital, ya que se consiguen resultados en regiones suaves como lo haría un polinomio de Lagrange, y las regiones que presentan singularidades mediante la combinación de polinomios detallada anteriormente.

Dependiendo del contexto en el cual se trabaje y de la calidad requerida, podría ser substituida por la técnica ENO con partición no jerárquica y en algún otro caso por la misma técnica con partición jerárquica del *stencil*; siempre y cuando las discontinuidades presentes en las imágenes con las cuales se trabaje, estén bien alejadas unas de otras. También se ha de descartar cualquier suposición relativa a la existencia de alguna correlación entre el tipo de imagen empleada y el la técnica idónea de interpolación. Después de observar los resultados obtenidos, a grandes rasgos todas las técnicas provén resultados que numéricamente fluctúan de manera similar entre distintos tipos de imágenes, salvo en las aspectos y casos comentados.

Capítulo 5

Conclusiones

El presente proyecto ha tratado de profundizar en las técnicas actuales para proporcionar un *zoom* digital aplicable a todo tipo de imágenes digitales, especialmente a imágenes médicas como los son las gammagrafías. Se ha detallado para todas ellas su motivación, idoneidad para el problema propuesto y sus características, además de un pseudocódigo que muestra una posible implementación.

En la sección de experimentos, han sido realizadas diversas pruebas con cada una de las técnicas y han sido evaluadas evaluadas mediante el uso de ciertos indicadores para imágenes digitales, y ello ha puesto de manifiesto las ventajas e inconvenientes que cada técnica produce, a la vista de los valores obtenidos y las imágenes de las reconstrucciones. Si bien, es verdad que las reconstrucciones no lineales tienden a disminuir tanto el fenómeno el fenómeno de Gibbs que aparece en las discontinuidades como la difusión de contornos, características típicas de los métodos lineales; la técnica que mejores resultados ha tenido presenta más similitudes en cuanto a resultados que los algoritmos lineales, que el resto de algoritmos para la reconstrucción de imágenes.

Después de haber analizado los resultados, no se detecta ningún patrón o relación que establezca qué tipo de técnica utilizar para una determinada imagen. Dicho esto, la técnica que mejores resultados ha obtenido es la PPH tal y como se ha detallado, y que hacen de ésta una técnica idónea, fácil de implementar y eficiente de ser incluida en cualquier *software* que requiera un *zoom* digital.

Por último, como posible ampliación de este trabajo, se propone el diseño de un experimento similar pero aplicado sobre imágenes en color. Ello solo conllevaría aplicar la técnica interpolatoria deseada a los diferentes canales de color de la imagen y quizás pudiera poner de manifiesto aspectos que en este estudio, o agudizar otros que hayan que los resultados de este estudio no hayan puesto de manifiesto.

Anexo I

A continuación se muestra el código de la aplicación.

Las archivos que han sido codificados y que aparecen a continuación son `int_eno.m`, `int_enoSRm`, `int_weno.m`, `int_racional.m`, `int_pph.m`, `getMask.m`, e `indicadores.m`.

`int_eno.m`

Se trata de una reconstrucción siguiendo la técnica ENO mediante una reconstrucción de cuatro puntos, tanto con selección jerárquica, como no jerárquica. (Ver sección 3.3.1).

Contiene las funciones:

```
function [b] = int_eno(niv, a, met_particion)
```

La función `int_eno` recibe como parámetros de entrada un nivel de zoom `niv`, la imagen de partida `im`, y un booleano `met_particion` que, como su propio nombre indica, establece el método de partición empleado. El valor 0 especifica un método no jerárquico de selección del *stencil*, y el 1 un método jerárquico de selección. El vector de salida contiene los valores de v además de valores intercalados interpolados en forma de vector fila.

```
function [f] = eno_zoom(v, n, met_particion)
```

La función `eno_zoom` recibe como parámetros de entrada un vector, su longitud y un booleano denominado `met_particion`. El vector de salida es f , que contiene los valores de v además de valores intercalados interpolados en forma de vector fila.

`int_enoSR.m`

Este fichero alberga las funciones `int_enoSR`, `Stencil` y `enoSR_zoom`. Implementa una reconstrucción ENO-SR de cuatro puntos basada en la técnica ENO jerárquica. (Ver sección 3.3.2)

Implementa las funciones:

```
function [b] = int_enoSR(niv, a)
```

Esta función lleva a cabo la interpolación ENO *Sucell Resolution*. Recibe como parámetros los niveles de zoom y la imagen a y obtiene una ampliación de dicha imagen, b , de nivel niv . En ella, se llama a la función descrita a continuación, para cada fila y columna.

```
function [f] = enoSR_zoom(v, n)
```

La función `enoSR_zoom` recibe como parámetros de entrada un vector v y su longitud, y devuelve en f el vector de entrada con los valores intercalados interpolados.

`int_weno.m`

Este archivo contiene la implementación de las funciones `int_weno` y `weno_zoom` que llevan a cabo una interpolación WENO también de cuatro puntos. (Ver sección 3.3.3)

En él se encuentran implementadas las funciones siguientes:

```
function [b] = int_weno(niv, a)
```

Esta función lleva a cabo la interpolación WENO mediante una reconstrucción de cuatro puntos. Recibe como parámetros los niveles de zoom y la imagen a y obtiene una ampliación de dicha imagen, b , de nivel niv . Para cada fila y columna, se realiza una llamada a la función que sigue.

```
function [f] = weno_zoom(v, n)
```

La función `weno_zoom` recibe como parámetros de entrada un vector v y su longitud, y devuelve en f el vector de entrada con los valores intercalados interpolados.

`int_racional.m`

Contiene las funciones `int_racional` y `racional_zoom`, necesarias para llevar a cabo una interpolación racional. (Ver sección 3.3.4)

Contiene las siguientes funciones:

```
function [b] = int_racional(niv, a)
```

Ésta función lleva a cabo la interpolación de una imagen a , basándose en la técnica Racional, y con los niveles de *zoom* niv pasados como parámetro. Para cada fila y columna de a tiene lugar una llamada a la función `racional_zoom`.

```
function [f] = racional_zoom(v, n)
```

Ésta es la que lleva a cabo propiamente la reconstrucción Racional. Para cada vector fila v , de longitud n , se devuelve un vector f con los valores de v , además de los respectivos valores intercalados obtenidos mediante ésta técnica.

`int_pph.m`

Fichero que implementa las funciones `int_pph` y `pph_zoom`. Se trata también de una reconstrucción de cuatro puntos, pero llevada a cabo mediante una reconstrucción PPH. (Ver sección 3.3.5)

En él se encuentran implementadas:

```
function [b] = int_int_pph(niv, a)
```

Análogamente a las anteriores, ésta función recibe los niveles de *zoom* deseados y la imagen de partida, a ; y obtiene en b la imagen resultante del proceso.

```
function [f] = pph_zoom(v, n)
```

Ésta es la encargada de proceder a la reconstrucción de cada vector fila recibido como parámetro, v , a partir de éste y de su longitud n . Como salida obtiene f , que consta de los valores del vector v recibido, además de aquellos que han sido obtenidos como resultado de la aplicación de la técnica PPH.

`getMask.m`

Esta función calcula las máscaras necesarias para el esquema de subdivisión basado en Lagrange. Su cabecera es `function [mas, numer, deno]=getMask(1, r)`

Los argumentos de entrada son `l` y `r`, que son respectivamente la cantidad de puntos existentes a derecha e izquierda del punto de interés en el stencil seleccionado. Como salida se obtienen `mas`,

`indicadores.m`

En este archivo se encuentra el código de los diferentes indicadores implementados. Las funciones que los obtienen son, respectivamente, `MeanSquareError`, `PeakSignaltoNoiseRatio`, `NormalizedCrossCorrelation`, `AverageDifference`, `MaximumDifference` y `NormalizedAbsoluteError`. Todas ellas reciben como entrada dos imágenes: una de referencia, `origImg`, y otra imagen `distImage` que en este caso es una aproximación a la primera.

```
function ECM = MeanSquareError(origImg, distImg)
```

La salida que produce la función es el Error Cuadrático Medio, definido anteriormente en el punto 4.1.2.

```
function PSNR = PeakSignaltoNoiseRatio(origImg, distImg)
```

Como resultado obtiene el PSNR según se ha definido en la sección 4.1.3.

```
function NK = NormalizedCrossCorrelation(origImg, distImg)
```

Produce el Coeficiente de Correlación Normalizada, tal y como se ha visto anteriormente en 4.1.4.

```
function AD = AverageDifference(origImg, distImg)
```

Calcula la diferencia media entre las dos imágenes de entrada. Ver sección 4.1.5.

```
function MD = MaximumDifference(origImg, distImg)
```

Obtiene la diferencia máxima entre dos píxeles como se ha visto en 4.1.6.

```
function EAN = NormalizedAbsoluteError(origImg, distImg)
```

Da como resultado el Error Absoluto Normalizado, definido en la sección ??.

```
%*****
% Main
%*****

% Programa para la obtención de indicadores de suavidad para imágenes
% digitales.

%ENTRADA:
% origImg - Imagen de referencia.
% origImg - Imagen interpolada.

%SALIDA:
% Se muestran por salida estándar los siguientes indicadores:
% ECM - Error Cuadrático Medio
% PSNR - Señal a ruido de pico
% NK - Coeficiente de Correlación Cruzada Normalizada
% AM - Diferencia Media
% DM - Diferencia Máxima
% EAN - Error Absoluto Normalizado

function main(origImg, distImg);
%Error Cuadrático Medio.
ECM = MeanSquareError(origImg, distImg);
disp('Error cuadratico medio = ');
disp(ECM);

%Peak Signal to Noise Ratio.
PSNR = PeakSignaltoNoiseRatio(origImg, distImg);
disp('Peak Signal to Noise Ratio = ');
disp(PSNR);

%Coeficiente de Correlación Normalizada.
NK = NormalizedCrossCorrelation(origImg, distImg);
disp('Coeficiente de Correlacion Cruzada Normalizado = ');
disp(NK);

%Diferencia Media.
AD = AverageDifference(origImg, distImg);
disp('Diferencia Media = ');
disp(AD);

%Máxima Diferencia.
DM = MaximumDifference(origImg, distImg);
disp('Diferencia Maxima = ');
disp(DM);

%Error Absoluto Normalizado
EAN = NormalizedAbsoluteError(origImg, distImg);
disp('Error Absoluto Normalizado = ');
disp(EAN);

%*****
% Error Cuadrático Medio
```

```
%*****

%ENTRADA:
% origImg - Imagen de referencia.
% distImg - Imagen interpolada.

%SALIDA:
% ECM - Error Cuadrático Medio.

%*****
% Error Cuadrático Medio
%*****

%ENTRADA:
% origImg - Imagen de referencia.
% distImg - Imagen interpolada.

%SALIDA:
% ECM - Error Cuadrático Medio.

function ECM = MeanSquareError(origImg, distImg)

Image1 = double(origImg);
Image2 = double(distImg);
ECM = (sum((Image1(:) - Image2(:)).^2)/numel(Image1));

%*****
% PSNR
%*****

%ENTRADA:
% origImg - Imagen de referencia.
% origImg - Imagen interpolada.

%SALIDA:
% PSNR - Peak Signal to Noise Ratio

function PSNR = PeakSignaltoNoiseRatio(origImg, distImg)

origImg = double(origImg);
distImg = double(distImg);
PSNR = 20*log10(255/sqrt(MeanSquareError(origImg, distImg)));
end

%*****
% Coeficiente de Correlación Cruzada Normalizado
%*****

%ENTRADA:
% origImg - Imagen de referencia.
% origImg - Imagen interpolada.

%SALIDA:
```



```
% NK - Coeficiente de Correlación Cruzada Normalizado

function NK = NormalizedCrossCorrelation(origImg, distImg)

origImg = double(origImg);
distImg = double(distImg);

NK = sum(sum(origImg .* distImg)) / sum(sum(origImg .* origImg));
end

%*****
% Diferencia Media
%*****

%ENTRADA:
% origImg - Imagen de referencia.
% distImg - Imagen interpolada.

%SALIDA:
% DM - Diferencia Media.

function DM = AverageDifference(origImg, distImg)

origImg = double(origImg);
distImg = double(distImg);

[M N] = size(origImg);
error = origImg - distImg;

DM = sum(sum(error)) / (M * N);
end

%*****
% Diferencia Máxima
%*****

%ENTRADA:
% origImg - Imagen de referencia.
% distImg - Imagen interpolada.

%SALIDA:
% MD - Diferencia Máxima.

function DM = MaximumDifference(origImg, distImg)

origImg = double(origImg);
distImg = double(distImg);

error = origImg - distImg;

DM = max(max(error));
end
```

```
%*****  
% Error Absoluto Normalizado  
%*****  
  
%ENTRADA:  
% origImg - Imagen de referencia.  
% distImg - Imagen interpolada.  
  
%SALIDA:  
% EAN - Error Cuadrático Medio  
  
function EAN = NormalizedAbsoluteError(origImg, distImg)  
  
origImg = double(origImg);  
distImg = double(distImg);  
  
error = origImg - distImg;  
  
EAN = sum(sum(abs(error))) / sum(sum(origImg));  
end
```

```
%*****
% Interpolacion ENO
%*****

%Función: int_eno
%*****

%ENTRADA:
% niv - Niveles de zoom.
% a - Imagen a interpolar.
% met_particion - 0 para no jerarquico, 1 para jerarquico..

%SALIDA:
% b - Imagen interpolada.

function [b] = int_eno(niv, a, met_particion)

[n m] = size(a);

% Bulce para los niveles de zoom.
for k=1:niv

    % Filas y columnas para la imagen interpolada.
    n_filas = 2*n; n_columnas = 2*m;

    % Creamos la nueva matriz.
    b = zeros(n_filas, n_columnas);
    b(1:2:n_filas, 1:2:n_columnas) = a(1:n, 1:m);

    % Algoritmo para la prediccion de las columnas.
    for j=1:n_columnas
        b(1:n_filas,j) = eno_zoom(b(1:2:n_filas, j), n, met_particion)';
    end

    % Algoritmo para la prediccion de las filas.
    for i =1:n_filas
        b(i, 1:n_columnas) = eno_zoom(b(i,1:2:n_columnas), m, met_particion);
    end

    % Acutalizaciones de las variables.
    n = n_filas;
    m = n_columnas;

    % Inicializacion matriz de zoom.
    a = b;
end
b = uint8(b);
end

% Función: eno_zoom
% *****

% ENTRADA:
```

```

% v - Vector fila al que se le va a aplicar el algoritmo.
% n - Longitud del vector v.
% met_particion - 0 para jerarquico, 1 para no jerarquico..
% SALIDA:
% f - Vector fila con valores interpolados.

function f = eno_zoom(v, n, met_particion)

% Creamos el vector f, vector de salida de la funcion.
f = zeros(1, 2*n);
f(1:2:2*n-1) = v(1:n);

% Obtencion de la mascarar ENO.
[m1] = getMask(4/2, 4/2);
[m3] = getMask(4/2+1, 4/2-1);
[m2] = getMask(4/2-1, 4/2+1);

% Prediccion del primer elemento con mascara lineal;
f(2) = m2(1)*v(1) + m2(2)*v(2) + m2(3)*v(3) + m2(4)*v(4);

if(met_particion == 0) % metodo no querarquico

    % Prediccion del segundo elemento.

    % Cálculo de las diferencias divididas.
    elem2 = abs(-v(1) + 3*v(2) - 3*v(3) + v(4));
    elem3 = abs(-v(2) + 3*v(3) - 3*v(4) + v(5));
    if(elem2<elem3)
        f(4) = m1(1)*v(1) + m1(2)*v(2) + m1(3)*v(3) + m1(4)*v(4);
    else
        f(4) = m2(1)*v(2) + m2(2)*v(3) + m2(3)*v(4) + m2(4)*v(5);
    end
    % Prediccion valores intermedios.
    for j=6:2:2*n-6
        elem1 = abs(-v(j/2-2) + 3*v(j/2-1) - 3*v(j/2) + v(j/2+1));
        elem2 = abs(-v(j/2-1) + 3*v(j/2) - 3*v(j/2+1) + v(j/2+2));
        elem3 = abs(-v(j/2) + 3*v(j/2+1) - 3*v(j/2+2) + v(j/2+3));
        if(elem2 <= elem1 && elem2 < elem3)
            f(j) = m1(1)*v(j/2-1) + m1(2)*v(j/2) + m1(3)*v(j/2+1)+ m1(4)*v(j/2+2); % Si
es s2
        else
            if(elem1 <= elem3)
                f(j) = m3(1)*v(j/2-2) + m3(2)*v(j/2-1) + m3(3)*v(j/2) + m3(4)*v(j/2+1);
            else
                f(j) = m2(1)*v(j/2) + m2(2)*v(j/2+1) + m2(3)*v(j/2+2) + m2(4)*v(j/2+3);
            end
        end
    end
end
% Prediccion del penultitmo elemento.
elem1 = abs(-v(n-4) + 3*v(n-3) - 3*v(n-2) + v(n-1));
elem2 = abs(-v(n-3) + 3*v(n-2) - 3*v(n-1) + v(n));
if(elem2 <= elem1)
    f(2*n-4) = m1(1)*v(n-3) + m1(2)*v(n-2) + m1(3)*v(n-1) + m1(4)*v(n);
else
    f(2*n-4) = m3(1)*v(n-4) + m3(2)*v(n-3) + m3(3)*v(n-2) + m3(4)*v(n-1);
end

```

```

end
% Prediccion del ultimo elemento
f(2*n-2) = m2(1)*v(n) + m2(2)*v(n-1) + m2(3)*v(n-2) + m2(4)*v(n-3);

else %met_particion = 'jerarquico'
elem1 = (v(2) - v(1));
elem2 = (v(3) - v(2));
elem3 = (v(4) - v(3));
elem4 = (v(5) - v(4));
dif1 = (elem2 - elem1); a_dif1 = abs(dif1);
dif2 = (elem3 - elem2); a_dif2 = abs(dif2);

if(a_dif1 < a_dif2)
    f(4) = m1(1)*v(1) + m1(2)*v(2) + m1(3)*v(3) + m1(4)*v(4);
else
    dif3 = elem4 - elem3;
    aux1= (dif2 - dif1); a_aux1 = abs(aux1);
    aux2 = (dif3 - dif2); a_aux2 = abs(aux2);
    if(a_aux1 <= a_aux2)
        f(4) = m1(1)*v(1) + m1(2)*v(2) + m1(3)*v(3) + m1(4)*v(4);
    else
        f(4) = m2(1)*v(2) + m2(2)*v(3) + m2(3)*v(4) + m2(4)*v(5);
    end
end

end
for j=6:2:2*n-6
    elem1 = (v(j/2-1) - v(j/2-2));
    elem2 = (v(j/2) - v(j/2-1));
    elem3 = (v(j/2+1) - v(j/2));
    elem4 = (v(j/2+1) - v(j/2+1));
    elem5 = (v(j/2+3) - v(j/2+2));
    dif1 = (elem2 - elem1);
    dif2 = (elem3 - elem2); a_dif2 = abs(dif2);
    dif3 = (elem4 - elem3); a_dif3 = abs(dif3);
    dif4 = (elem5 - elem4);
    if(a_dif2 <= a_dif3)
        aux1 = (elem2 - elem1); a_aux1 = abs(aux1);
        aux2 = (elem3 - elem2); a_aux2 = abs(aux2);
        if(a_aux2 <= a_aux1)
            f(j) = m1(1)*v(j/2-1) + m1(2)*v(j/2) + m1(3)*v(j/2+1)+ m1(4)*v(j/2+2);
        else
            f(j) = m3(1)*v(j/2-2) + m3(2)*v(j/2-1) + m3(3)*v(j/2) + m3(4)*v(j/2+1);
        end
    else
        aux2 = (dif3 - dif2); a_aux2 = abs(aux2);
        aux3= (dif4 - dif3); a_aux3 = abs(aux3);
        if(a_aux2 <= a_aux3)
            f(j) = m1(1)*v(j/2-1) + m1(2)*v(j/2) + m1(3)*v(j/2+1)+ m1(4)*v(j/2+2);
        else
            f(j) = m2(1)*v(j/2) + m2(2)*v(j/2+1) + m2(3)*v(j/2+2) + m2(4)*v(j/2+3);
        end
    end
end
end
% Prediccion penultimo elemento
elem1 = (v(n-3) - v(n-4));

```

```

end
% Prediccion del ultimo elemento
f(2*n-2) = m2(1)*v(n) + m2(2)*v(n-1) + m2(3)*v(n-2) + m2(4)*v(n-3);

else %met_particion = 'jerarquico'
elem1 = (v(2) - v(1));
elem2 = (v(3) - v(2));
elem3 = (v(4) - v(3));
elem4 = (v(5) - v(4));
dif1 = (elem2 - elem1); a_dif1 = abs(dif1);
dif2 = (elem3 - elem2); a_dif2 = abs(dif2);

if(a_dif1 < a_dif2)
    f(4) = m1(1)*v(1) + m1(2)*v(2) + m1(3)*v(3) + m1(4)*v(4);
else
    dif3 = elem4 - elem3;
    aux1= (dif2 - dif1); a_aux1 = abs(aux1);
    aux2 = (dif3 - dif2); a_aux2 = abs(aux2);
    if(a_aux1 <= a_aux2)
        f(4) = m1(1)*v(1) + m1(2)*v(2) + m1(3)*v(3) + m1(4)*v(4);
    else
        f(4) = m2(1)*v(2) + m2(2)*v(3) + m2(3)*v(4) + m2(4)*v(5);
    end
end

end
for j=6:2:2*n-6
    elem1 = (v(j/2-1) - v(j/2-2));
    elem2 = (v(j/2) - v(j/2-1));
    elem3 = (v(j/2+1) - v(j/2));
    elem4 = (v(j/2+1) - v(j/2+1));
    elem5 = (v(j/2+3) - v(j/2+2));
    dif1 = (elem2 - elem1);
    dif2 = (elem3 - elem2); a_dif2 = abs(dif2);
    dif3 = (elem4 - elem3); a_dif3 = abs(dif3);
    dif4 = (elem5 - elem4);
    if(a_dif2 <= a_dif3)
        aux1 = (elem2 - elem1); a_aux1 = abs(aux1);
        aux2 = (elem3 - elem2); a_aux2 = abs(aux2);
        if(a_aux2 <= a_aux1)
            f(j) = m1(1)*v(j/2-1) + m1(2)*v(j/2) + m1(3)*v(j/2+1)+ m1(4)*v(j/2+2);
        else
            f(j) = m3(1)*v(j/2-2) + m3(2)*v(j/2-1) + m3(3)*v(j/2) + m3(4)*v(j/2+1);
        end
    else
        aux2 = (dif3 - dif2); a_aux2 = abs(aux2);
        aux3= (dif4 - dif3); a_aux3 = abs(aux3);
        if(a_aux2 <= a_aux3)
            f(j) = m1(1)*v(j/2-1) + m1(2)*v(j/2) + m1(3)*v(j/2+1)+ m1(4)*v(j/2+2);
        else
            f(j) = m2(1)*v(j/2) + m2(2)*v(j/2+1) + m2(3)*v(j/2+2) + m2(4)*v(j/2+3);
        end
    end
end
end
% Prediccion penultimo elemento
elem1 = (v(n-3) - v(n-4));

```

```
%*****
% Interpolacion ENO-SR
%*****

%Función: int_enoSR
%*****

%ENTRADA:
% niv - Niveles de zoom.
% a - Imagen a interpolar.

%SALIDA:
% b - Imagen interpolada.

function [ b ] = int_enoSR(niv, a)
[n m] = size(a);

% Bulce para los niveles de zoom.
for k=1:niv
% Filas y columnas de la imagen interpolada.
    n_filas = 2*n;
    n_columnas = 2*m;

% Creamos la nueva matriz.
    b = zeros(n_filas, n_columnas);
    b(1:2:n_filas, 1:2:n_columnas) = a(1:n, 1:m);

% Algoritmo de prediccion para las columnas.
    for j=1:2:n_columnas
        b(1:n_filas, j) = enoSR_zoom(b(1:2:n_filas, j), n)';
    end

% Algoritmo de preciccion para las filas.
    for i =1:n_filas
        b(i, 1:n_columnas) = enoSR_zoom(b(i,1:2:n_columnas), m);
    end

% Actualización de variables.
    n = n_filas;
    m = n_columnas;

% Inicialización de la matriz de zoom.
    a = b;
end

b = uint8(b);
end

% Función: enosr_zoom
% *****

% ENTRADA:
% v - Vector fila al que se le va a aplicar el algoritmo.
% n - Longitud del vector v.
```

```
% SALIDA:
% f - Vector fila con valores interpolados.

function [f] = enoSR_zoom(v, n)

% Cramos el vector f, vector de salida de la funcion.
f = zeros(1, 2*n);
f(1:2:2*n-1) = v(1:n);

% Obtencion de la mascaras.

[m1] = getMask(orden/2, orden/2);
[m3] = getMask(orden/2+1, orden/2-1);
[m2] = getMask(orden/2-1, orden/2+1);

% Prediccion del primer elemento con mascara lineal;
f(2) = m2(1)*v(1) + m2(2)*v(2) + m2(3)*v(3) + m2(4)*v(4);

% Prediccion del segundo elemento.

% Cálculo de las diferencias divididas.
elem1 = (v(2) - v(1));
elem2 = (v(3) - v(2));
elem3 = (v(4) - v(3));
elem4 = (v(5) - v(4));
dif1 = (elem2 - elem1); a_dif1 = abs(dif1);
dif2 = (elem3 - elem2); a_dif2 = abs(dif2);

if(a_dif1 <= a_dif2)
    f(4) = m1(1)*v(1) + m1(2)*v(2) + m1(3)*v(3) + m1(4)*v(4);
else
    dif3 = (elem4 - elem3);
    t1 = dif2 - dif1; a_t1 = abs(t1);
    t2 = dif3 - dif2; a_t2 = abs(t2);
    if(a_t1 <= a_t2)
        f(4) = m1(1)*v(1) + m1(2)*v(2) + m1(3)*v(3) + m1(4)*v(4);
    else
        f(4) = m2(1)*v(2) + m2(2)*v(3) + m2(3)*v(4) + m2(4)*v(5);
    end
end

% Prediccion del tercer elemento con ENO jerarquico.
elem1 = (v(2) - v(1));
elem2 = (v(3) - v(2));
elem3 = (v(4) - v(3));
elem4 = (v(5) - v(4));
elem5 = (v(6) - v(5));
dif1 = (elem2 - elem1); a_dif1 = abs(dif1);
dif2 = (elem3 - elem2); a_dif2 = abs(dif2);
dif3 = (elem4 - elem3); a_dif3 = abs(dif3);
dif4 = (elem5 - elem4); a_dif4 = abs(dif4);

if(a_dif2<=a_dif3)
    t1 = a_dif2 - a_dif1; a_t1 = abs(t1);
```



```

t2 = a_dif3 - a_dif2; a_t2 = abs(t2);
if(a_t2 <= a_t1)
    f(6) = m1(1)*v(2) + m1(2)*v(3) + m1(3)*v(4)+ m1(1)*v(5);
else
    f(6) = m3(1)*v(1) + m3(2)*v(2) + m3(3)*v(3) + m3(4)*v(4);
end
else
t2 = dif2 - dif2; a_t2= abs(t2);
t3 = dif4 - dif3; a_t3 = abs(t3);
if(a_t2 <= a_t3)
    f(6) = m1(1)*v(3) + m1(2)*v(4) + m1(3)*v(5)+ m1(1)*v(6);
else
    f(6) = m3(1)*v(3) + m3(2)*v(4) + m3(3)*v(5) + m(4)*v(6);
end
end

% Prediccion de valores intermedios
for j=8:2:2*n-8
    %Paso 1 : para cada intervalo determinamos los stencil a la izquierda
    %y a la derecha.. Si no esta descentrado aplicamos eno-h.
    s_izq = stencil(v(j/2-3:j/2+2));
    s_der = stencil(v(j/2-1:j/2+4));
    if(s_izq == 'I' & s_der == 'D')
        %Continuamos con el metodode resolucio n subcelda comprobamos si
        %g(x_i*g(x_(i+1))) <0.
        q11 = v(j/2);
        q12 = -v(j/2-1) + 4*v(j/2-2) - 6*v(j/2-1) + 4*v(j/2);
        q21 = 3*v(j/2+1) - 6*v(j/2+2)+ 4*v(j/2+3) - v(j/2+4);
        q22 = (j/2+1);

        g1 = q21 - q11;
        g2 = q22 - q12;

        if(g1*g2 < 0)
            %Hay una discontinuidad en el intervalo.
            q1m = -5/16*v(j/2-3) + 21/16*v(j/2-2) - 35/16*v(j/2-1) + 35/16*v(j/2);
            q2m = 35/16*v(j/2+1) - 35/16*v(j/2+2) - 21/16*v(j/2+3) - 5/16*v(j/2+4);

            %funcion g evaluada en punto medio.
            gm = q2m - q1m;
            if(g1*gm < 0)
                f(j) = q2m;
            else
                f(j)= q1m;
            end
        else
            %ENO jerarquico.
            p1 = v(j/2-1) - v(j/2-2);
            p2 = v(j/2) - v(j/2-1);
            p3 = v(j/2+1) - v(j/2);
            p4 = v(j/2+2) - v(j/2+1);
            p5 = v(j/2+3) - v(j/2+2);
            dif1 = p2 - p1;
            dif2 = p3 - p2;
            dif3 = p4 - p3;

```

```

dif4 = p5 - p4;

if(as2 <= as3)
    aux1 = dif2 - dif1; a_aux1 = abs(aux1);
    aux2 = dif3 - dif2; a_aux2 = abs(aux2);

    if(a_aux2 <= a_aux1)
        f(j) = m1(1)*v(1) + m1(2)*v(2) + m1(3)*v(3) + m1(4)*v(4);
    else
        f(j) = m3(1)*v(j/2-2) + m3(2)*v(j/2-1) + m3(3)*v(j/2) + m3(4)*v(j/2+1);
    end
else
    aux2 = dif3 - dif2; a_aux2 = abs(aux2);
    aux3 = dif3 - dif3; a_aux3 = abs(aux3);
    if(a_aux2 <= a_aux3)
        f(j) = m1(1)*v(1) + m1(2)*v(2) + m1(3)*v(3) + m1(4)*v(4);
    else
        f(j) = m2(1)*v(j/2) + m2(2)*v(j/2+1) + m2(3)*v(j/2+2) + m2(4)*v(j/2+3);
    end
end
end
else
    % Aplicamos ENO jerárquico.
    elem1 = (v(j/2-1) - v(j/2-2));
    elem2 = (v(j/2) - v(j/2-1));
    elem3 = (v(j/2+1) - v(j/2));
    elem4 = (v(j/2+1) - v(j/2+1));
    elem5 = (v(j/2+3) - v(j/2+2));
    dif1 = (elem2 - elem1);
    dif2 = (elem3 - elem2); a_dif2 = abs(dif2);
    dif3 = (elem4 - elem3); a_dif3 = abs(dif3);
    dif4 = (elem5 - elem4);

    if(a_dif2 <= a_dif3)
        aux1 = (elem2 - elem1); a_aux1 = abs(aux1);
        aux2 = (elem3 - elem2); a_aux2 = abs(aux2);
        if(a_aux2 <= a_aux1)
            f(j) = m1(1)*v(j/2-1) + m1(2)*v(j/2) + m1(3)*v(j/2+1) + m1(4)*v(j/2+2);
        else
            f(j) = m3(1)*v(j/2-2) + m3(2)*v(j/2-1) + m3(3)*v(j/2) + m3(4)*v(j/2+1);
        end
    else
        aux2 = (dif3 - dif2); a_aux2 = abs(aux2);
        aux3 = (dif4 - dif3); a_aux3 = abs(aux3);
        if(a_aux2 <= a_aux3)
            f(j) = m1(1)*v(j/2-1) + m1(2)*v(j/2) + m1(3)*v(j/2+1) + m1(4)*v(j/2+2);
        else
            f(j) = m2(1)*v(j/2) + m2(2)*v(j/2+1) + m2(3)*v(j/2+2) + m2(4)*v(j/2+3);
        end
    end
end
end
end
end

```

```

% Prediccion penultimo elemento.
elem1 = (v(n-4) - v(n-5));
elem2 = (v(n-3) - v(n-4));
elem3 = (v(n-2) - v(n-3));
elem4 = (v(n-1) - v(n-2));
elem5 = (v(n) - v(n-1));
dif1 = (elem2 - elem1); a_dif1 = abs(dif1);
dif2 = (elem3 - elem2); a_dif2 = abs(dif2);
dif3 = (elem4 - elem3); a_dif3 = abs(dif3);
dif4 = (elem5 - elem4);
if(a_dif2 <= a_dif3)
    aux1 = (elem2 - elem1); a_aux1 = abs(aux1);
    aux2 = (elem3 - elem2); a_aux2 = abs(aux2);
    if(a_aux2 <= a_aux1)
        f(2*n-6) = m1(1)*v(n-4) + m1(2)*v(n-3) + m1(3)*v(n-2) + m1(4)*v(n-1);
    else
        f(2*n-6) = m3(1)*v(n-5) + m3(2)*v(n-4) + m3(3)*v(n-3) + m3(4)*v(n-2);
    end
else
    aux2 = (elem3 - elem2); a_aux2 = abs(aux2);
    aux3 = (elem4 - elem3); a_aux3 = abs(aux2);
    if(a_aux2 <= a_aux3)
        f(2*n-6) = m1(1)*v(n-4) + m1(2)*v(n-3) + m1(3)*v(n-2) + m1(4)*v(n-1);
    else
        f(2*n-6) = m2(1)*v(n-3) + m2(2)*v(n-2) + m2(3)*v(n-1) + m2(4)*v(n);
    end
end

% Prediccion penultimo elemento.
elem1 = (v(n-3) - v(n-4));
elem2 = (v(n-2) - v(n-3));
elem3 = (v(n-1) - v(n-2));
elem4 = (v(n) - v(n-1));
dif1 = (elem2 - elem1); a_dif1 = abs(dif1);
dif2 = (elem3 - elem2); a_dif2 = abs(dif2);
dif3 = (elem4 - elem3); a_dif3 = abs(dif3);
if(a_dif3 <= a_dif2)
    f(2*n-4) = m1(1)*v(n-3) + m1(2)*v(n-2) + m1(3)*v(n-1) + m1(4)*v(n);
else
    aux1 = (dif2 - dif1); a_aux1 = abs(aux1);
    aux2 = (dif3 - dif2); a_aux2 = abs(aux2);
    if(a_aux2 <= a_aux1)
        f(2*n-4) = m1(1)*v(n-3) + m1(2)*v(n-2) + m1(3)*v(n-1) + m1(4)*v(n);
    else
        f(2*n-4) = m3(1)*v(n-4) + m3(2)*v(n-3) + m3(3)*v(n-2) + m3(4)*v(n-1);
    end
end

% Prediccion del ultimo elemento.
f(2*n-2) = m2(1)*v(n) + m2(2)*v(n-1) + m2(3)*v(n-2) + m2(4)*v(n-3);
end

% Función: stencil
% *****

% ENTRADA:

```

```
% v - .  
  
% SALIDA:  
% r - .  
  
function [r] = stencil(v)  
    elem1 = (v(2) - v(1));  
    elem2 = (v(3) - v(2));  
    elem3 = (v(4) - v(3));  
    elem4 = (v(5) - v(4));  
    elem5 = (v(6) - v(5));  
    dif1 = (elem2 - elem1); a_dif1 = abs(dif1);  
    dif2 = (elem3 - elem2); a_dif2 = abs(dif2);  
    dif3 = (elem4 - elem3); a_dif3 = abs(dif3);  
    dif4 = (elem5 - elem4);  
  
    if(a_dif2 <= a_dif3)  
        t1 = dif2 - dif1; a_t1 = abs(t1);  
        t2 = dif3 - dif2; a_t2 = abs(t2);  
        if(a_t2 <= a_t1)  
            r = 'C';  
        else  
            r = 'I';  
        end  
    else  
        t2 = dif3 - dif2; a_t2 = abs(t2);  
        t3 = dif4 - dif3; a_t3 = abs(t3);  
        if(a_t2 <= a_t3)  
            r = 'C';  
        else  
            r = 'I';  
        end  
    end  
end  
end
```

```
%*****  
% Interpolacion WENO  
%*****  
  
%Función: int_weno  
%*****  
  
%ENTRADA:  
% niv - Niveles de zoom.  
% a - Imagen a interpolar.  
  
%SALIDA:  
% b - Imagen interpolada.  
  
function [ b ] = int_weno(niv, a)  
  
[n m] = size(a);  
  
%Bucle para los niveles de zoom.  
for k=1:niv  
  
    % Filas y columnas para la imagen interpolada.  
    n_filas = 2*n;  
    n_columnas = 2*m;  
  
    % Creamos la nueva matriz.  
    b=zeros(n_filas,n_columnas);  
    b(1:2:n_filas,1:2:n_columnas) = a(1:n,1:m);  
  
    %Algoritmo de prediccion para las columnas.  
    for j=1:2:n_columnas  
        b(1:n_filas,j) = weno_zoom(b(1:2:n_filas, j), n)';  
    end  
  
    %Algoritmo de preciccion para las filas.  
    for i =1:n_filas  
        b(i, 1:n_columnas) = weno_zoom(b(i,1:2:n_columnas),m);  
    end  
  
    %Actualización de variables.  
    n = n_filas;  
    m = n_columnas;  
  
    % Inicializacion matriz de zoom.  
    a = b;  
end  
  
b = uint8(b);  
end  
  
% Función: weno_zoom  
% *****
```

```

% ENTRADA:
% v - Vector fila al que se le va a aplicar el algoritmo.
% n - Longitud del vector v.

% SALIDA:
% f - Vector fila con valores interpolados.

function [f] = weno_zoom(v, n)
f = zeros(1, 2*n);
f(1:2:2*n-1) = v(1:n);

%Obtencion de la mascara ENO.
[m1] = getMask(orden/2, orden/2);
[m3] = getMask(orden/2-1, orden/2+1);
[m2] = getMask(orden/2+1, orden/2);

%Prediccion del primer elemento.
f(2) = m1(1)*v(1) + m1(2)*v(2) + m1(3)*v(3) + m1(4)*v(4);

%Prediccion del segundo elemento.
elem2 = abs(-v(1) + 3*v(2) - 3*v(3) + v(4));
elem3 = abs(-v(2) + 3*v(3) - 3*v(4) + v(5));
if(elem2 < elem3)
    f(4) = m1(1)*v(1) + m1(2)*v(2) + m1(3)*v(3) + m1(4)*v(4);
else
    f(4) = m2(1)*v(2) + m2(2)*v(3) + m2(3)*v(4) + m2(4)*v(5);
end
%Prediccion valores intermedios
epsilon = 10^(-6);
for j=6:2:2*n-6
    elem1 = (v(j/2-1) - v(j/2-2));
    elem2 = (v(j/2) - v(j/2-1));
    elem3 = (v(j/2+1) - v(j/2));
    elem4 = (v(j/2+2) - v(j/2+1));
    elem5 = (v(j/2+3) - v(j/2+2));

    %Calculo de los indicadores de suavidad:
    i_izq = 1/2 * ((elem2 - elem1)^2 + (elem3 - elem2)^2) + (elem3 - 2*elem2 + elem1)^2;
    i_central = 1/2 * ((elem3 - elem2)^2 + (elem4 - elem3)^2) + (elem4 - 2*elem3 +
elem2)^2;
    i_der = 1/2 * ((elem4 - elem3)^2 + (elem5 - elem4)^2) + (elem5 - 2*elem4 + elem3)^2;

    %Calulo de los numeradores delos pesos.
    alfa_izq = 3./16/(epsilon + i_izq)^3;
    alfa_der = 10./16/(epsilon + i_der)^3;
    alfa_central = 3./16/(epsilon + i_central)^3;
    peso_izq = alfa_izq/(alfa_izq + alfa_der + alfa_central);
    peso_der = alfa_der/(alfa_izq + alfa_der + alfa_central);
    peso_central = alfa_central/(alfa_izq + alfa_der + alfa_central);

    %Suma de los 3 polinomios por sus pesos.
    q_central = m1(1)*v(j/2-1) + m1(2)*v(j/2) + m1(3)*v(j/2+1) + m1(4)*v(j/2+2);
    q_izquierdo = m3(1)*v(j/2-2) + m3(2)*v(j/2-1) + m3(3)*v(j/2) + m3(4)*v(j/2+1);
    q_derecho = m2(1)*v(j/2) + m2(2)*v(j/2+1) + m2(3)*v(j/2+2) + m2(4)*v(j/2+3);
    f(j) = peso_izq*q_izquierdo + peso_der*q_derecho + peso_central*q_central;

```

end

%Predicción del pultimo elemento.

elem1 = abs(-v(n-4) + 3*v(n-3) - 3*v(n-2) + v(n-1));

elem2 = abs(-v(n-3) + 3*v(n-2) - 3*v(n-1) + v(n));

if(elem2 <= elem1)

 %f(2*n-4) = (-v(n-3)+9*v(n-2)+9*v(n-1)-v(n))/16;

 f(2*n-4) = m1(1)*v(n-3) + m1(2)*v(n-2) + m1(3)*v(n-1) + m1(4)*v(n);

else

 f(2*n-4) = m3(1)*v(n-4) + m3(2)*v(n-3) + m3(3)*v(n-2) + m3(4)*v(n-1);

end

%prediccion del ultimo elemento

f(2*n-2) = m2(1)*v(n) + m2(2)*v(n-1) + m2(3)*v(n-2) + m2(4)*v(n-3);

end

```
%*****
% Interpolacion Racional
%*****

%Función: int_racional
%*****

%ENTRADA:
% niv - Niveles de zoom.
% a - Imagen a interpolar.

%SALIDA:
% b - Imagen interpolada.

function [ b ] = int_racional(niv, a)

[n m] = size(a);

%Bucle para los niveles de zoom.
for k=1:niv

    % Filas y columnas para la imagen interpolada.
    n_filas = 2*n;
    n_columnas = 2*m;

    % Creamos la nueva matriz.
    b=zeros(n_filas,n_columnas);
    b(1:2:n_filas,1:2:n_columnas) = a(1:n,1:m);

    % Algoritmo de prediccion para las columnas.
    for j=1:2:n_columnas
        b(1:n_filas,j) = racional_zoom(b(1:2:n_filas, j), n)';
    end

    % Algoritmo de preciccion para las filas.
    for i =1:n_filas
        b(i, 1:n_columnas) = racional_zoom(b(i,1:2:n_columnas),m);
    end

    %Actualización de variables.
    n = n_filas;
    m = n_columnas;

    % Inicializacion matriz de zoom.
    a = b;
end

b = uint8(b);
end

% Función: racional_zoom
% *****

% ENTRADA:
% v - Vector fila al que se le va a aplicar el algoritmo.
```



```
% n - Longitud del vector v.

% SALIDA:
% f - Vector fila con valores interpolados.

function [f] = racional_zoom(v, n)

f = zeros(1, 2*n);
f(1:2:2*n-1) = v(1:n);

% Obtencion de la mascara ENO.
[m1] = getMask(orden/2, orden/2);
[m3] = getMask(orden/2-1, orden/2+1);
[m2] = getMask(orden/2+1, orden/2);

% Prediccion del primer elemento.
f(2) = m1(1)*v(1) + m1(2)*v(2) + m1(3)*v(3) + m1(4)*v(4);

% Prediccion del segundo elemento.
elem2 = abs(-v(1) + 3*v(2) - 3*v(3) + v(4));
elem3 = abs(-v(2) + 3*v(3) - 3*v(4) + v(5));
if(elem2 < elem3)
    f(4) = m1(1)*v(1) + m1(2)*v(2) + m1(3)*v(3) + m1(4)*v(4);
else
    f(4) = m2(1)*v(2) + m2(2)*v(3) + m2(3)*v(4) + m2(4)*v(5);
end
% Prediccion valores intermedios
epsilon = 10^(-6);
for j=6:2:2*n-4
    elem1 = (v(j/2+1)-v(j/2+2));
    elem2 = (v(j/2-1) - v(j/2));

    %pesos
    a = 1/2;
    w0 = (1 + a*(elem1^2))/(2+a*(elem2^2+elem1^2));
    w1 = (1 + a*(elem2^2))/(2+a*(elem2^2+elem1^2));
    f(j) = w0*v(j/2) + w1*v(j/2+1);
end

% Prediccion del ultimo elemento.
f(2*n-2) = m2(1)*v(n) + m2(2)*v(n-1) + m2(3)*v(n-2) + m2(4)*v(n-3);

end
```

```
%*****
% Interpolacion PPH
%*****

%Función: int_pph
%*****

%ENTRADA:
% niv - Niveles de zoom.
% a - Imagen a interpolar.

%SALIDA:
% b - Imagen interpolada.

function [b]=int_pph(niv, a)

    [f, c] = size(a);

    % Bulce para los niveles de zoom.
    for k=1:niv

        % Filas y columnas para la imagen interpolada.
        nf = 2*f;
        nc = 2*c;

        % Creamos la nueva matriz.
        b = zeros(nf, nc);
        b(1:2:nf, 1:2:nc) = a(1:f, 1:c);

        % Algoritmo para la prediccion de las columnas.
        for j=1:2:nc
            b(1:nf, j) = pphzoom2(b(1:2:nf, j), f)';
        end

        %Prediccion de las filas.
        for i=1:nf
            b(i, 1:nc) = pphzoom2(b(i, 1:2:nc), c);
        end

        % Acutalizaciones de las variables.
        f = nf;
        c = nc;

        % Inicializacion matriz de zoom.
        a = b;
    end

    b = uint8(b);
end

%Función: int_pph
%*****

% ENTRADA:
```

```
% v - Vector fila al que se le va a aplicar el algoritmo.
% n - Longitud del vector v.

% SALIDA:
% f - Vector fila con valores interpolados.

function f = pphzoom2(v, n)

% Creamos el vector f, vector de salida de la funcion.
f = zeros(1, 2*n);
f(1:2:2*n-1) = v(1:n);

% Calculo de las mascaras.
[m2] = getMask(4/2-1, 4/2+1);

% Prediccion del primer elemento.
f(2) = m2(1)*v(1) + m2(2)*v(2) + m2(3)*v(3) + m2(4)*v(4);

% Prediccion de los valores intermedios.
for j=4:2:2*n-4
    elem1 = v(j/2-1) - 2*v(j/2) + v(j/2+1);
    elem2 = v(j/2) - 2*v(j/2+1) + v(j/2+2);
    prod = elem1*elem2;
    aux = (v(j/2) + v(j/2+1))/2;
    if(prod > 0)
        f(j) = aux - prod/(4*(elem1+elem2));
    else
        f(j) = aux;
    end
end

% Prediccion del ultimo elemento.
f(2*n-2) = m2(1)*v(n) + m2(2)*v(n-1) + m2(3)*v(n-2) + m2(4)*v(n-3);
end
```

```
%*****  
% Método para la obtención de las mascararas  
%*****  
  
%Función: getMask  
%*****  
% Calcula las máscaras del esquema de subdivisión basado en Lagrange de  
% orden n.  
%ENTRADA:  
% l - Número de puntos a la izquierda.  
% r - Número de puntos a la derecha.  
  
%SALIDA:  
% mas - Vector que contiene los valores de la mascara.  
% num - Vector que contiene los numeradores del polinomio de la mascara.  
% den - Vector que contiene los denominadores del polinomio de la mascara.  
  
function [mas,numer,deno]=getMask(l, r)  
  
alfas=-(2*l-1):2:(2*r-1);  
n=l+r;  
for i=0:(n-1)  
    prod=1;  
    prod2=1;  
    for j=0:(n-1)  
        if(j~=i)  
            prod=prod*alfas(j+1);  
            prod2=prod2*(alfas(i+1)-alfas(j+1));  
        end  
    end  
    numer(i+1)=(-1)^(n-1)*prod;  
    deno(i+1)=prod2;  
    mas(i+1)=numer(i+1)/deno(i+1);  
  
end  
end
```

Bibliografía

- [1] Olshan, AF, Bunin, *Epidemiology of Neuroblastoma*. 2000.
- [2] Matthias Schell, Christophe Bergeron, *Neuroblastoma*. Centre Léon Bérard, Service de Pédiatrie, 69373 Lyon cedex 08, France, 2003.
- [3] Carmen Gloria Rosti3n, Lorena Jáuregui, Valentina Broussain, Karol Gac y Angélica Paulos *Neuroblastoma : Forma de presentaci3n y probabilidad de resecci3n quirúrgica*. Departamento de Pediatría y Cirugía Infantil, Facultad de Medicina, Universidad de Chile, 2005.
- [4] Francisco José Ruiz Díaz, *Zoom de imágenes Digitales usando Esquemas de Subdivisi3n*. Tesis Doctoral, Universitat de València, 2008.
- [5] José Prades Nebot, *Interpolaci3n de imágenes Mediante Algoritmos No Lineales*. Tesis Doctoral, Escuela técnica Superior de Ingenieros de Telecomunicaci3n, Universitat Politècnica València.
- [6] F. Aràndiga and A. Belda, *Communications in Nonlinear Science and Numerical Simulation*. 2003.
- [7] F. Aràndiga, A. Belda and P. Mulet, *Point-value WENO multiresolution applications to stable image compression*. Journal of Scientific Computing, 2010.
- [8] J.Baccou and J. Liandrat, *Position dependent Lagrange interpolating multi-resolutions*. Int. J. Wavelets Multiresolut. Inf. Proc., 2007.
- [9] X.-D- Liu, S Osher and T. Chan, *Weighted essentially non-oscillatory schemes*. Journal of Computational Physics, 1994.
- [10] T. Alan Keahey, Edward L. Robertson, *Techniques for Nonlinear Magnification Transformations*. Indiana University, 1996.
- [11] Manuel Doblás Exposito, *Técnicas interpolatorias no lineales y aplicaciones*. Tesis Doctoral. Universitat de València, 2010.

- [12] Sonia Gandhi *Evo interpolation for image compression*. Dwarkadas J. Sanghvi College of Engineering, Mumbai, 2002.
- [13] M.G. Forero, W.J. Sarmiento, G.M. Díaz, *Aplicación para la lectura y conversión de formatos DICOM no estándar y génesis*. Memorias II Congreso Latinoamericano de Ingeniería Biomédica, La Habana, Mayo de 2001.
- [14] Chin-Chen Chang, Yung-Chen Chou, Yuan-Hui Yu, Kai-Jung Shih, *An image zooming technique based on vector quantization approximation*. Taiwan, 2004.
- [15] Miguel Gálvez Rodríguez, *Compresión y zoom de datos geológicos utilizando algoritmos de multirresolución*. Julio 2009.
- [16] David del Río Medina , Carlos Bocanegra Sánchez, David Santo Orcero, *La cabecera del estándar DICOM*. Revista esalud.com, Universidad de Málaga y Consejería de Innovación de la Junta de Andalucía, 2008.
- [17] S. Carrato and G. Ramponi, *Interpolation of the DC Component of Coded Images Using a Rational Filter* Proc. Fourth IEE Internal Conference on Image Processing, Santa Barbara (CA), 1997.