

TESIS DOCTORAL

Técnicas interpolatorias no lineales y aplicaciones

Manuel Doblas Expósito

Director: Francesc Aràndiga Llaudes

Universitat de València
Valencia, 2010.

TÉCNICAS INTERPOLATORIAS NO LINEALES Y APLICACIONES

Memoria presentada por Manuel Doblas Expósito, Licenciado en Ciencias Matemáticas; realizada en el Departamento de Matemática Aplicada de la Universitat de València bajo la Dirección de Francesc Aràndiga Llaudes, Profesor del mencionado Departamento, con el objetivo de aspirar al Grado de Doctor en Matemáticas.

Valencia, 4 de mayo de 2010

Francesc Aràndiga Llaudes
Director de la Memoria

Manuel Doblas Expósito
Aspirante al Grado de Doctor

DEPARTAMENTO DE MATEMÁTICA APLICADA
FACULTAD DE MATEMÁTICAS
UNIVERSITAT DE VALÈNCIA

Agradecimientos

En primer lugar, debo dar las gracias a Paco, ya que sin su dirección, ayuda y apoyo desde un primer momento no hubiera sido posible la finalización de éste trabajo.

Quiero expresar mi agradecimiento a todos los miembros del departamento de Matemática Aplicada, quienes siempre que he necesitado su ayuda no han dudado en ofrecérmela.

A mis compañeros becarios: Ana, Toni y Anna, por los momentos compartidos.

Gracias a mis suegros, por animarme y mostrar su ilusión por que acabara lo que en su día empecé.

A mis padres quiero agradecerles el esfuerzo realizado, sin el cual habría sido imposible el empezar este proyecto.

Y para finalizar, gracias a Marta, por todo.

Valencia, 2010

Manuel

Índice general

Introducción	XI
1. Interpolación de valores puntuales	1
1.1. Conceptos previos	1
1.2. Interpolación ENO	6
1.3. Interpolación ENO Subcell Resolution	8
1.4. Interpolación WENO	10
1.5. Interpolación racional	13
1.6. Modificaciones de la interpolación racional	17
1.6.1. Interpolación racional de orden 4	17
1.6.2. Extrapolación racional ponderada	21
1.6.3. Interpolación - Extrapolación racional	23
1.7. Ejemplo numérico	25
1.8. Interpolación 2D para valores puntuales	29
1.9. Interpolación no lineal en 2D para valores puntuales	31
1.9.1. <i>Stencils</i> cuadrados	32
1.9.2. <i>Stencils</i> triangulares	33
1.9.3. Implementación de una interpolación 2D	34
1.10. Resultados numéricos	35
2. Interpolación para medias en celda	39
2.1. Conceptos previos	39
2.2. Interpolación racional para medias en celda	45
2.3. Ejemplo numérico	49
2.4. Interpolación 2D para medias en celda	52
2.5. Interpolación ENO 2D para medias en celda	55

2.6. Resultados numéricos	61
2.6.1. Ampliación de una imagen geométrica	62
2.6.2. Ampliación de una imagen real	64
3. Multirresolución de Harten	67
3.1. Introducción	67
3.2. Multirresolución de Harten para valores puntuales	73
3.2.1. Multirresolución en $[0, 1]$	74
3.2.2. Multirresolución en $[0, 1] \times [0, 1]$	76
3.3. Modificación del algoritmo de control del error	81
3.4. Resultados numéricos	83
3.4.1. Imagen geométrica	84
3.4.2. Imagen real	84
3.4.3. Modificación del algoritmo de control del error	91
3.5. Multirresolución de Harten para medias en celda	94
3.5.1. Multirresolución en $[0, 1]$	95
3.5.2. Multirresolución en $[0, 1] \times [0, 1]$	99
3.6. Resultados numéricos	103
3.6.1. Imagen geométrica	104
3.6.2. Imagen real	108
4. Detección de contornos	113
4.1. Introducción	113
4.2. Detección de contorno mediante filtros	115
4.2.1. Detector de contornos de Roberts	115
4.2.2. Detector de contornos de Sobel	116
4.2.3. Detector de contornos Prewitt	117
4.3. Detector de contornos de Canny	117
4.4. Resultados numéricos	119
4.5. Detección utilizando la selección de <i>stencil</i>	123
4.6. Un método nuevo para la detección de contornos	124
4.7. Detección de contornos continuos	131
5. Multirresolución 2D no separable para valores puntuales	137
5.1. Introducción	137
5.2. Multirresolución de Harten dependiente	139
5.2.1. Caso uno dimensional	139
5.2.2. Predicción 2D mapa-dependiente	147
5.3. Construcción del mapa	155

5.3.1. Construcción de $map^{k_{max}}$	156
5.3.2. Mapas de contornos de distintos niveles	157
5.4. Codificación de la información comprimida	158
5.5. Aplicaciones y comparaciones numéricas	158
5.5.1. Imagen geométrica	159
5.5.2. Imagen real	160
5.6. Conclusiones	165
6. Multirresolución 2D no separable para medias en celda	167
6.1. Introducción	167
6.2. Predicción en contornos	169
6.2.1. Caso 1: Imagen constante a trozos	171
6.2.2. Caso 2: Imagen polinómica a trozos	174
6.2.3. Consistencia de la predicción	179
6.3. Detección de contornos	182
6.4. Interpolación adaptada	182
6.5. Resultados numéricos	183
6.5.1. Interpolación de imágenes	184
6.5.2. Compresión de imágenes	191
6.6. Conclusiones	198
7. Conclusiones y perspectivas	199

Introducción

El trabajo presentado en esta memoria está dedicado al desarrollo de técnicas interpolatorias para la ampliación de imágenes (zooming) y para la compresión de imágenes.

Un problema habitual en teoría de aproximación consiste en reconstruir una función a partir de un conjunto discreto de datos. Este problema puede abordarse desde el punto de vista de valores puntuales o de medias en celda, dependiendo de si consideramos los datos discretos como los valores que toma una función en un conjunto de puntos o las medias en unos intervalos, respectivamente. De este modo la función puede ser aproximada por otra función que posea los mismos valores en el conjunto de puntos, o las mismas medias en los intervalos.

Las técnicas interpolatorias lineales, independientes de los datos, como la interpolación polinómica de Lagrange, son frecuentemente empleadas para reconstruir funciones. Sin embargo, el orden de la aproximación se ve reducido cuando el conjunto de puntos empleados, *stencil*, para construir la función interpolante contiene una singularidad de la función. El aumentar el orden de la técnica interpolatoria no resuelve el problema, pues en este caso se ven afectados un mayor número de *stencils* por la singularidad, aumentando por tanto la zona donde la calidad de la aproximación no es óptima.

Una correcta selección de los nodos adecuados para construir el interpolante, de modo que el *stencil* no contenga ninguna singularidad, produce una mejora en la interpolación. En [31], Harten et al. introducen la técnica interpolatoria no lineal ENO (*Essentially Non-Oscillatory*, Esencialmente no oscilatoria), en la cual el

polinomio interpolador se construye, siempre y cuando es posible, tomando información de las zonas donde la función es suave. De este modo, si las singularidades de la función están lo suficientemente aisladas, se consigue reducir la zona donde la aproximación se ve degradada al intervalo que la contiene. Si se posee información acerca de la localización de la singularidad, la técnica ENO Subcell Resolution (Resolución subcelda) [28] permite reducir la región de pérdida de exactitud a un entorno alrededor de la singularidad.

La técnica interpolatoria WENO (*Weighted ENO*, ENO Ponderado) introducida por Liu et al. [34] constituye otra mejora de la técnica ENO. Ésta consiste en construir el interpolante a partir de combinaciones convexas de todas las aproximaciones obtenidas a partir de *stencils* que contienen el intervalo a interpolar, de modo que en dicha combinación se priman las aproximaciones de los *stencils* de zonas suaves y si la función es suave en todos los *stencils* se obtiene una aproximación de orden óptimo.

La importancia de obtener una técnica interpolatoria eficiente está estrechamente relacionada con la compresión de imágenes. La multirresolución de Harten [29, 30] emplea un operador *predicción* que permite transitar entre un nivel de resolución inferior y el inmediatamente superior. Este operador predicción no es nada más que la aplicación de un esquema interpolatorio al nivel de resolución más grosero. La posibilidad de elegir un operador predicción no lineal permite adaptarse a las características de los datos. Así, un esquema de compresión basado en multirresolución puede describirse del siguiente modo:

$$\begin{aligned} f^L &\longrightarrow Mf^L = (f^0, d^1, \dots, d^L) \\ &\longrightarrow \hat{M}f^L = Q_\epsilon Mf^L = (\hat{f}^0, \hat{d}^1, \dots, \hat{d}^L), \end{aligned}$$

donde f^L representa los datos en un nivel de resolución dado, y Mf^L su transformada por el esquema de multirresolución. Esta última está compuesta por los datos en un nivel de resolución inferior, f^0 , más una sucesión de *detalles*, d^k , que nos permiten recuperar f^L . Posteriormente se aplica un operador de cuantización no reversible, Q_ϵ , codificando la información resultante. En el esquema de multirresolución de Harten, los coeficientes detalle, d^k , son producto de los errores cometidos por el operador predicción

al reconstruir f^{k+1} a partir de f^k . De este modo una mejora en el operador predicción, produce como resultado una disminución de las magnitudes de los detalles y, por tanto, una mayor compresión.

Si bien los métodos basados en la transformada wavelet constituyen herramientas eficaces en la compresión de imágenes [36], los resultados obtenidos no son del todo satisfactorios por dos razones: en primer lugar, emplean un proceso multiescala lineal y, por tanto, no se adaptan a las propiedades locales de la imagen (posición de los contornos) y, en segundo lugar, la aplicación tensorial de la transformada wavelet no resulta óptima para el tratamiento de perfiles no verticales y no horizontales.

Con el objetivo de mejorar los resultados obtenidos, se han desarrollado nuevos métodos numéricos que incorporan la geometría de la imagen como los Bandelets [40], Countourlets [25], etc.

Nuestro objetivo es desarrollar técnicas interpolatorias no lineales dos dimensionales que se adapten a la geometría de las imágenes.

Este trabajo está organizado del siguiente modo, el Capítulo 1 está dedicado al estudio de técnicas interpolatorias para valores puntuales. En primer lugar repasamos las técnicas clásicas uno dimensionales, y posteriormente, tras explicar la interpolación racional introducida por G. Ramponi [42], se proponen diversas modificaciones de ésta, estudiando la relación entre la interpolación racional y WENO. Del mismo modo se proponen diversos métodos para realizar una interpolación para valores puntuales directamente dos-dimensional. Para finalizar el capítulo se comparan los resultados obtenidos al ampliar una imagen de carácter geométrico.

El Capítulo 2 está dedicado a la interpolación para medias en celda. Se estudia como adaptar la interpolación racional a este contexto proponiendo dos modificaciones de ésta y posteriormente se estudian operadores interpolatorios dos dimensionales. Finalmente se aplican estas técnicas a la interpolación de imágenes.

El Capítulo 3 está dedicado a la multirresolución de Harten. Se comparan los resultados obtenidos a partir de las técnicas interpolatorias anteriores. Con el objetivo de asegurar la estabilidad de la multirresolución de Harten se aplica el algoritmo de control del

error [1, 11], proponiendo una modificación de este en el contexto de los valores puntuales [8] consistente en aplicar el operador predicción en dos pasos, el segundo de los cuales aprovecha la información actualizada por los detalles correspondientes.

Con el objetivo de realizar una interpolación adaptada a la geometría de la imagen, el Capítulo 4 está dedicado a la detección de contornos. En éste se proponen dos nuevos algoritmos de detección de contornos que se emplearan en los Capítulos 5 y 6.

En el Capítulo 5 presentamos un algoritmo multidireccional para la compresión de imágenes [5] el cual hace uso de la geometría de la imagen detectada en el nivel de resolución más fino para localizarlos en niveles de resolución inferiores y adaptar así la predicción. Este algoritmo supone que trabajamos en el contexto de los valores puntuales.

Del mismo modo, en el Capítulo 6 se propone un operador predicción adaptado a los contornos, pero en este caso para medias en celda. En este se obtiene la información relativa a la geometría de la imagen a partir de una detección de contornos en cada nivel, y posteriormente se proponen dos posibilidades para interpolar en las celdas afectadas por la presencia de un perfil [6, 22, 39].

1

Interpolación de valores puntuales

1.1

Conceptos previos

El problema de la interpolación consiste en: dados los valores que toma una función $f(x)$ desconocida en una discretización dada, obtener una función de modo que coincida con la función $f(x)$ en los nodos de la discretización anterior. Esta nueva función recibe el nombre de interpolante.

La interpolación polinómica de Lagrange nos permite, conocidos los valores que toma la función f en $r + 1$ nodos, x_0, \dots, x_r , con $x_i \neq x_j$ si $i \neq j$, construir un polinomio de grado r , $q(x)$, interpolando a $f(x)$ en los nodos anteriores, i.e. $q(x_i) = f(x_i)$, $\forall i = 0, \dots, r$. La

forma de Newton del polinomio interpolador nos ofrece una forma sencilla de calcular dicho polinomio, cuya expresión viene dada por:

$$q(x) = f[x_0] + f[x_0, x_1](x - x_0) + f[x_0, x_1, x_2](x - x_0)(x - x_1) + \dots \\ + f[x_0, x_1, \dots, x_r](x - x_0) \dots (x - x_{r-1}), \quad (1.1)$$

donde $f[x_i, \dots, x_{i+n}]$ es la n -ésima diferencia dividida de $f(x)$, calculada en los nodos x_i, \dots, x_{i+n} , la cual, si $x_{i+k} \neq x_{i+l}$ con $k \neq l$, $k, l = 0, \dots, n$, se define como:

$$f[x_i, \dots, x_{i+n}] = \frac{f[x_{i+1}, \dots, x_{i+n}] - f[x_i, \dots, x_{i+n-1}]}{x_{i+n} - x_i},$$

donde $f[x_{i+k}] = f(x_{i+k})$, $k = 0, \dots, n$. El conjunto de nodos empleado para la construcción del polinomio interpolador $q(x)$, $\mathcal{S} = \{x_0, x_1, \dots, x_r\}$, recibe el nombre de *stencil*.

Se puede demostrar que se verifica la relación siguiente

$$f(x) = q(x) + f[x_0, \dots, x_r, x] \prod_{i=0}^r (x - x_i) = \\ = q(x) + f[\mathcal{S}, x] \prod_{i=0}^r (x - x_i). \quad (1.2)$$

Si $f(x)$ es lo suficientemente diferenciable (diferenciable $r + 1$ veces), entonces

$$f[x_0, \dots, x_r, x] = \frac{f^{(r+1)}(\xi_x)}{(r+1)!}, \quad (1.3)$$

donde ξ_x pertenece a la envoltura convexa de $\{x_0, x_1, \dots, x_r\}$.

La interpolación polinómica a trozos, nos permite obtener una buena aproximación a $f(x)$ en un dominio amplio, veamos en que consiste esta técnica. Sea $X = \{x_i\}_{i=0}^N$ una partición de nodos igualmente espaciados del intervalo $I = [0, 1]$, i.e. $x_i = ih$, $Nh = 1$, y sean $\{f_i\}_{i=0}^N$ los valores que toma $f(x)$ en los nodos anteriores, i.e. $f_i = f(x_i)$, $i = 0, \dots, N$. De este modo definimos en cada intervalo $I_i = [x_{i-1}, x_i]$, un polinomio $p_i(x)$ verificando $q_i(x_{i-1}) = f_{i-1}$ y $q_i(x_i) = f_i$, para todo $i = 1, \dots, N$. Definimos el interpolante de $f(x)$ sobre I como

$$\mathcal{I}(x) = q_i(x), x \in I_i = [x_{i-1}, x_i], i = 1, \dots, N.$$

Así, si construimos un interpolante polinómico a trozos, y $f(x)$ es lo suficientemente suave, haciendo uso de las relaciones (1.2) y (1.3) obtenemos

$$f(x) - p_i(x) = \frac{f^{(r+1)}(\xi_x)}{(r+1)!} \prod_{k=0}^r (x - x_{i-1-s+k}), \quad (1.4)$$

con $s \in \{0, \dots, r-1\}$ y $x \in I_i$, de donde podemos deducir que

$$f(x) = \mathcal{I}(x) + O(h^{r+1}), \quad \forall x \in [0, 1]. \quad (1.5)$$

Notar que la fórmula (1.4) nos asegura que la aproximación obtenida es exacta para polinomios de grado menor o igual que r , de este modo diremos que la técnica interpolatoria empleada es de orden $r+1$.

En el caso de que $f(x)$ posea una discontinuidad de salto en el intervalo I_i , podemos comprobar que

$$f[x_{i-1}, x_i] = \frac{f(x_i) - f(x_{i-1}))}{h} = \frac{O([f])}{h},$$

donde $O([f])$ es la magnitud del salto de $f(x)$ en el intervalo I_i . Análogamente, se puede probar si tomamos un conjunto de $s+1$ nodos ($s > 1$) conteniendo los nodos x_{i-1}, x_i entonces $f[x_l, \dots, x_{l+s}] = O([f])/h^s$. Así se puede deducir que si empleamos un *stencil* \mathcal{S} que cruce la discontinuidad, es decir $x_{i-1}, x_i \in \mathcal{S}$, el error cometido viene dado por

$$f(x) = q_i(x) + O([f]), \quad x \in I_i, \quad (1.6)$$

pues empleando (1.2) obtenemos

$$\begin{aligned} f(x) &= q_i(x) + f[x_l, \dots, x_{l+s}, x] \prod_{i=l}^{l+s} (x - x_i) = \\ &= q_i(x) + \frac{O([f])}{h^{s+1}} O(h^{s+1}) = q_i(x) + O([f]), \quad \forall x \in I_i. \end{aligned}$$

De este modo el error cometido en caso de que el *stencil* empleado cruce una discontinuidad es de la magnitud de la discontinuidad.

El hecho de que el *stencil* cruce una discontinuidad de $f^{(p)}$, $p \geq 1$, también produce un aumento del error de interpolación. Supongamos que $f^{(p)}$, $p \geq 1$, es discontinua en $x_d \in I_i$, entonces cualquier diferencia dividida con $s+1$ nodos conteniendo a x_{i-1} y x_i

verifica

$$f[x_l, \dots, x_{l+s}] = \begin{cases} O([f^{(p)}])/h^{s-p} & \text{si } s > p \\ O(\|f^{(s)}\|) & \text{en otro caso,} \end{cases} \quad (1.7)$$

donde $\|\cdot\|$ es la norma supremo de $f^{(s)}$ en la envoltura convexa de los nodos del *stencil* empleado, y $[f^{(p)}] = f^{(p)}(x_d^+) - f^{(p)}(x_d^-)$.

Si empleamos la relación anterior, podemos probar que si $q_l(x)$ es un polinomio de grado r , que interpola a f en el intervalo I_l , entonces si el *stencil* empleado para la construcción de $q_l(x)$, S_l , cruza una discontinuidad de $f^{(p)}$, el error cometido viene dado por

$$f(x) = q_l(x) + O([f^{(p)}])h^p, \quad \text{con } p \leq r, \quad \forall x \in I_l. \quad (1.8)$$

Notar que si $\mathcal{I}(x)$ es un interpolante polinómico a trozos de orden $r+1$ y f es lo suficientemente suave, entonces se verifica la siguiente relación entre la derivada m -ésima de $\mathcal{I}(x)$ y la derivada m -ésima de $f(x)$:

$$\frac{d^m}{dx^m} \mathcal{I}(x) = \frac{d^m}{dx^m} f(x) + O(h^{r+1-m}), \quad 0 \leq m \leq r, \quad (1.9)$$

pues derivando m veces (1.2), tenemos:

$$\begin{aligned} \frac{d^m}{dx^m} f(x) &= \frac{d^m}{dx^m} q(x) + \left(\frac{d^m}{dx^m} f[x_0, \dots, x_r, x] \right) \prod_{i=0}^r (x - x_i) + \\ &+ f[x_0, \dots, x_r, x] \frac{d^m}{dx^m} \left(\prod_{i=0}^{r+1} (x - x_i) \right) = \\ &= \frac{d^m}{dx^m} q(x) + O(h^{r+1}) + O(h^{r+1-m}). \end{aligned}$$

Así, si optamos por emplear una interpolación polinómica a trozos de grado r , obtenemos una aproximación de orden $O(h^{r+1})$ en aquellos intervalos donde el *stencil* no cruce ninguna discontinuidad de la función o sus derivadas, siendo ésta a la vez la mejor aproximación a la que podemos aspirar con dicha técnica interpolatoria. En caso de que el *stencil* empleado en algún intervalo contenga alguna singularidad, entonces la calidad de la aproximación en dicho intervalo se verá afectada. Si queremos mejorar

la aproximación obtenida en aquellos intervalos que no posean singularidades, entonces deberemos emplear para la construcción del polinomio interpolador *stencils* libres de discontinuidades de la función y de sus derivadas.

La interpolación lineal consiste en emplear en todos los intervalos I_l el mismo *stencil*. Habitualmente se emplean *stencils* centrados en el intervalo I_l , de este modo dichos *stencils* están formados por un número par de nodos, $r + 1 = 2s$, donde el *stencil* empleado para el intervalo I_l , es $\mathcal{S}_l = \{x_{l-s}, \dots, x_{l+s-1}\}$. Notar que en caso de emplear interpolación lineal con $2s$ nodos, si el intervalo I_i contiene una singularidad, entonces existen $2s - 1$ *stencils* conteniendo los nodos x_{i-1} , x_i , y por tanto $2s - 1$ intervalos donde la aproximación obtenida no es óptima. En esta situación, el aumento del orden de interpolación no produce una mejora en la aproximación obtenida como cabría esperar, ya que dicho aumento implica un mayor número de intervalos afectados por la singularidad. La figura 1.1 nos muestra el número de *stencils* afectados por una discontinuidad en el intervalo I_j , cuando dichos *stencils* están formados por 4 nodos, i.e., $s = 2$.

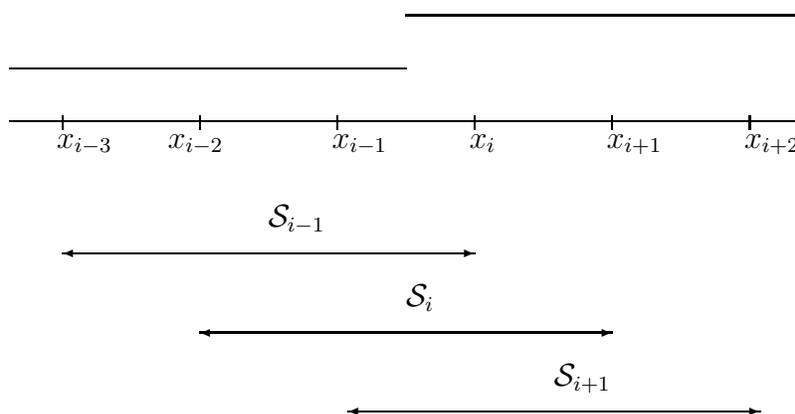


Figura 1.1: Stencils afectados por discontinuidad en I_i , $s = 2$

Las técnicas no lineales nos permiten realizar una selección del *stencil* más adecuado para interpolar en un determinado in-

tervalo. De modo que podemos conseguir una mejora en aquellos intervalos en los que, pese a no contener ninguna singularidad, la aproximación obtenida no es óptima a causa de que el *stencil* empleado con la interpolación lineal cruza alguna singularidad.

A. Harten introduce en [31] una técnica interpolatoria polinómica a trozos no lineal que recibe el nombre de interpolación ENO (*Essentially Non-Oscillatory*). La intención original de la interpolación ENO era evitar el fenómeno de Gibbs que se producía al aproximar funciones con discontinuidades de salto mediante interpolación centrada. El interpolante ENO intenta construir los polinomios usando información de la parte suave de la función. Ésto es posible siempre y cuando las singularidades estén lo suficientemente aisladas, excepto para aquellos intervalos que contengan alguna singularidad.

1.2

Interpolación ENO

A diferencia de la interpolación lineal, la técnica interpolatoria ENO realiza una selección del *stencil*, de modo que se intenta tomar para cada intervalo un *stencil* que no cruce ninguna singularidad, lo cual es posible si las discontinuidades están lo suficientemente separadas las unas de las otras. Si empleamos una interpolación polinómica a trozos, de modo que el *stencil* para el intervalo $I_i = [x_{i-1}, x_i]$ esté constituido por $r + 1$ nodos, la técnica ENO trata de escoger el *stencil* más suave de entre aquellos que contienen a x_{i-1} y x_i . Como medida de suavidad de los *stencils* se emplean las diferencias divididas de los nodos que lo forman. En [31], los autores proponen dos algoritmos de selección de *stencil*:

Algoritmo. 1.1 (Selección jerárquica del *stencil*).

for $i = 1, \dots, J$

$s_0 = i$

for $l = 0, \dots, r - 2$

if $|f[x_{s_l-2}, \dots, x_{s_l+l}]| < |f[x_{s_l-1}, \dots, x_{s_l+l+1}]|$ **then** $s_{l+1} = s_l - 1$

```

end
     $s_i = s_{r-1}$ 
end

```

Algoritmo. 1.2 (Selección no jerárquica del *stencil*).

for $i = 1, \dots, J$, *elegir* s_i *que verifique*

$$|f[x_{s_i-1}, \dots, x_{s_i+r-1}]| = \min \{ |f[x_l-1, \dots, x_{l+r-1}]|, i-r+1 \leq l \leq i \}$$

Denotaremos al *stencil* ENO del intervalo I_i con $r+1$ nodos de la siguiente forma:

$$\mathcal{S}_i^{ENO} = \{x_{s_i-1}, x_{s_i}, \dots, x_{s_i+r-1}\}.$$

Notar que, tanto si empleamos el algoritmo de selección jerárquico como el no jerárquico, los nodos x_{i-1} , x_i pertenecen a \mathcal{S}_i^{ENO} . Además, en caso de que f tenga una discontinuidad de salto en $x_d \in I_i$, y sean \mathcal{S} un *stencil* que no cruza dicha discontinuidad y \mathcal{S}^* un *stencil* conteniendo los nodos x_{i-1} y x_i , ambos *stencils* con $s+1$ nodos, entonces empleando (1.7) tenemos:

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

Si la discontinuidad pertenece a la primera derivada, empleando la misma notación obtenemos entonces

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

De este modo, en el caso de que f solamente presentara la discontinuidad en el intervalo I_i , puesto que los algoritmos de selección anteriores se alejan de las discontinuidades, entonces los polinomios interpoladores verificarán

$$f(x) = q_l(x) + O(h^{r+1}), \quad x \in [x_{l-1}, x_l], \quad l \leq i-1, l \geq i+1, \quad (1.10)$$

obteniendo una aproximación óptima en todos los intervalos excepto en el intervalo I_i que contiene la singularidad.

Así, en presencia de discontinuidades de la función y de su primera derivada, y siempre y cuando éstas estén lo suficientemente aisladas, al emplear interpolación ENO se consigue que las zonas donde la aproximación queda afectada por las singularidades quede reducida exclusivamente a los intervalos que contienen dichas singularidades, independientemente del algoritmo de selección de *stencil* empleado. Además, en el caso especial de que la singularidad x_d coincida con un nodo de interpolación, i.e. $x_d = x_i$, se puede obtener máxima precisión en todos los intervalos siempre y cuando el *stencil* sea elegido del modo siguiente:

$$\{x_{i+1}\} \cap \mathcal{S}_l^{ENO} = \emptyset, \quad l \leq i \quad \{x_{i-1}\} \cap \mathcal{S}_l^{ENO} = \emptyset, \quad l \geq i+1. \quad (1.11)$$

Por tanto $q_l(x) = f(x) + O(h^{r+1})$, $x \in I_l$, $\forall l$, ya que ningún *stencil* cruza la discontinuidad, consiguiendo una aproximación óptima en todo el intervalo $[0, 1]$.

En caso de que la singularidad no coincida con un nodo, i.e. $x_d \neq x_i$, $\forall i$, podemos mejorar la aproximación obtenida si conocemos la localización de la singularidad, ya que en este caso podemos extender los polinomios interpoladores correspondientes a los intervalos vecinos hasta la singularidad, de modo que la relación $\mathcal{I}(x) = f(x) + O(h^{r+1})$ sería válida en todo el intervalo que contiene la singularidad, excepto en una pequeña banda alrededor de ella. Ésta es la idea clave en la interpolación Subcell Resolution.

1.3

Interpolación ENO Subcell Resolution

Sea $f(x)$ una función continua, de modo que posee una esquina en $x_d \in I_i = [x_{i-1}, x_i]$, es decir, supongamos

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

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

Denotemos por $q_{i-1}(x)$ y $q_{i+1}(x)$ a los polinomios interpoladores de f obtenidos mediante interpolación ENO en los intervalos I_{i-1} , I_{i+1} respectivamente. Así, si empleamos (1.10) tenemos:

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

Definimos la función:

$$G_i(x) = q_{i+1}(x) - q_{i-1}(x). \quad (1.12)$$

Si tomamos desarrollos de Taylor alrededor de x_d obtenemos:

$$\begin{aligned} q_{i+1}(x) &= f(x_d^+) + f'(x_d^+)(x - x_d) + O((x - x_d)^2) + O(h^{r+1}), \\ q_{i-1}(x) &= f(x_d^-) + f'(x_d^-)(x - x_d) + O((x - x_d)^2) + O(h^{r+1}), \end{aligned} \quad (1.13)$$

y de aquí:

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

Si tomamos h lo suficientemente pequeño, se verifica $G_i(x) \approx [f']_{x_d}(x - x_d)$, 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:

$$G_i(x_i)G_i(x_{i-1}) = a(a-1)[f']_{x_d}^2 h^2 + O(h^2) \approx a(a-1)[f']_{x_d}^2 h^2 < 0. \quad (1.14)$$

Sea ξ esta raíz, en este caso podemos probar que $|\xi - x_d| = O(h^{r+1})$, para ello definamos $\tilde{G}(x) = f_r(x) - f_l(x)$, 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 x_d :

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

donde tomando h lo suficientemente pequeño 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} , i.e.:

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

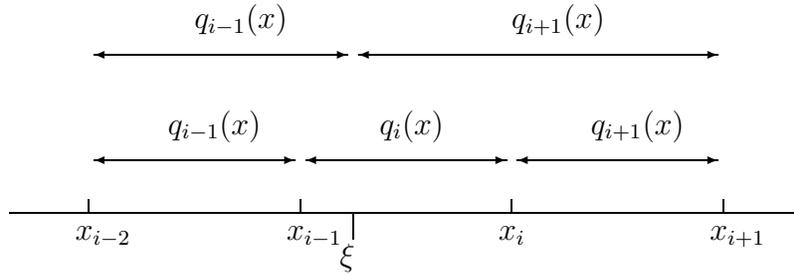


Figura 1.2: Polinomio interpolador ENO-SR.

Así definido, el error de interpolación es

$$\mathcal{I}(x) = f(x) + O(h^{r+1}), \quad (1.15)$$

excepto en una pequeña banda alrededor de x_d .

1.4

Interpolación WENO

Como hemos visto en la sección anterior, la interpolación ENO nos permite seleccionar el *stencil* más adecuado en cada intervalo, pudiendo obtener de este modo, si las discontinuidades están lo suficientemente aisladas, un orden de aproximación $r + 1$ en aquellos intervalos que no contienen ninguna singularidad. La técnica interpolatoria WENO, introducida por Liu et al. en el contexto de las leyes de conservación (ver [34] y [15]), se presenta como una mejora de la interpolación ENO. La idea fundamental de esta técnica interpolatoria consiste en emplear la información dada por los $2r$ nodos implicados en el proceso de selección de *stencil*. Así, la interpolación WENO emplea una combinación convexa de los polinomios correspondientes a todos los *stencils* posibles para un intervalo dado. Denotemos por \mathcal{S}_{i+k} , $k = 0, \dots, r - 1$, los r *stencils* conteniendo al intervalo I_i ,

$$\mathcal{S}_{i+k} = \{x_{i+k-r}, \dots, x_{i+k}\}, \quad k = 0, \dots, r - 1, \quad (1.16)$$

y sea $p_{i+k}(x)$ el polinomio construido a partir del *stencil* \mathcal{S}_{i+k} . Definimos el polinomio interpolador WENO en el intervalo I_i como

$$p_i^{WENO}(x) = \sum_{k=0}^{r-1} \omega_k^i p_{i+k}(x), \quad (1.17)$$

donde

$$\omega_k^i \geq 0, \quad k = 0, \dots, r-1, \quad \sum_{k=0}^{r-1} \omega_k^i = 1. \quad (1.18)$$

Notar que para cada intervalo, el interpolante WENO toma información de $2r$ nodos. De este modo se pretende alcanzar una aproximación de orden $2r$ en aquellos intervalos donde la función sea suave. Veamos como podemos conseguir dicho orden de aproximación. Sea \hat{p}_i^{2r-1} la aproximación obtenida empleando el *stencil* formado por $2r$ nodos $\{x_{i-r}, \dots, x_{i+r-1}\}$, y sean \hat{p}_{i+k}^r las aproximaciones obtenidas con los *stencils* \mathcal{S}_{i+k} . Podemos expresar la aproximación de orden $2r$ como una combinación lineal de las r aproximaciones de orden $r+1$, i.e.:

$$\hat{p}_i^{2r-1} = \sum_{k=0}^{r-1} C_k^r \hat{p}_{i+k}^r, \quad (1.19)$$

con $C_k^r \geq 0, \forall k$ y $\sum_{k=0}^{r-1} C_k^r = 1$. Las constantes C_k^r se denominan pesos óptimos. La tabla 1.1 nos muestra los pesos óptimos para $r = 2, 3$, en [3, 4, 15] podemos encontrar los detalles sobre como obtener dichos pesos.

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

Tabla 1.1: Pesos óptimos para $r = 2, 3$.

Si definimos los pesos ω_k^i de modo que verifiquen la siguiente relación:

$$\omega_k^i = C_k^r + O(h^{r-1}), \quad k = 0, \dots, r-1, \quad (1.20)$$

se puede demostrar (ver [32, 15]) que la aproximación obtenida es de orden $2r$. Además, se desea que la contribución de los polinomios cuyo *stencil* cruce alguna singularidad sea lo menor posible, y por tanto sus pesos en (1.17) deben ser muy próximos a 0.

Con el objetivo de satisfacer las condiciones anteriores, se definen los pesos como sigue:

$$\omega_k^i = \frac{\alpha_k^i}{\sum_{s=0}^{r-1} \alpha_s^i}, \quad k = 0, \dots, r-1, \quad (1.21)$$

con

$$\alpha_k^i = \frac{C_k^r}{(\epsilon + IS_{i+k})^2}. \quad (1.22)$$

Notar que el valor ϵ introducido en (1.22) es una constante positiva para evitar que el denominador se anule, y IS_{i+k} es un indicador de suavidad de $f(x)$ en el *stencil* S_{i+k} .

Resulta evidente que así definidos, los pesos verifican (1.18) independientemente del indicador de suavidad utilizado. Para que la contribución de los polinomios que cruzan una singularidad sea prácticamente nula, es suficiente con que $IS_{i+k} = O(h^2)$ para aquellos *stencils* en los que $f(x)$ es suave, y $IS_{i+k} = O(1)$ en aquellos *stencils* que la función presenta alguna singularidad. Finalmente, se puede demostrar que si el indicador de suavidad verifica,

$$IS_{i+k} = D(1 + O(h^{r-1})), \quad k = 0, \dots, r-1, \quad (1.23)$$

se satisface la relación (1.20) (ver [3, 4, 15]), donde D es una constante no nula que puede depender de h pero no de k .

Liu et al. ([34]) proponen el siguiente indicador de suavidad basado en las diferencias divididas de $f(x)$ en los nodos de interpolación,

$$IS_{i+k} = \sum_{l=1}^r \left(\sum_{m=1}^l (\Delta^{r-l+1}[f_{i+k+m-r-1}])^2 \right) / l, \quad (1.24)$$

donde $\Delta^s[f_l]$ no es más que la diferencia dividida de f en los nodos $\{x_l, \dots, x_{l+s}\}$. Se puede ver que con este indicador de suavidad se alcanza el orden óptimo, 4, si $r = 2$, pero sin embargo cuando $r = 3$ se obtiene orden 5, siendo 6 el orden óptimo en este caso.

Jiang y Shu, definen en [32] un nuevo indicador de suavidad,

$$IS_{i+k} = \sum_{l=1}^r \int_{x_{i-1}}^{x_i} h^{2l-1} (p_{i+k}^{(l)}(x))^2 dx. \quad (1.25)$$

Este nuevo indicador hace posible alcanzar órdenes óptimos tanto para el caso $r = 2$ como $r = 3$. Las demostraciones de los órdenes de exactitud alcanzados con ambos métodos, para $r = 2, 3$, se pueden consultar en [3, 4, 15].

1.5

Interpolación racional

En esta sección se analizan los filtros racionales empleados por G. Ramponi et al. en el procesamiento de imágenes [19, 20, 21, 42]. También se estudia el error cometido al interpolar empleando dichos filtros, y se muestra su semejanza con la interpolación WENO. Para finalizar, se proponen diversas extensiones de la interpolación racional, analizando el error cometido por cada una de ellas.

De forma análoga a las secciones anteriores, consideraremos $X = \{x_i\}_{i=0}^N$, $x_i = ih$, $Nh = 1$, una partición de nodos igualmente espaciados del intervalo $[0, 1]$, y $f_i = f(x_i)$ los valores que toma $f(x)$ en los nodos de la partición.

S. Carrato y G. Ramponi proponen emplear como aproximación de $f(x_{i-1} + h/2) = f_{i-1/2}$ una media ponderada entre f_{i-1} y f_i , (ver [19]), i.e.,

$$\hat{f}_{i-1/2} = \omega_{i-1} f_{i-1} + \omega_i f_i, \quad \text{con } \omega_{i-1} + \omega_i = 1, \quad (1.26)$$

donde $\hat{f}_{i-1/2}$ es la aproximación obtenida.

Los valores empleados para los pesos pueden venir dados por las siguientes expresiones:

$$\begin{aligned} w_{i-1}^1 &= \frac{1 + \alpha(f_i - f_{i+1})^2}{2 + \alpha((f_{i-2} - f_{i-1})^2 + (f_i - f_{i+1})^2)}, \\ w_i^1 &= \frac{1 + \alpha(f_{i-2} - f_{i-1})^2}{2 + \alpha((f_{i-2} - f_{i-1})^2 + (f_i - f_{i+1})^2)}, \end{aligned} \quad (1.27)$$

$$\begin{aligned}
w_{i-1}^2 &= \frac{1 + \alpha(f_{i-1} - f_{i+1})^2}{2 + \alpha((f_{i-2} - f_i)^2 + (f_{i-1} - f_{i+1})^2)}, \\
w_i^2 &= \frac{1 + \alpha(f_{i-2} - f_i)^2}{2 + \alpha((f_{i-2} - f_i)^2 + (f_{i-1} - f_{i+1})^2)}, \quad (1.28)
\end{aligned}$$

$$\begin{aligned}
w_{i-1}^3 &= \frac{1 + \alpha((f_{i-1} - f_{i+1})^2 + (f_i - f_{i+1})^2)}{2 + \alpha(\sum_{k=0}^1((f_{i-2} - f_{i-k})^2 + (f_{i-k} - f_{i+1})^2))}, \\
w_i^3 &= \frac{1 + \alpha((f_{i-2} - f_i)^2 + (f_{i-2} - f_{i-1})^2)}{2 + \alpha(\sum_{k=0}^1((f_{i-2} - f_{i-k})^2 + (f_{i-k} - f_{i+1})^2))}, \quad (1.29)
\end{aligned}$$

donde α es un parámetro variable. Notar que en el caso de tomar $\alpha = 0$, la predicción coincide en los tres casos con el valor medio de f_{i-1} y f_i , es decir:

$$\hat{f}_{i-1/2} = \omega_{i-1}^k f_{i-1} + \omega_i^k f_i = \frac{1}{2} f_{i-1} + \frac{1}{2} f_i, \quad k = 1, 2, 3. \quad (1.30)$$

La predicción obtenida también coincide con el valor medio entre f_{i-1} y f_i cuando los datos que tenemos verifican una de las siguientes relaciones:

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

con $x \pm h \in [x_{i-2}, x_{i+1}]$, pues en estos casos, la función es *simétrica* respecto $x_{i-1/2}$ (ver figura 1.3), y se verifican las siguientes relaciones:

$$|f_{i-2} - f_{i-1}| = |f_i - f_{i+1}| \quad , \quad |f_{i-1} - f_{i+1}| = |f_{i-2} - f_i|, \quad (1.31)$$

las cuales implican que $w_{i-1}^k = w_i^k = \frac{1}{2}$, $k = 1, 2, 3$.

Si $f(x)$ no es simétrica respecto $x_{i-1/2}$ en $[x_{i-2}, x_{i+1}]$, el uso de (1.27) provoca que la predicción $\hat{f}_{i-1/2}$ tenga una mayor influencia de f_{i-1} si $(f_i - f_{i+1})^2 > (f_{i-2} - f_{i-1})^2$ y de f_i si la desigualdad se da en sentido contrario. En caso de emplear (1.28), la contribución de f_{i-1} dominará cuando $(f_{i-1} - f_{i+1})^2 > (f_{i-2} - f_i)^2$, mientras que el uso de (1.29) nos ofrece un comportamiento intermedio entre (1.27) y (1.28).

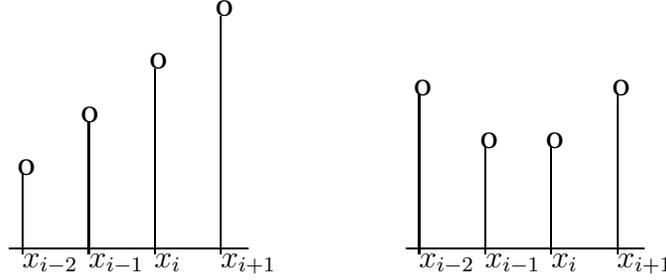


Figura 1.3: Datos simétricos respecto $x_{i-1/2}$.

Analicemos el error cometido al aproximar $\hat{f}_{i-1/2}$ por la expresión (1.26). Para ello denotemos por $p_j^s(x)$ al polinomio de grado s que interpola a $f(x)$ en $\{x_j, \dots, x_{j+s}\}$ y sea $\hat{p}_j^s = p_j^s(x_{i-1/2})$, i.e.

$$p_j^s(x_{j+l}) = f_{j+l}, \quad l = 0, \dots, s \quad \text{y} \quad \hat{p}_j^s = p_j^s(x_{i-1/2}). \quad (1.32)$$

Es sencillo comprobar la siguiente relación:

$$\hat{p}_{i-1}^1 = \frac{1}{2}\hat{p}_{i-1}^0 + \frac{1}{2}\hat{p}_i^0. \quad (1.33)$$

Por otra parte, suponiendo que la función f es suave en $[x_{i-2}, x_{i+1}]$, haciendo desarrollos de Taylor, tenemos:

$$f(x_{i+n}) - f(x_{i+m}) = (n - m)f'(x_i)h + O(h^2) = O(h), \quad n \neq m. \quad (1.34)$$

De esta forma, empleando (1.34) en (1.27), (1.28) y (1.29) obtenemos:

$$\omega_{i-1}^k = \frac{1 + \alpha O(h^2)}{2 + \alpha O(h^2)}, \quad \omega_{i-1}^k = \frac{1 + \alpha O(h^2)}{2 + \alpha O(h^2)}, \quad k = 1, 2, 3, \quad (1.35)$$

de donde podemos deducir la siguiente relación:

$$\omega_{i-1}^k = \frac{1}{2} + O(h^2), \quad \omega_i^k = \frac{1}{2} + O(h^2), \quad k = 1, 2, 3, \quad (1.36)$$

ya que

$$w_l^k - \frac{1}{2} = \frac{1 + \alpha O(h^2)}{2 + \alpha O(h^2)} - \frac{1}{2} = \frac{2\alpha O(h^2) - \alpha O(h^2)}{2(2 + \alpha O(h^2))} = O(h^2), \quad l = i - 1, i.$$

Así podemos probar que el error que cometemos al aproximar $\hat{f}_{i-1/2}$ es de orden 2, pues

$$\begin{aligned}
& \sum_{l=-1}^0 w_{i+l}^k \hat{p}_{i+l}^0 - \hat{p}_{i-1}^1 = \sum_{l=-1}^0 w_{i+l}^k \hat{p}_{i+l}^0 - \sum_{l=-1}^0 \frac{1}{2} \hat{p}_{i+l}^0 = \\
&= \sum_{l=-1}^0 w_{i+l}^k \hat{p}_{i+l}^0 - \sum_{l=-1}^0 \frac{1}{2} \hat{p}_{i+l}^0 + f_{i-1/2} \sum_{l=-1}^0 \frac{1}{2} - f_{i-1/2} \sum_{l=-1}^0 w_{i+l}^k \\
&= \sum_{l=-1}^0 w_{i+l}^k (\hat{p}_{i+l}^0 - f_{i-1/2}) - \sum_{l=-1}^0 \frac{1}{2} (\hat{p}_{i+l}^0 - f_{i-1/2}) = \\
&= \sum_{l=-1}^0 \left(w_{i+l}^k - \frac{1}{2} \right) (\hat{p}_{i+l}^0 - f_{i-1/2}) = O(h^2) \cdot O(h) = O(h^3).
\end{aligned}$$

y haciendo uso de la relación anterior, tenemos:

$$\begin{aligned}
w_{i-1}^k f_{i-1} + w_i^k f_i - f_{i-1/2} &= \sum_{l=0}^1 w_{i-l}^k \hat{p}_{i-l}^0 - f_{i-1/2} = \\
&= \left(\sum_{l=0}^1 w_{i-l}^k \hat{p}_{i-l}^0 - \hat{p}_{i-1}^1 \right) + (\hat{p}_{i-1}^1 - f_{i-1/2}) = \\
&= O(h^3) + O(h^2) = O(h^2), \quad k = 1, 2, 3. \quad (1.37)
\end{aligned}$$

Por tanto, empleando cualquier definición de los pesos dadas en (1.27), (1.28), (1.29), podemos considerar (1.26) como una interpolación WENO de orden 2, pues alcanzamos el orden óptimo empleando 2 nodos. Además hemos visto en (1.36) que los pesos empleados aproximan a los pesos óptimos del mismo modo que los pesos utilizados en la interpolación WENO.

Notar que en caso de que haya una discontinuidad en el intervalo $[x_{i-1}, x_i]$, entonces la aproximación dada por \hat{p}_{i-1}^1 es del orden del salto, i.e.

$$p_{i-1}^1 - f_{i-1/2} = O([f]),$$

y procediendo de forma análoga a (1.37) obtenemos que el error cometido es del orden de la discontinuidad:

$$w_{i-1}^k f_{i-1} + w_i^k f_i - f_{i-1/2} = O([f]). \quad (1.38)$$

1.6

Modificaciones de la interpolación racional

En la siguiente sección, proponemos diversas modificaciones de la interpolación racional. La idea es a partir de combinaciones convexas de aproximaciones de un orden inferior conseguir una mejor aproximación de orden mayor.

1.6.1

Interpolación racional de orden 4

2 polinomios de orden 2

Consideremos los polinomios interpoladores $p_{i-2}^2(x)$, $p_{i-1}^2(x)$ y $p_{i-2}^3(x)$, definidos del mismo modo que en (1.32), y sean \hat{p}_{i-2}^2 , \hat{p}_{i-1}^2 , \hat{p}_{i-2}^3 los valores de éstos en $x_{i-1/2}$ respectivamente, los cuales tienen la siguiente expresión:

$$\begin{aligned}\hat{p}_{i-2}^2 &= -\frac{1}{8}f_{i-2} + \frac{6}{8}f_{i-1} + \frac{3}{8}f_i, \\ \hat{p}_{i-1}^2 &= \frac{3}{8}f_{i-1} + \frac{6}{8}f_i - \frac{1}{8}f_{i+1}, \\ \hat{p}_{i-2}^3 &= -\frac{1}{16}f_{i-2} + \frac{9}{16}f_{i-1} + \frac{9}{16}f_i - \frac{1}{16}f_{i+1}.\end{aligned}\quad (1.39)$$

En este caso, podemos comprobar que se verifica la relación (1.19),

$$\hat{p}_{i-2}^3 = \frac{1}{2}\hat{p}_{i-2}^2 + \frac{1}{2}\hat{p}_{i-1}^2. \quad (1.40)$$

Podemos definir así, una aproximación a $f(x_{i-1/2})$ como sigue:

$$\hat{f}_{i-1/2} = w_{i-1}^k \hat{p}_{i-2}^2 + w_i^k \hat{p}_{i-1}^2, \quad k = 1, 2, 3, \quad (1.41)$$

donde w_{i-1}^k , w_i^k son los pesos definidos en (1.27), (1.28) o (1.29).

Veamos que si la función es suave en $[x_{i-2}, x_{i+1}]$, la interpolación dada por (1.41) es de orden $O(h^4)$. Procediendo igual que en la sección anterior, y empleando (1.36):

$$\begin{aligned} \sum_{l=0}^1 w_{i-l} \hat{p}_{i-1-l}^2 - \hat{p}_{i-2}^3 &= \sum_{l=0}^1 (w_{i-l}^k - \frac{1}{2})(\hat{p}_{i-1-l}^2 - f_{i-1/2}) = \\ &= O(h^2)O(h^3) = O(h^5), \end{aligned} \quad (1.42)$$

lo cual nos permite hallar el error cometido en nuestra aproximación:

$$\begin{aligned} \sum_{l=0}^1 w_{i-l} \hat{p}_{i-1-l}^2 - f_{i-1/2} &= \left(\sum_{l=0}^1 w_{i-l} \hat{p}_{i-1-l}^2 - \hat{p}_{i-2}^3 \right) + (\hat{p}_{i-2}^3 - f_{i-1/2}) = \\ &= O(h^5) + O(h^4) = O(h^4). \end{aligned} \quad (1.43)$$

Así obtenemos una aproximación de orden 4, siendo este el orden óptimo que podemos alcanzar empleando un polinomio interpolador de 4 nodos. En el supuesto de que f posea una discontinuidad en $x_d \in [x_{i-2}, x_{i+1}]$, resulta que $\hat{p}_{i-2}^3 - f_{i-1/2} = O([f])$, siendo el error cometido del orden del salto, es decir:

$$\sum_{l=0}^1 w_{i-l} \hat{p}_{i-1-l}^2 - f_{i-1/2} = O([f]). \quad (1.44)$$

En este caso, (1.45), (1.46) y (1.47) nos muestran las variaciones de las magnitudes de los pesos w_{i-1} , w_i , en función de la localización de la discontinuidad x_d .

$x_d \in]x_{i-2}, x_{i-1}[$:

$$\begin{aligned} w_{i-1}^1 &= \frac{1 + \alpha O(h^2)}{2 + \alpha O([f]^2)}, & w_i^1 &= \frac{1 + \alpha O([f]^2)}{2 + \alpha O([f]^2)}, \\ w_{i-1}^2 &= \frac{1 + \alpha O(h^2)}{2 + \alpha O([f]^2)}, & w_i^2 &= \frac{1 + \alpha O([f]^2)}{2 + \alpha O([f]^2)}, \\ w_{i-1}^3 &= \frac{1 + \alpha O(h^2)}{2 + \alpha O([f]^2)}, & w_i^3 &= \frac{1 + \alpha O([f]^2)}{2 + \alpha O([f]^2)}, \end{aligned} \quad (1.45)$$

$x_d \in]x_{i-1}, x_i[$:

$$\begin{aligned} w_{i-1}^1 &= \frac{1 + \alpha O(h^2)}{2 + \alpha O(h^2)}, & w_i^1 &= \frac{1 + \alpha O(h^2)}{2 + \alpha O(h^2)}, \\ w_{i-1}^2 &= \frac{1 + \alpha O([f]^2)}{2 + \alpha O([f]^2)}, & w_i^2 &= \frac{1 + \alpha O([f]^2)}{2 + \alpha O([f]^2)}, \\ w_{i-1}^3 &= \frac{1 + \alpha O([f]^2)}{2 + \alpha O([f]^2)}, & w_i^3 &= \frac{1 + \alpha O([f]^2)}{2 + \alpha O([f]^2)}, \end{aligned} \quad (1.46)$$

$x_d \in]x_i, x_{i+1}[$:

$$\begin{aligned} w_{i-1}^1 &= \frac{1 + \alpha O([f]^2)}{2 + \alpha O([f]^2)}, & w_i^1 &= \frac{1 + \alpha O(h^2)}{2 + \alpha O([f]^2)}, \\ w_{i-1}^2 &= \frac{1 + \alpha O([f]^2)}{2 + \alpha O([f]^2)}, & w_i^2 &= \frac{1 + \alpha O(h^2)}{2 + \alpha O([f]^2)}, \\ w_{i-1}^3 &= \frac{1 + \alpha O([f]^2)}{2 + \alpha O([f]^2)}, & w_i^3 &= \frac{1 + \alpha O(h^2)}{2 + \alpha O([f]^2)}. \end{aligned} \quad (1.47)$$

Gracias a las relaciones anteriores, podemos ver que la contribución de aquel polinomio cuyo *stencil* cruce una discontinuidad será menor. En caso de que la discontinuidad cruce los dos *stencils* implicados, $x_d \in [x_{i-1}, x_i]$, ambos pesos poseen la misma magnitud.

3 polinomios de orden 1

En este caso vamos a obtener una aproximación a $f_{i-1/2}$ a partir de las interpolaciones dadas por los polinomios $p_{i-2}^1(x)$, $p_{i-1}^1(x)$ y $p_i^1(x)$ (ver (1.32)). La evaluación de dichos polinomios en $x_{i-1/2}$ viene dada por las siguientes expresiones:

$$\begin{aligned} \hat{p}_{i-2}^1 &= -\frac{1}{2}f_{i-2} + \frac{3}{2}f_{i-1}, \\ \hat{p}_{i-1}^1 &= \frac{1}{2}f_{i-1} + \frac{1}{2}f_i, \\ \hat{p}_i^1 &= \frac{3}{2}f_i - \frac{1}{2}f_{i+1}. \end{aligned} \quad (1.48)$$

Teniendo en cuenta las expresiones anteriores, y la expresión de p_{i-2}^3 que podemos ver en (1.39), se puede comprobar la relación

siguiente:

$$\hat{p}_{i-2}^3 = \frac{1}{8}\hat{p}_{i-2}^1 + \frac{6}{8}\hat{p}_{i-1}^1 + \frac{1}{8}\hat{p}_i^1. \quad (1.49)$$

De este modo, definimos la siguiente aproximación a $f(x_{i-1/2})$:

$$\hat{f}_{i-1/2} = w_{i-2}p_{i-2}^1 + w_{i-1}p_{i-1}^1 + w_i p_i^1, \quad (1.50)$$

donde los pesos w_{i-2} , w_{i-1} y w_i vienen dados por la siguiente expresión:

$$\begin{aligned} w_{i-2} &= \frac{1 + |f_{i-1} - f_i|^2 + |f_i - f_{i+1}|^2}{8 + 2 \sum_{k=-2}^0 |f_{i+k} - f_{i+k+1}|^2}, \\ w_{i-1} &= \frac{6 + |f_{i-2} - f_{i-1}|^2 + |f_i - f_{i+1}|^2}{8 + 2 \sum_{k=-2}^0 |f_{i+k} - f_{i+k+1}|^2}, \\ w_i &= \frac{1 + |f_{i-2} - f_{i-1}|^2 + |f_{i-1} - f_i|^2}{8 + 2 \sum_{k=-2}^0 |f_{i+k} - f_{i+k+1}|^2}. \end{aligned} \quad (1.51)$$

Notar que bajo el supuesto de que la función $f(x)$ sea suave en $[x_{i-2}, x_{i+1}]$, los pesos verifican las relaciones siguientes:

$$w_{i-2} - \frac{1}{8} = O(h^2), \quad w_{i-1} - \frac{6}{8} = O(h^2), \quad w_i - \frac{1}{8} = O(h^2). \quad (1.52)$$

las cuales se obtienen fácilmente procediendo de igual modo que en (1.36). Notar además que en caso de que la función posea una discontinuidad, x_d , en el intervalo $[x_{i-2}, x_{i-1}]$, el polinomio afectado por la discontinuidad tiene una menor contribución en la aproximación, para esto es suficiente observar como varía la magnitud de los pesos en función de la localización de la discontinuidad:

$$x_d \in [x_{i-2}, x_{i-1}]$$

$$w_{i-2} = \frac{1 + O(h^2)}{8 + 2O([f]^2)}, \quad w_{i-1} = \frac{6 + O([f]^2)}{8 + 2O([f]^2)}, \quad w_i = \frac{1 + O([f]^2)}{8 + 2O([f]^2)},$$

$$x_d \in [x_{i-1}, x_i]$$

$$w_{i-2} = \frac{1 + O([f]^2)}{8 + 2O([f]^2)}, \quad w_{i-1} = \frac{6 + O(h^2)}{8 + 2O([f]^2)}, \quad w_i = \frac{1 + O([f]^2)}{8 + 2O([f]^2)},$$

$$x_d \in [x_i, x_{i+1}]$$

$$w_{i-2} = \frac{1 + O([f]^2)}{8 + 2O([f]^2)}, \quad w_{i-2} = \frac{6 + O([f]^2)}{8 + 2O([f]^2)}, \quad w_i = \frac{1 + O(h^2)}{8 + 2O([f]^2)}.$$

Veamos ahora que la aproximación dada por (1.50) es de orden $O(h^4)$,

$$\begin{aligned} \sum_{k=-2}^0 w_{i+k} \hat{p}_{i+k}^1 - \hat{p}_{i-2}^3 &= \sum_{k=-2}^0 w_{i+k} \hat{p}_{i+k}^1 - \sum_{k=-2}^0 C_k \hat{p}_{i+k}^1 = \\ &= \sum_{k=-2}^0 w_{i+k} \hat{p}_{i+k}^1 - f_{i-1/2} \sum_{k=-2}^0 w_{i+k} + \\ &\quad + f_{i-1/2} \sum_{k=-2}^0 C_k - \sum_{k=-2}^0 C_k \hat{p}_{i+k}^1 = \\ &= \sum_{k=-2}^0 w_{i+k} (\hat{p}_{i+k}^1 - f_{i-1/2}) - \sum_{k=-2}^0 C_k (\hat{p}_{i+k}^1 - f_{i-1/2}) = \\ &= \sum_{k=-2}^0 (w_{i+k} - C_k) (\hat{p}_{i+k}^1 - f_{i-1/2}) = O(h^2)O(h^2) = \\ &= O(h^4), \end{aligned} \tag{1.53}$$

donde

$$C_{i-2} = \frac{1}{8}, \quad C_{i-1} = \frac{6}{8}, \quad C_i = \frac{1}{8}.$$

Así, haciendo uso de (1.53), obtenemos el orden deseado:

$$\begin{aligned} \sum_{k=-2}^0 w_{i+k} \hat{p}_{i+k}^1 - f_{i-1/2} &= \left(\sum_{k=-2}^0 w_{i+k} \hat{p}_{i+k}^1 - \hat{p}_{i-2}^3 \right) + (\hat{p}_{i-2}^3 - f_{i-1/2}) = \\ &= O(h^4) + O(h^4) = O(h^4). \end{aligned} \tag{1.54}$$

1.6.2

Extrapolación racional ponderada

La interpolación propuesta en [19], dada por la relación (1.26), puede ser interpretada como aproximar el valor de $f(x_{i-1/2})$ a partir de una combinación convexa de los valores extrapolados por

un polinomio de grado 0 a la izquierda de $x_{i-1/2}$, $p_{i-1}^0 = f_{i-1}$, y un polinomio de grado 0 a la derecha de $x_{i-1/2}$, $p_i^0 = f_i$.

Así, podemos obtener una aproximación a $f_{i-1/2}$ empleando dos funciones: $f_l(x)$ y $f_r(x)$, donde $f_l(x)$ interpola a $f(x)$ en nodos x_j , $j \leq i-1$ y $f_r(x)$ interpola a $f(x)$ en nodos x_j , $j \geq i$. Definiendo una aproximación a $f_{i-1/2}$ por medio de la siguiente expresión:

$$\hat{f}_{i-1/2} = w_{i-1}f_l(x_{i-1/2}) + w_i f_r(x_{i-1/2}),$$

donde $w_{i-1} + w_i = 1$, $w_{i-1}, w_i \geq 0$. En caso de que la aproximación dada por f_l o f_r no sea lo suficientemente buena por algún motivo, por ejemplo que el *stencil* empleado cruce alguna discontinuidad si consideramos interpolación polinómica a trozos, nos gustaría que su peso correspondiente, w_{i-1} o w_i respectivamente, sea lo más pequeño posible para así afectar lo menos posible la aproximación.

De este modo, si consideramos los polinomios $p_{i-2}^1(x)$ y $p_i^1(x)$, ver (1.32), podemos definir una nueva aproximación a $f_{i-1/2}$ como:

$$\hat{f}_{i-1/2}^1 = w_{i-1}^k \hat{p}_{i-2}^1 + w_i^k \hat{p}_i^1, \quad k = 1, 2, 3, \quad (1.55)$$

donde w_{i-1}^k , w_i^k son los pesos definidos anteriormente en (1.27), (1.28) y (1.29), y \hat{p}_{i-2}^1 , \hat{p}_i^1 vienen dados por:

$$\begin{aligned} \hat{p}_{i-2}^1 &= -\frac{1}{2}f_{i-2} + \frac{3}{2}f_{i-1}, \\ \hat{p}_i^1 &= \frac{3}{2}f_i - \frac{1}{2}f_{i+1}. \end{aligned}$$

En caso de que la función no posea ninguna discontinuidad en $[x_{i-2}, x_{i+1}]$, observamos gracias a (1.2) que:

$$\hat{p}_{i-2}^1 - f_{i-1/2} = O(h^2), \quad \hat{p}_i^1 - f_{i-1/2} = O(h^2), \quad (1.56)$$

y como $w_{i-1}^k + w_i^k = 1$, $k = 1, 2, 3$, es fácil ver que la aproximación dada por (1.55) es $O(h^2)$, pues

$$\begin{aligned} f_{i-1/2} - (w_{i-1}^k \hat{p}_{i-2}^1 + w_i^k \hat{p}_i^1) &= \\ &= f_{i-1/2} - w_{i-1}^k (f_{i-1/2} + O(h^2)) - w_i^k (f_{i-1/2} + O(h^2)) = O(h^2). \end{aligned}$$

Si dicho intervalo posee una discontinuidad, entonces esta puede estar bien en $[x_{i-2}, x_{i-1/2}]$ o en $[x_{i-1/2}, x_{i+1}]$, obteniendo en el primer caso:

$$\hat{p}_{i-2}^1 - f_{i-1/2} = O([f]), \quad \hat{p}_i^1 - f_{i-1/2} = O(h^2), \quad (1.57)$$

y en el segundo:

$$\hat{p}_{i-2}^1 - f_{i-1/2} = O(h^2), \quad \hat{p}_i^1 - f_{i-1/2} = O([f]). \quad (1.58)$$

Así, si el coeficiente que multiplica al valor extrapolado correspondiente al polinomio afectado por la discontinuidad es lo suficientemente pequeño, esperamos que nuestra aproximación no se vea afectada demasiado por la presencia de la discontinuidad.

1.6.3

Interpolación - Extrapolación racional

En este apartado proponemos un esquema interpolatorio que resulta del uso conjunto de los esquemas propuestos en los apartados 1.6.1 (empleando dos polinomios de segundo orden) y 1.6.2.

Como demostramos anteriormente, ver (1.43) y (1.44), el error cometido empleando la aproximación dada por:

$$\hat{f}_{i-1/2} = w_{i-1}^k \hat{p}_{i-2}^2 + w_i^k \hat{p}_{i-1}^2, \quad k = 1, 2, 3,$$

viene dado por:

$$\hat{f}_{i-1/2} - \sum_{l=0}^1 w_{i-l}^k \hat{p}_{i-l-1}^2 = \begin{cases} O(h^4), & f \text{ suave en } [x_{i-2}, x_{i+1}], \\ O([f]), & f \text{ discontinua en } [x_{i-2}, x_{i+1}]. \end{cases}$$

Por otra parte, si aproximamos $f_{i-1/2}$ por medio de la combinación convexa de $\hat{p}_{i-1}^0 = f_{i-1}$ y $\hat{p}_i^0 = f_i$ obtenemos una aproximación de orden $O(h^2)$ si la función es suave en $[x_{i-1}, x_i]$ y de $O([f])$ en caso de que la función sea discontinua en dicho intervalo. Donde la aproximación obtenida es una aproximación monótona que intenta conservar la discontinuidad. Si empleamos como en la sección anterior los valores extrapolados por los polinomios de orden 1 sustituidos en $x_{i-1/2}$, $p_{i-2}^1(x_{i-1/2})$ y $p_i^1(x_{i-1/2})$ obtenemos una aproximación de orden $O(h^2)$ si la función es suave en $[x_{i-2}, x_{i+1}]$, y de $O([f])$ en caso de que la función sea discontinua en dicho intervalo. No obstante bajo el supuesto de que la función sea discontinua, nos interesa que el peso que acompaña al valor dado por el polinomio cuya aproximación se ve degradada por la presencia de

la discontinuidad sea pequeño, y por tanto la aproximación calculada se vea afectada lo mínimo posible. Entonces en caso de que la discontinuidad se encuentre en $[x_{i-1/2}, x_i]$ o $[x_{i-1}, x_{i-1/2}]$, nos interesa obtener nuestra aproximación por medio de valores extrapolados bien a partir de polinomios constantes, bien haciendo uso del esquema propuesto en la subsección 1.6.2, ya la aproximación dada por éste vendrá dada prácticamente por \hat{p}_{i-1}^1 o \hat{p}_i^1 respectivamente, siendo estos valores calculados a partir de polinomios interpoladores cuya *stencil* no cruza ninguna discontinuidad.

Si suponemos que $x_d \in]x_{i-1}, x_i[$ y fuera de este intervalo la función es suave, entonces se verifican las siguientes relaciones:

$$\begin{aligned} |f_{i-2} - f_{i-1}| &= O(h), \\ |f_{i-1} - f_i| &= O([f]), \\ |f_i - f_{i+1}| &= O(h), \end{aligned}$$

las cuales son sencillas de demostrar mediante desarrollos de Taylor. De esta forma, si se verifica la relación siguiente:

$$|f_{i-1} - f_i| > \text{máx} \{|f_{i-2} - f_{i-1}|, |f_i - f_{i+1}|\},$$

supondremos que $x_d \in [x_{i-1}, x_i]$, y por lo tanto aproximamos $f_{i-1/2}$ por la combinación convexa de \hat{p}_{i-1}^0 , \hat{p}_i^0 ó \hat{p}_{i-2}^1 , \hat{p}_i^1 si decidimos extrapolar por medio de rectas.

Resumiendo, nuestra aproximación a $f_{i-1/2}$ viene dada por la siguiente expresión:

$$\hat{f}_{i-1/2} = \delta_i \hat{p}_{k,l}^E + (1 - \delta_i) \hat{p}_k^I, \quad k = 1, 2, 3, \quad l = 0, 1, \quad (1.59)$$

donde δ_i es:

$$\delta_i = \begin{cases} 1, & |f_{i-1} - f_i| > \text{máx} \{|f_{i-2} - f_{i-1}|, |f_i - f_{i+1}|\}, \\ 0, & |f_{i-1} - f_i| \leq \text{máx} \{|f_{i-2} - f_{i-1}|, |f_i - f_{i+1}|\}, \end{cases}$$

\hat{p}_k^I es la aproximación obtenida en la subsección 1.6.1, i.e.

$$\hat{p}_k^I = w_{i-1}^k \hat{p}_{i-2}^2 + w_i^k \hat{p}_{i-1}^2, \quad k = 1, 2, 3,$$

y $\hat{p}_{k,l}^E$ es la aproximación obtenida haciendo uso de los valores extrapolados de polinomios constantes si $l = 0$ o rectas si $l = 1$:

$$\hat{p}_{k,l}^E = \begin{cases} w_{i-1}^k \hat{p}_{i-1}^0 + w_i^k \hat{p}_i^0, & \text{si } l = 0 \\ w_{i-1}^k \hat{p}_{i-2}^1 + w_i^k \hat{p}_i^1, & \text{si } l = 1. \end{cases}$$

1.7

Ejemplo numérico

Veamos a continuación un ejemplo de como se comportan las técnicas interpolatorias anteriores frente a una discontinuidad. Para ello hemos tomado dos discretizaciones uniformes del intervalo $[0, 1]$, X^1 y X^2 ,

$$X^1 = \{x_i^1\}_{i=0}^{J_{10}}, \quad J_{10} = 2^{10}, \quad h_{J_5} = x_i^1 - x_{i-1}^1 = 2^{-10}, \quad i = 1, \dots, 2^{10},$$

$$X^2 = \{x_i^2\}_{i=0}^{J_5}, \quad J_5 = 2^5, \quad h_{J_5} = x_i^2 - x_{i-1}^2 = 2^{-5}, \quad i = 1, \dots, 2^5,$$

y a partir de la discretización X^2 hemos aplicado las diferentes técnicas interpolatorias recursivamente hasta obtener una aproximación a los datos en la discretización X^1 . La figura 1.4 muestra los datos en ambas discretizaciones. Las figuras 1.5 y 1.6 muestran

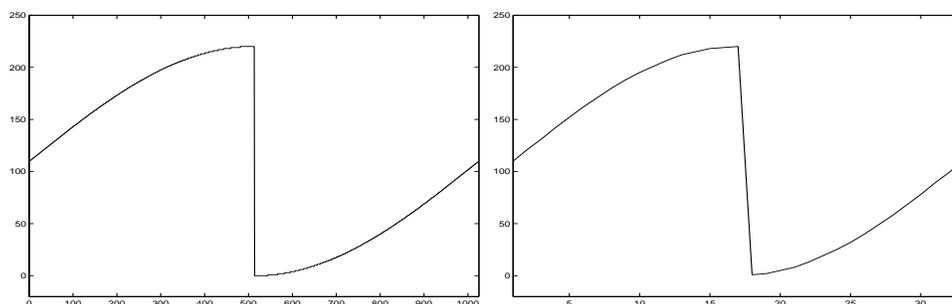


Figura 1.4: Izquierda: $f(X^1)$. Derecha: $f(X^2)$.

los resultados obtenidos con las distintas técnicas interpolatorias utilizadas, siendo éstas:

LIN4 : interpolación lineal. Aproximación de orden 4 utilizando polinomios centrados de grado 3.

ENO4 : interpolación ENO. Aproximación de orden 4 utilizando polinomios de grado 3 (Sección 1.2).

WENO4 : interpolación WENO. Aproximación de orden 4 utilizando polinomios de grado 2 (Sección 1.4).

RAT : interpolación racional propuesta por S. Carrato y G. Ramponi (Sección 1.5).

RAT42 : modificación de la interpolación racional empleando dos polinomios de segundo grado (Sección 1.6.1).

RAT41 : modificación de la interpolación racional empleando tres polinomios de primer grado (Sección 1.6.1).

RATE2 : extrapolación racional (Sección 1.6.2).

RATIE1 : interpolación-extrapolación racional, empleando polinomios constantes para extrapolar (Sección 1.6.3).

RATIE2 : interpolación-extrapolación racional, empleando polinomios de primer grado para extrapolar (Sección 1.6.3).

Se puede observar que únicamente al emplear la técnica interpolatoria lineal obtenemos oscilaciones alrededor de la discontinuidad. Además la técnica RATE2 es la que recupera dicha discontinuidad mejor. Por otra parte, la tabla 1.2 nos muestra el error en $\|\cdot\|_2$ cometido por cada técnica interpolatoria.

LIN4	0.7028	RAT	0.8384
ENO4	0.5586	RAT42	0.7082
WENO4	0.6862	RAT41	0.7457
RATE2	0.8393	RATIE1	0.7306
RATIE2	0.7222		

Tabla 1.2: Errores cometidos en $\|\cdot\|_2$

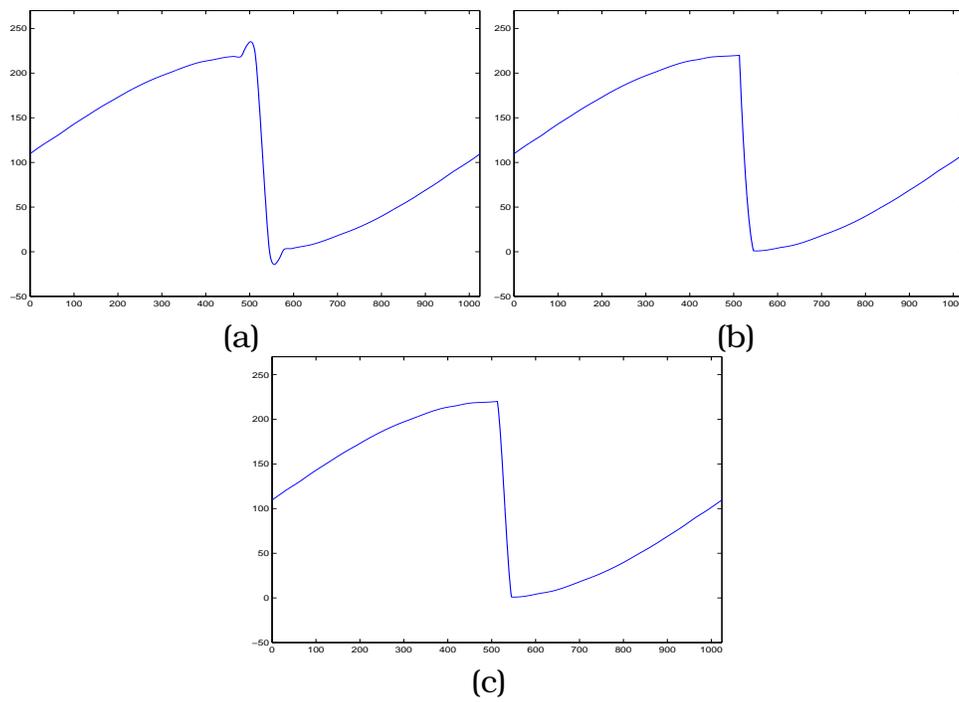


Figura 1.5: (a) LIN4 (b) ENO4 (c) WENO4

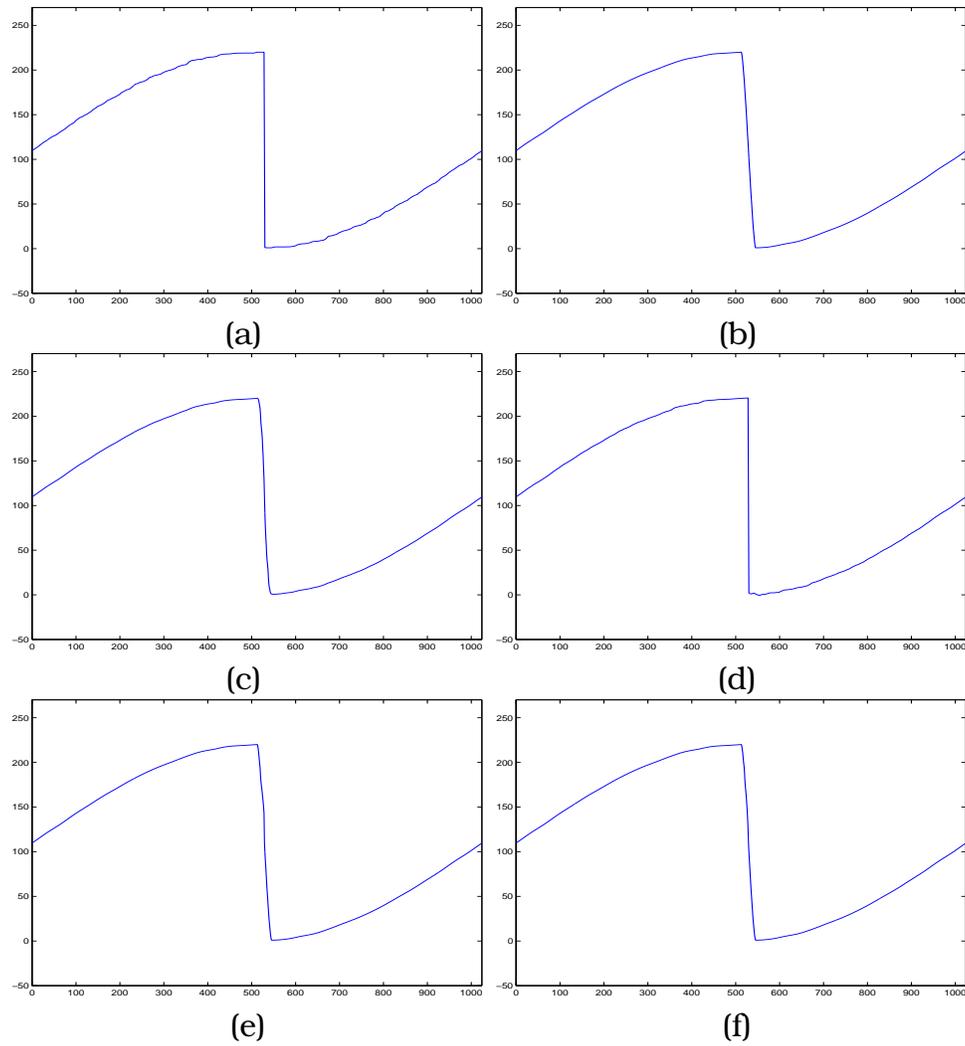


Figura 1.6: (a) RAT (b) RAT42 (c) RAT41 (d) RATE2 (e) RATIE1 (f) RATIE2

1.8

Interpolación 2D para valores puntuales

Supongamos, en esta sección, que los datos que tenemos, $\{f_{i,j}^k\}_{i,j=0}^{J_k}$ son los valores puntuales de una función real de dos variables definida sobre $[0, 1] \times [0, 1]$, i.e.,

$$f : [0, 1] \times [0, 1] \longrightarrow \mathbb{R}.$$

De este modo $\{f_{i,j}^k\}_{i,j=0}^{J_k}$ son los valores que toma la función $f(x, y)$ en una discretización uniforme del cuadrado $[0, 1] \times [0, 1]$, es decir $f(x_i^k, y_j^k)$, donde

$$x_i^k = i \cdot h_k, \quad y_j^k = j \cdot h_k, \quad h_k = \frac{1}{J_k}, \quad i, j = 0, \dots, J_k,$$

y k indica el nivel de resolución. En este caso, podemos hallar una aproximación a los valores de f en un nivel de resolución superior de dos formas: empleando técnicas interpolatorias uno dimensionales explicadas anteriormente via producto tensorial o, empleando una interpolación puramente dos dimensional.

En el caso de hacer uso de las técnicas interpolatorias uno dimensionales, en primer lugar hallaríamos una aproximación a un nivel de resolución intermedio al k y $k + 1$, que denotaremos por $k + 1/2$, éste nivel estaría constituido por la aproximación de f en la malla constituida por los nodos $(x_i^{k+1/2}, y_j^k)$, donde $x_i^{k+1/2} = i \cdot h_{k+1/2}$, $h_{k+1/2} = h_k/2$, $i = 0, \dots, J_{k+1/2} = 2J_k$, y $y_j^k = j \cdot h_k$, $j = 0, \dots, J_k$ donde dicha aproximación sería obtenida aplicando la técnica interpolatoria deseada a $\{f_{i,j}^k\}_{i=0}^{J_k}$ para cada $0 \leq j \leq J_k$. Posteriormente repetiríamos el proceso anterior pero ahora con $\{f_{i,j}^{k+1/2}\}_{j=0}^{J_k}$ para cada $0 \leq i \leq J_{k+1/2}$, siendo el resultado final $\{f_{i,j}^{k+1}\}_{i,j=0}^{J_{k+1}}$, (ver figura 1.7). En este caso diríamos que hemos obtenido la aproximación al nivel $k + 1$ aplicando tensorialmente una técnica interpolatoria uno dimensional. El lector interesado sobre la aplicación tensorial de técnicas interpolatorias puede consultar [1, 12, 3, 15].

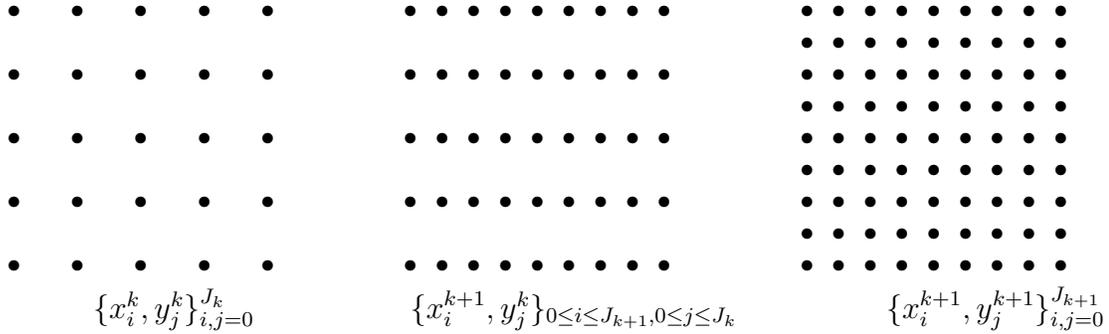


Figura 1.7: Mallas discretas en los niveles k , $k + 1/2$ y $k + 1$.

Veamos ahora como obtener una aproximación a los datos en el nivel de resolución $k + 1$ empleando una interpolación directamente en dos dimensiones. El caso más sencillo es emplear una interpolación polinómica a trozos lineal, y por tanto independiente de los datos. Así fijado $s \geq 1$, podemos considerar la siguiente familia de polinomios en dos variables:

$$p_{i,j}^s = \sum_{0 \leq l, m \leq 2s} a_{l,m} x^l y^m, \quad (1.60)$$

los coeficientes de dichos polinomios son determinados imponiendo las siguientes condiciones,

$$p_{i,j}^s(x_{i+l}^k, y_{j+m}^k) = f_{i+l, j+m}^k, \quad l, m = -s + 1, \dots, s. \quad (1.61)$$

De este modo, si $p(x, y)$ es un polinomio que verifica las relaciones dadas en (1.61), para $s \geq 1$, definimos entonces la aproximación en el nivel de resolución $k + 1$ del siguiente modo:

$$\begin{aligned} f_{2i, 2j}^{k+1} &= f_{i,j}^k, & i = 0, \dots, J_k, \quad j = 0, \dots, J_k & \quad (1.62) \\ f_{2i+1, 2j}^{k+1} &= p_{i,j}^s(x_{2i+1}^{k+1}, y_{2j}^{k+1}), & i = 0, \dots, J_k - 1, \quad j = 0, \dots, J_k \\ f_{2i, 2j+1}^{k+1} &= p_{i,j}^s(x_{2i}^{k+1}, y_{2j+1}^{k+1}), & i = 0, \dots, J_k, \quad j = 0, \dots, J_k - 1 \\ f_{2i+1, 2j+1}^{k+1} &= p_{i,j}^s(x_{2i+1}^{k+1}, y_{2j+1}^{k+1}), & i = 0, \dots, J_k - 1, \quad j = 0, \dots, J_k - 1 \end{aligned}$$

Del mismo modo que en el caso uno-dimensional, el polinomio interpolador de una función f en los nodos (x_i, y_j) , $0 \leq i \leq m$,

$0 \leq j \leq n$, donde x_i, y_j se distribuyen arbitrariamente en los ejes OX y OY , y $x_l \neq x_h, y_l \neq y_h$ si $l \neq h$ viene dado por la expresión siguiente (ver [27, 44]):

$$p(x, y) = \sum_{i=0}^m \sum_{j=0}^n f[x_0, \dots, x_i; y_0, \dots, y_j] \prod_{h=0}^{i-1} (x - x_h) \prod_{k=0}^{j-1} (y - y_k), \quad (1.63)$$

donde los productos vacíos toman el valor 1.

Además para f lo suficientemente suave, se puede demostrar que existen ξ, ξ', μ, μ' verificando:

$$\begin{aligned} f(x, y) - p(x, y) = R(x, y) &= \frac{\partial^{m+1} f(\xi, y)}{\partial x^{m+1}} \frac{\prod_{i=0}^m (x - x_i)}{(m+1)!} + \\ &+ \frac{\partial^{m+1} f(x, \mu)}{\partial y^{m+1}} \frac{\prod_{j=0}^n (y - y_j)}{(n+1)!} - \\ &- \frac{\partial^{m+n+2} f(\xi', \mu')}{\partial x^{m+1} \partial y^{n+1}} \frac{\prod_{i=0}^m (x - x_i)}{(m+1)!} \frac{\prod_{j=0}^n (y - y_j)}{(n+1)!}. \end{aligned} \quad (1.64)$$

Así, en caso de la función f sea lo suficientemente suave, a partir de (1.64) podemos ver que la aproximación obtenida con los polinomios $p_{i,j}^s$, polinomios que interpolan a f en $(2s+1) \times (2s+1)$ nodos, es de orden $O(h_k^{2s+1})$. De igual modo que en el caso unidimensional, la calidad de la aproximación se verá afectada en aquellas zonas en las que el *stencil* empleado cruce una discontinuidad de la función.

A continuación veremos como realizar una interpolación no lineal en la que el *stencil* empleado dependa de la zona en la que deseamos hacer la predicción con el objetivo de evitar que éste cruce discontinuidades y, por tanto, evitar que el orden de interpolación se vea afectado.

1.9

Interpolación no lineal en 2D para valores puntuales

En [27], M. Gasca y T. Sauer repasan los resultados esenciales acerca de la interpolación polinómica en varias variables. En este

32 1.9. Interpolación no lineal en 2D para valores puntuales

artículo podemos encontrar desde como construir el polinomio interpolador, así como el error cometido por éste en la aproximación. Dado que nuestro objetivo es realizar una interpolación polinómica a trozos en dos variables, de modo que los *stencils* empleados no crucen discontinuidades, a continuación proponemos diversa formas de elegir los *stencils*.

En los apartados siguientes, supondremos que poseemos una discretización uniforme del cuadrado $[0, 1] \times [0, 1]$, es decir $\{(x_i, y_j)\}_{i,j=0}^N$, $x_i = y_i = i \cdot h$, $h = 1/N$.

1.9.1

Stencils cuadrados

Una posibilidad es considerar *stencils* cuadrados formados por $4s^2$ nodos, con $s \geq 1$. De este modo, para cada par (l, n) verificando $1 - s \leq l, n \leq s - 1$, obtenemos el *stencil* $S_{l,n}^s$, formado por

$$S_{l,n}^s = \{(x_{i-s+1+l}, y_{j+m}), \dots, (x_{i+s+l}, y_{j+m})\}_{m=-s+1+n}^{s+n}.$$

Notar que tenemos entonces $(2s-1)^2$ *stencils* distintos, conteniendo cada uno de ellos a los nodos $\{(x_i, y_j), (x_{i+1}, y_j), (x_i, y_{j+1}), (x_{i+1}, y_{j+1})\}$. El *stencil* seleccionado para cada (x_i, y_j) será aquel que minimice el siguiente valor:

$$\sum_{m=-s+1+n}^{s+n} \sum_{c=-s+1+l}^{s+l-1} |f_{i+1+c, j+m}^k - f_{i+c, j+m}| + \sum_{m=-s+1+n}^{s+n-1} \sum_{c=-s+1+l}^{s+l} |f_{i+c, j+m+1}^k - f_{i+c, j+m}|, \quad (1.65)$$

el cual no es más que la suma de todas las diferencias divididas de primer orden entre los nodos consecutivos del *stencil*.

Una vez elegido el *stencil*, definimos la predicción de igual modo que en (1.62) empleando el polinomio interpolador de f en los nodos seleccionados, el cual pertenecerá a la familia de polinomios dada por (1.60). Así en caso de que la función f sea lo suficientemente suave, podemos ver haciendo uso de (1.64) que la técnica interpolatoria es de orden $O(h^{2s})$

1.9.2

Stencils triangulares

Otra posibilidad es construir los polinomios interpoladores haciendo uso de *stencils* triangulares, de este modo podemos construir un polinomio $p^{6t}(x, y)$,

$$p^{6t}(x, y) = \sum_{0 \leq l+m \leq 2} a_{l,m} x^l y^m. \quad (1.66)$$

Dicho polinomio interpolaría los datos originales en los 6 nodos del *stencil* seleccionado, eligiendo aquel *stencil* de igual modo que en el caso de emplear *stencils* cuadrados, i.e. aquel que minimiza la suma de las diferencias divididas entre nodos consecutivos. Una vez construido el polinomio interpolador, la predicción vendrá dada por la relación (1.62). La figura 1.8 nos muestra los 4 posibles *stencils* que se presentan en el caso de emplear 6 nodos.

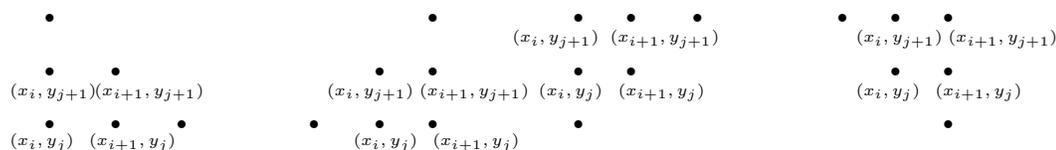


Figura 1.8: 4 posibles *stencils* triangulares empleando 6 nodos.

Procediendo de igual modo, podemos considerar *stencils* triangulares formados por 10 nodos, siendo el polinomio resultante la siguiente forma:

$$p^{10t}(x, y) = \sum_{0 \leq l+m \leq 3} a_{l,m} x^l y^m. \quad (1.67)$$

En este caso el número de posibles *stencils* se eleva a 12, procediendo igual que en el caso anterior tanto para la elección del *stencil* como para la predicción.

1.9.3

Implementación de una interpolación 2D

El polinomio interpolador de una función $f(x, y)$ en los nodos $\{(x_k, y_l)\}_{0 \leq k \leq m, 0 \leq l \leq n}$ dado por la expresión (1.63), puede expresarse del siguiente modo:

$$\begin{aligned}
 p(x, y) &= \sum_{k=0}^{m-1} f[x_0, \dots, x_k; y_0] \prod_{i=0}^{k-1} (x - x_k) + \\
 &+ (y - y_0) \left(\sum_{k=0}^{m-1} f[x_0, \dots, x_k; y_0, y_1] \prod_{i=0}^{k-1} (x - x_k) \right) + \\
 &+ \dots \\
 &+ \prod_{l=0}^{n-1} (y - y_l) \left(\sum_{k=0}^{m-1} f[x_0, \dots, x_k; y_0, \dots, y_n] \prod_{i=0}^{k-1} (x - x_k) \right).
 \end{aligned} \tag{1.68}$$

Resulta evidente, observando (1.68), que los valores obtenidos al interpolar cuando $y = y_0$ no vienen influenciados por los valores $\{f(\cdot, y_j)\}_{j \neq 0}$ (análogamente los valores $\{f(x_i, \cdot)\}_{i \neq 0}$ no influyen en las aproximaciones de los nodos con $x = x_0$). Este hecho nos facilita el conocimiento de los filtros a emplear en caso de realizar una interpolación dos-dimensional, siempre y cuando la coordenada x o la coordenada y del punto donde queramos conocer su aproximación coincida con alguna de los nodos empleados.

Así, si empleamos la notación de la sección 1.8, y consideramos que los datos que tenemos en el nivel de resolución k son $\{f_{i,j}^k\}_{i,j=0}^{J_k}$, podemos obtener las aproximaciones a $f_{2i+1,2j}^{k+1}$ y $f_{2i,2j+1}^{k+1}$ a partir de filtros uno-dimensionales. Pues si optamos por emplear *stencils* cuadrados o triangulares, una vez seleccionado el *stencil*, el valor $\hat{f}_{2i+1,2j}^{k+1}$ es calculado a partir de $\{f_{\cdot,j}^k\}$ y de igual modo en el cálculo de $\hat{f}_{2i,2j+1}^{k+1}$ solo intervienen los valores $\{f_{i,\cdot}^k\}$. Respecto al cálculo de $\hat{f}_{2i+1,2j+1}^{k+1}$, en caso de emplear *stencils* dos-dimensionales, no es posible realizar la simplificación anterior.

1.10

Resultados numéricos

Dada una imagen cualquiera de tamaño $(2^{n_0} + 1) \times (2^{m_0} + 1)$, se puede aumentar su tamaño aplicándole las técnicas interpolatorias anteriores de modo que el resultado final es una imagen de tamaño $(2^{n_0+1} + 1) \times (2^{m_0+1} + 1)$. En caso de aplicar k veces la misma técnica interpolatoria recursivamente, el tamaño final de la imagen es $(2^{n_0+k} + 1) \times (2^{m_0+k} + 1)$. En este experimento hemos partido de una imagen que tiene de una resolución de 65×65 píxeles, a la que hemos aplicado recursivamente la misma técnica interpolatoria hasta obtener una imagen de tamaño 513×513 píxeles. En la figura 1.9 podemos ver la imagen empleada a baja resolución y en la resolución original que pretendemos recuperar tras aplicar recursivamente las diversas técnicas interpolatorias.

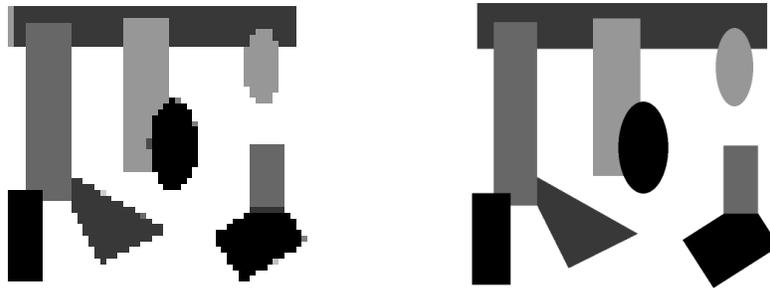


Figura 1.9: *Imagen geométrica. Izquierda: 65×65 píxeles. Derecha: 513×513 píxeles*

Los acrónimos empleados para diferenciar los distintos tipos de interpolación son los siguientes:

LIN4 : interpolación lineal.

ENO4 : interpolación ENO4 producto tensorial.

WENO4 : interpolación WENO4 producto tensorial.

ENO2DT : interpolación ENO dos dimensional con *stencils* triangulares (10 nodos).



Figura 1.10: Arriba: *LIN4 - ENO4*. Abajo: *WENO4 - ENO2DT*

RAT : interpolación racional propuesta por S. Carrato y G. Ramponi.

RAT42 : modificación de la interpolación racional empleando dos polinomios de segundo grado.

RAT41 : modificación de la interpolación racional empleando tres polinomios de primer grado.

RATE2 : extrapolación racional.

RATIE2 : interpolación-extrapolación racional, empleando polinomios de primer grado para extrapolar.

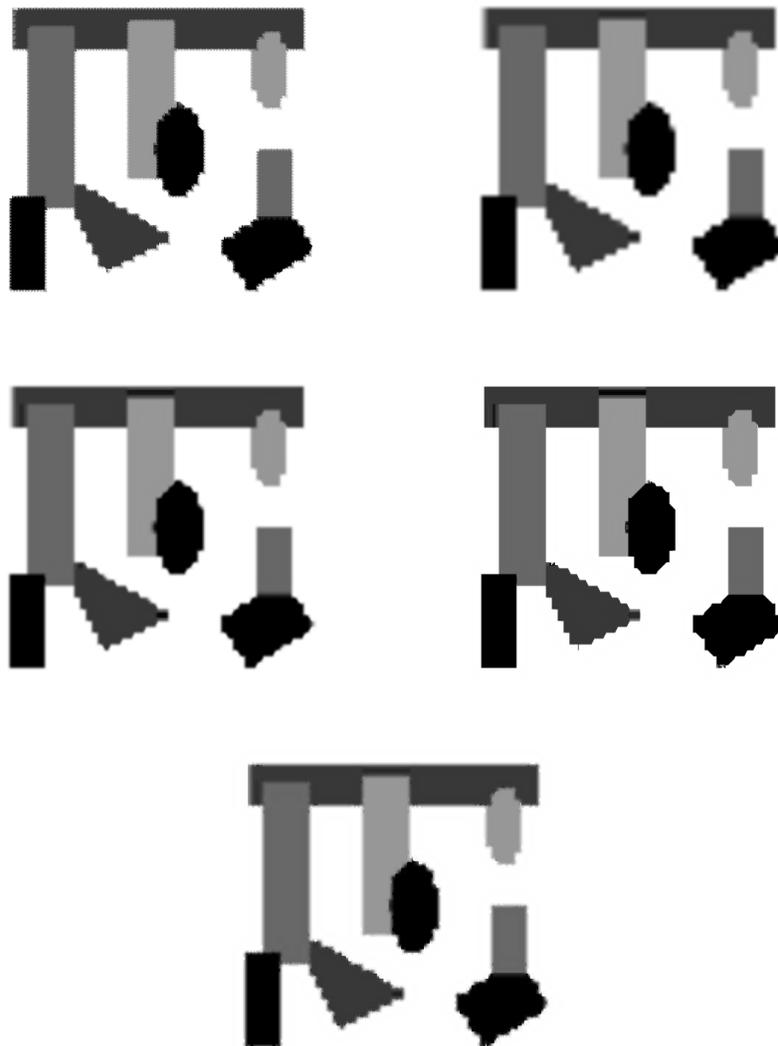


Figura 1.11: Arriba: RAT - RAT42. Centro: RAT41 - RATE2. Abajo: RATIE2

Podemos observar que la técnica ENO2DT reduce la difusión de los contornos obtenida por LIN4 y a la vez los resultados obtenidos en el rectángulo horizontal superior son mejores que los que proporciona la aplicación tensorial de la técnica ENO. También se puede apreciar que los resultados obtenidos con la técnica ENO2DT son similares a los obtenidos con la técnica WENO4. Respecto a las técnicas derivadas de la interpolación racional, todas producen unos contornos más definidos que LIN4, ENO4, WENO4 y ENO2DT, siendo RAT, RATE2 y RATIE2 las que consiguen una mayor definición en los perfiles. Sin embargo, hay que destacar el pronunciado efecto escalera presente en los perfiles diagonales, producto de la aplicación tensorial de técnicas interpolatorias uno dimensionales.

2

Interpolación para medias en celda

2.1

Conceptos previos

El problema de la interpolación para medias en celda consiste en: conocidas las medias de una función $f(x)$ en una discretización dada, encontrar una función de modo que posea las mismas medias que $f(x)$ en dicha discretización. Denotaremos por $\{\bar{f}_i\}_{i=1}^N$ los datos iniciales, donde \bar{f}_i es la media de f en el intervalo $I_i = [x_{i-1}, x_i]$, i.e.

$$\bar{f}_i = \frac{1}{h} \int_{x_{i-1}}^{x_i} f(x) dx, \text{ con } h = x_i - x_{i-1}$$

con $x_i = ih$, $i = 0, \dots, N$, $Nh = 1$.

De igual modo que en la interpolación para valores puntuales, podemos emplear un interpolante polinómico a trozos. Así, en este contexto, la interpolación lineal consiste en construir, para cada intervalo I_i , un polinomio $p_i(x)$ que posea la misma media que f en dicho intervalo, de modo que el *stencil* empleado para construir los polinomios interpoladores sea el mismo para todos los intervalos y, por tanto, independiente de los datos. La elección más habitual consiste en elegir un *stencil* centrado compuesto por $2s + 1$ intervalos ($s \geq 1$), siendo así p_i un polinomio de grado $2s$ que verifica las condiciones siguientes:

$$\frac{1}{h} \int_{I_{i+l}} p_i(x) dx = \bar{f}_{i+l}, \quad \text{con } l \in \{-s, \dots, s\}. \quad (2.1)$$

La existencia y unicidad del polinomio $p_i(x)$ es sencilla de probar estudiando el carácter del sistema lineal dado por (2.1).

La *reconstrucción vía función primitiva*, nos muestra otra manera de construir un polinomio que interpole las medias de la función f . Esta construcción aprovecha la relación existente entre los valores puntuales de la primitiva de una función y las medias de ésta. Sea $F(x)$ la única primitiva de $f(x)$ que verifica $F(0) = 0$, i.e. $F(x) = \int_0^x f(y) dy$, y sea F_i los valores puntuales correspondientes a F en los nodos $x_i = 0, \dots, N$,

$$F_i = \int_0^{x_i} f(y) dy, \quad i = 0, \dots, N.$$

Se verifican entonces las siguientes igualdades:

$$\begin{aligned} \bar{f}_i &= \frac{F_i - F_{i-1}}{h}, \quad \forall i = 1, \dots, N, \\ \bar{F}_i &= h \sum_{k=0}^i \bar{f}_k, \quad \forall i = 0, \dots, N, \end{aligned} \quad (2.2)$$

que nos muestran las relaciones existentes entre los valores puntuales de la función F y la medias en celda de f . Así si tomamos un interpolante de los valores puntuales de F en $\{x_i\}_{i=0}^N$, $\mathcal{I}(x; F)$, se puede definir una función, $\mathcal{R}(x; \bar{f})$ que posea las mismas medias que la función f en los intervalos I_i , $i = 1, \dots, N$, del siguiente modo:

$$\mathcal{R}(x; \bar{f}) = \frac{d}{dx} \mathcal{I}(x; F), \quad (2.3)$$

pues dado que $\mathcal{I}(x_i, F) = F_i$, tenemos $\forall 1 \leq i \leq N$:

$$\frac{1}{h} \int_{x_{i-1}}^{x_i} \mathcal{R}(x; \bar{f}) \, dx = \frac{1}{h} \int_{x_{i-1}}^{x_i} \frac{d}{dx} \mathcal{I}(x; F) \, dx = \frac{F_i - F_{i-1}}{h} = \bar{f}_i.$$

Veamos ahora el error cometido al aproximar $f(x)$ por $\mathcal{R}(x; \bar{f})$. Supongamos que $\mathcal{I}(x; F)$ es un interpolante polinómico a trozos de F de orden $r + 1$, construido a partir de los nodos x_0, \dots, x_r , con $x_i = ih$. Así, a partir de (1.2) se tiene que:

$$F(x) - \mathcal{I}(x; F) = F[x_0, \dots, x_r, x] \prod_{i=0}^r (x - x_i), \quad \forall x \in [x_0, x_r].$$

Si f es continua en x , derivando la expresión anterior se obtiene el error cometido al aproximar $f(x)$ por $\mathcal{R}(x; \bar{f})$:

$$\begin{aligned} f(x) - \mathcal{R}(x; \bar{f}) &= f(x) - \frac{d}{dx} \mathcal{I}(x; F) = \frac{d}{dx} F[x_0, \dots, x_r, x] \prod_{i=0}^r (x - x_i) \\ &\quad + F[x_0, \dots, x_r, x] \sum_{j=0}^r \prod_{\substack{i=0 \\ i \neq j}}^r (x - x_i), \end{aligned} \quad (2.4)$$

donde si $f^{(p)}$ posee una discontinuidad en $[x_0, x_r]$, y por tanto discontinuidad de $F^{(p+1)}$, haciendo uso de (1.7) obtenemos:

$$f(x) - \mathcal{R}(x; \bar{f}) = \begin{cases} O(h^r) & \text{si } r \leq p, \\ O([f^{(p)}])h^p & \text{en otro caso.} \end{cases} \quad (2.5)$$

Si por el contrario f es discontinua en x , y suponemos que existe un $\delta > 0$ de modo que f es continua en $]x - \delta, x[$ y $]x, x + \delta[$, podemos ver el error cometido en x^+ y x^- empleando la expresión (2.4),

$$\begin{aligned} f(x \pm \delta) - \mathcal{R}(x \pm \delta; \bar{f}) &= \frac{d}{dx} F[x_0, \dots, x_r, x \pm \delta] \prod_{i=0}^r (x \pm \delta - x_i) \\ &\quad + F[x_0, \dots, x_r, x \pm \delta] \sum_{j=0}^r \prod_{\substack{i=0 \\ i \neq j}}^r (x \pm \delta - x_i), \end{aligned}$$

donde haciendo $\delta \rightarrow 0$, obtenemos otra vez la expresión (2.5) para x^\pm .

De igual modo que en la interpolación de valores puntuales, la calidad de la aproximación se ve degradada en aquellos casos que el *stencil* empleado cruza una singularidad de f . Las técnicas interpolatorias no lineales nos permiten ampliar la zona de aproximación óptima. Así podemos construir un interpolante polinómico a trozos, en el que los *stencils* utilizados son elegidos mediante técnicas ENO (ver algoritmos 1.1 y 1.2). Así, si $f(x)$ presenta un salto en $x_d \in I_i$, y fuera de este intervalo es suficientemente suave, empleando polinomios de grado r para la construcción de $\mathcal{I}(x; F)$, tenemos que el error cometido viene dado por:

$$f(x) = \begin{cases} \mathcal{R}(x; \bar{f}) + O(h^r), & \forall x \in I_k, k \neq i, \\ \mathcal{R}(x; \bar{f}) + O([f]) & \forall x \in I_i. \end{cases}$$

Notar que a pesar de haber definido el interpolante polinómico de $f(x)$ como la derivada de un interpolante de su función primitiva $F(x)$, el cual expresado en la forma del polinomio de Newton (1.1) emplea las diferencias divididas de F , puede ser expresado en función de las diferencias divididas de \bar{f} gracias a la siguiente relación:

$$\bar{f}[x_i, \dots, x_{i+n}] = (n+1)F[x_{i-1}, \dots, x_{i+n}]. \quad (2.6)$$

Dicha relación es sencilla de demostrar por inducción, pues si $n = 0$, es evidente que:

$$\bar{f}_i = \frac{F_i - F_{i-1}}{h} = F[x_{i-1}, x_i].$$

Supongamos (2.6) cierta para diferencias divididas de orden n , veamos que se verifica para $n+1$:

$$\begin{aligned} \bar{f}[x_i, \dots, x_{i+n+1}] &= \frac{\bar{f}[x_{i+1}, \dots, x_{i+n+1}] - \bar{f}[x_i, \dots, x_{i+n}]}{x_{i+n+1} - x_i} \\ &= \frac{(n+1)\bar{F}[x_i, \dots, x_{i+n+1}] - (n+1)\bar{F}[x_{i-1}, \dots, x_{i+n}]}{(n+1)h} \\ &= (n+2) \frac{\bar{F}[x_i, \dots, x_{i+n+1}] - \bar{F}[x_{i-1}, \dots, x_{i+n}]}{(n+2)h} \\ &= (n+2)F[x_{i-1}, \dots, x_{i+n+1}]. \end{aligned}$$

En el caso de que f posea un salto en $x_d \in I_j$, podemos aplicar la técnica ENO subcell resolution a la función F , ya que x_d es una esquina de ésta. Aprovechando este hecho podemos localizar la posición de la discontinuidad en el intervalo I_j .

Supongamos ahora que tenemos una partición uniforme del intervalo $[0, 1]$, en un nivel de resolución k , $X^k = \{x_i^k\}_{i=0}^{N_k}$, $x_i^k = ih_k$, $h_k = 1/N_k$, y conocemos las medias de una función $f(x)$, $\{\bar{f}_i^k\}_{i=0}^{N_k}$, en los intervalos $I_i^k = [x_{i-1}^k, x_i^k]$, $i = 1, \dots, N_k$,

$$\bar{f}_i^k = \frac{1}{h_k} \int_{x_{i-1}^k}^{x_i^k} f(x) dx, i = 1, \dots, N_k.$$

Consideremos un nivel de resolución mayor, $k + 1$, de modo que $X^{k+1} = \{x_i^{k+1}\}_{i=0}^{N_{k+1}}$, verifique

$$x_i^{k+1} = ih_{k+1}, N_{k+1} = 2N_k, h_{k+1} = 1/N_{k+1} = h_k/2. \quad (2.7)$$

Notar que siempre en este caso

$$I_i^k = I_{2i-1}^{k+1} \cup I_{2i}^{k+1},$$

y por tanto para obtener una aproximación en un nivel de resolución mayor debemos obtener una aproximación de las medias de la función $f(x)$ en las celdas I_{2i-1}^{k+1} y I_{2i}^{k+1} . Dichas aproximaciones serán, en función del *stencil* S_i^k empleado, las medias de un polinomio p_i^s de grado $2s$ cuyas medias coinciden con las de $f(x)$ en las $2s + 1$ celdas del *stencil* calculadas en I_{2i-1}^{k+1} y I_{2i}^{k+1} .

A continuación mostramos las aproximaciones obtenidas, en función del *stencil* seleccionado, para los casos $s = 0$ y $s = 1$ son:

- $s = 0$

$$S_i^k = \{I_i^k\}$$

$$\begin{aligned} \hat{f}_{2i-1}^{k+1} &= \bar{f}_i^k, \\ \hat{f}_{2i}^{k+1} &= \bar{f}_i^k. \end{aligned}$$

- $s = 1$

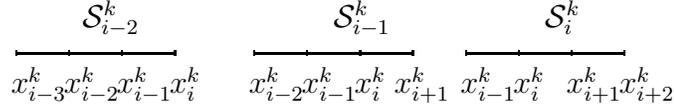


Figura 2.1: Stencils posibles para el caso $s = 1$.

$$\mathcal{S}_{i-2}^k = \{I_{i-2}^k, I_{i-1}^k, I_i^k\}$$

$$\hat{f}_{2i-1}^{k+1} = -\frac{1}{8}\bar{f}_{i-2}^k + \frac{4}{8}\bar{f}_{i-1}^k + \frac{5}{8}\bar{f}_i^k,$$

$$\hat{f}_{2i}^{k+1} = \frac{1}{8}\bar{f}_{i-2}^k - \frac{4}{8}\bar{f}_{i-1}^k + \frac{11}{8}\bar{f}_i^k.$$

$$\mathcal{S}_{i-1}^k = \{I_{i-1}^k, I_i^k, I_{i+1}^k\}$$

$$\hat{f}_{2i-1}^{k+1} = \frac{1}{8}\bar{f}_{i-1}^k + \bar{f}_i^k - \frac{1}{8}\bar{f}_{i+1}^k,$$

$$\hat{f}_{2i}^{k+1} = -\frac{1}{8}\bar{f}_{i-1}^k + \bar{f}_i^k + \frac{1}{8}\bar{f}_{i+1}^k.$$

$$\mathcal{S}_i^k = \{I_i^k, I_{i+1}^k, I_{i+2}^k\}$$

$$\hat{f}_{2i-1}^{k+1} = \frac{11}{8}\bar{f}_i^k - \frac{4}{8}\bar{f}_{i+1}^k + \frac{1}{8}\bar{f}_{i+2}^k,$$

$$\hat{f}_{2i}^{k+1} = \frac{5}{8}\bar{f}_i^k + \frac{4}{8}\bar{f}_{i+1}^k - \frac{1}{8}\bar{f}_{i+2}^k.$$

Notar que se verifica la siguiente relación:

$$\hat{f}_{2i-1}^{k+1} + \hat{f}_{2i}^{k+1} = 2\bar{f}_i^k,$$

ya que

$$\begin{aligned} \bar{f}_i^k &= \frac{1}{h_k} \int_{x_{i-1}^k}^{x_i^k} f(x) dx = \frac{1}{h_k} \left(\int_{x_{2i-2}^{k+1}}^{x_{2i-1}^{k+1}} f(x) dx + \int_{x_{2i-1}^{k+1}}^{x_{2i}^{k+1}} f(x) dx \right) = \\ &= \frac{1}{2} \left(\frac{2}{h_k} \left(\int_{x_{2i-2}^{k+1}}^{x_{2i-1}^{k+1}} p_i^s(x) dx + \int_{x_{2i-1}^{k+1}}^{x_{2i}^{k+1}} p_i^s(x) dx \right) \right) = \\ &= \frac{1}{2} \left(\hat{f}_{2i-1}^{k+1} + \hat{f}_{2i}^{k+1} \right). \end{aligned}$$

De este modo, conocida la aproximación a la media de la función en la celda I_{2i-1}^{k+1} , \hat{f}_{2i}^{k+1} se obtiene de forma inmediata a partir de:

$$\hat{f}_{2i}^{k+1} = 2\bar{f}_i^k - \hat{f}_{2i-1}^{k+1} \quad (2.8)$$

2.2

Interpolación racional para medias en celda

En esta sección proponemos dos modificaciones de la interpolación racional para valores puntuales de G. Ramponi ([19, 20, 21, 19, 42]) para el contexto de la interpolación de medias en celda. Así, continuando con la notación anterior, conocidas las medias de $f(x)$ en la discretización X^k , obtendremos una aproximación a las medias de $f(x)$ en el nivel de resolución $k + 1$, donde X^{k+1} viene definido como en en (2.7). De igual modo que la aproximación racional para valores puntuales, nuestra propuesta para medias en celda consiste en aproximar la media del intervalo deseado por una combinación convexa de todas las aproximaciones de polinomios de un determinado orden que interpolen la media del intervalo deseado. En nuestro caso proponemos dos aproximaciones, la primera obtenida a partir de dos polinomios de primer grado y la segunda calculada a partir de tres polinomios de segundo grado. Notar que en ambos casos, únicamente calculamos la aproximación para la celda I_{2i-1}^{k+1} , ya que \hat{f}_{2i}^{k+1} puede ser obtenida utilizando la relación (2.8).

Aproximación a partir de dos polinomios de primer grado

Sea $(p_i^r)_{2j-1}$ la media del polinomio de grado r , $p_i^r(x)$, cuyas medias coinciden con las medias de la función $f(x)$ en los intervalos $\{I_i^k, I_{i+1}^k, \dots, I_{i+r}^k\}$,

$$\int_{I_{i+l}^k} p_i^r(x) dx = \int_{I_{i+l}^k} f(x) dx, \quad \forall l \in \{0, \dots, r\},$$

calculada en el intervalo I_{2j-1}^{k+1} ,

$$(p_i^r)_{2j-1} = \frac{1}{I_{2j-1}^{k+1}} \int_{I_{2j-1}^{k+1}} p_i^r(x) dx.$$

Así tenemos que las medias de los polinomio $p_{i-1}^2(x), p_{i-1}^1(x)$ y $p_i^1(x)$ en el intervalo I_{2i-1}^{k+1} vienen dadas por las siguientes expresiones:

$$\begin{aligned}(p_{i-1}^2)_{2i-1} &= \frac{1}{8}\bar{f}_{i-1}^k + \bar{f}_i^k - \frac{1}{8}\bar{f}_{i+1}^k, \\(p_{i-1}^1)_{2i-1} &= \frac{1}{4}\bar{f}_{i-1}^k + \frac{3}{4}\bar{f}_i^k, \\(p_i^1)_{2i-1} &= \frac{5}{4}\bar{f}_i^k - \frac{1}{4}\bar{f}_{i+1}^k.\end{aligned}$$

Es fácil verificar que:

$$(p_{i-1}^2)_{2i-1} = \frac{1}{2}(p_{i-1}^1)_{2i-1} + \frac{1}{2}(p_i^1)_{2i-1}.$$

Aproximamos \hat{f}_{2i-1}^{k+1} por el valor dado por la expresión siguiente:

$$\hat{f}_{2i-1}^{k+1} = \omega_{i-1}(p_{i-1}^1)_{2i-1} + \omega_i(p_i^1)_{2i-1}, \quad (2.9)$$

siendo ω_{i-1}, ω_i :

$$\begin{aligned}\omega_{i-1} &= \frac{1 + |\bar{f}_{i+1}^k - \bar{f}_i^k|^2}{2 + |\bar{f}_i^k - \bar{f}_{i-1}^k|^2 + |\bar{f}_{i+1}^k - \bar{f}_i^k|^2}, \\ \omega_i &= \frac{1 + |\bar{f}_i^k - \bar{f}_{i-1}^k|^2}{2 + |\bar{f}_i^k - \bar{f}_{i-1}^k|^2 + |\bar{f}_{i+1}^k - \bar{f}_i^k|^2}.\end{aligned} \quad (2.10)$$

Veamos el error cometido por nuestra aproximación. Bajo el supuesto de que los intervalos $\{I_{i-1}, I_i, I_{i+1}\}$ no presenten ninguna singularidad, se verifican entonces las siguientes relaciones:

$$\begin{aligned}(p_{i-1}^2)_{2i-1} &= \hat{f}_{2i-1}^{k+1} + O(h_k^3), \\(p_{i-1}^1)_{2i-1} &= \hat{f}_{2i-1}^{k+1} + O(h_k^2), \\(p_i^1)_{2i-1} &= \hat{f}_{2i-1}^{k+1} + O(h_k^2).\end{aligned}$$

Además,

$$\begin{aligned}\omega_i - \frac{1}{2} &= \frac{1 + |\bar{f}_{i+1}^k - \bar{f}_i^k|^2}{2 + |\bar{f}_i^k - \bar{f}_{i-1}^k|^2 + |\bar{f}_{i+1}^k - \bar{f}_i^k|^2} - \frac{1}{2} = \\ &= \frac{2 \cdot (1 + |\bar{f}_{i+1}^k - \bar{f}_i^k|^2) - (2 + |\bar{f}_i^k - \bar{f}_{i-1}^k|^2 + |\bar{f}_{i+1}^k - \bar{f}_i^k|^2)}{2 \cdot (2 + |\bar{f}_i^k - \bar{f}_{i-1}^k|^2 + |\bar{f}_{i+1}^k - \bar{f}_i^k|^2)} = \\ &= \frac{|\bar{f}_{i+1}^k - \bar{f}_i^k|^2 - |\bar{f}_i^k - \bar{f}_{i-1}^k|^2}{2 \cdot (2 + |\bar{f}_i^k - \bar{f}_{i-1}^k|^2 + |\bar{f}_{i+1}^k - \bar{f}_i^k|^2)} = O(h_k^2),\end{aligned}$$

y de igual modo se demuestra que $\omega_{i-1} - \frac{1}{2} = O(h_k^2)$.

Para demostrar que $|\bar{f}_{i+1}^k - \bar{f}_i^k| = O(h_k)$ es suficiente emplear la relación entre \bar{f}_i^k y la función primitiva, (2.2), y tomar desarrollos de Taylor entorno a x_i^k bajo el supuesto de que la función es suave (ver [3] para más detalles).

Así tenemos que:

$$\begin{aligned}
 & \sum_{l=-1}^0 w_{i+l} (p_{i+l}^1)_{2i-1} - (p_{i-1}^2)_{2i-1} = \sum_{l=-1}^0 w_{i+l} (p_{i+l}^1)_{2i-1} - \sum_{l=-1}^0 \frac{1}{2} (p_{i+l}^1)_{2i-1} = \\
 & = \sum_{l=-1}^0 w_{i+l} (p_{i+l}^1)_{2i-1} - \sum_{l=-1}^0 \frac{1}{2} (p_{i+l}^1)_{2i-1} + \bar{f}_{2i-1}^{k+1} \sum_{l=-1}^0 \frac{1}{2} - \bar{f}_{2i-1}^{k+1} \sum_{l=-1}^0 w_{i+l} = \\
 & = \sum_{l=-1}^0 w_{i+l} ((p_{i+l}^1)_{2i-1} - \bar{f}_{2i-1}^{k+1}) - \sum_{l=-1}^0 \frac{1}{2} ((p_{i+l}^1)_{2i-1} - \bar{f}_{2i-1}^{k+1}) = \\
 & = \sum_{l=-1}^0 \left(w_{i+l} - \frac{1}{2} \right) \cdot ((p_{i+l}^1)_{2i-1} - \bar{f}_{2i-1}^{k+1}) = O(h_k^2) \cdot O(h_k^2) = O(h_k^4).
 \end{aligned}$$

A partir de esta relación, obtenemos finalmente el orden de nuestra aproximación:

$$\begin{aligned}
 \sum_{l=-1}^0 w_{i+l} (p_{i+l}^1)_{2i-1} - \bar{f}_{2i-1}^{k+1} & = \left(\sum_{l=-1}^0 w_{i+l} (p_{i+l}^1)_{2i-1} - (p_{i-1}^2)_{2i-1} \right) \\
 & + ((p_{i-1}^2)_{2i-1} - \bar{f}_{2i-1}^{k+1}) = \\
 & = O(h_k^4) + O(h_k^3) = O(h_k^3).
 \end{aligned}$$

Luego la aproximación obtenida es de orden $O(h_k^3)$ siendo así el error cometido del mismo orden que el obtenido al interpolar empleando un polinomio de segundo grado.

Aproximación a partir de tres polinomios de segundo grado

En este caso emplearemos las aproximaciones obtenidas por los tres polinomios de segundo grado que contienen al intervalo I_i^k , intentado obtener el mismo orden de aproximación que al aproximar

por el polinomio de cuarto grado que interpola las medias de los intervalos $\{I_{i+l}^k\}_{l=-2}^2$. Siguiendo la notación anterior tenemos:

$$\begin{aligned} (p_{i-2}^4)_{2i-1} &= -\frac{3}{128}\bar{f}_{i-2}^k + \frac{11}{64}\bar{f}_{i-1}^k + \bar{f}_i^k - \frac{11}{64}\bar{f}_{i+1}^k + \frac{3}{128}\bar{f}_{i+2}^k, \\ (p_{i-2}^2)_{2i-1} &= -\frac{1}{8}\bar{f}_{i-2}^k + \frac{1}{2}\bar{f}_{i-1}^k + \frac{5}{8}\bar{f}_i^k, \\ (p_{i-1}^2)_{2i-1} &= \frac{1}{8}\bar{f}_{i-1}^k + \bar{f}_i^k - \frac{1}{8}\bar{f}_{i+1}^k, \\ (p_i^2)_{2i-1} &= \frac{11}{8}\bar{f}_i^k - \frac{1}{2}\bar{f}_{i+1}^k + \frac{1}{8}\bar{f}_{i+2}^k. \end{aligned}$$

Teniendo en cuenta las relaciones anteriores, tenemos que:

$$(p_{i-2}^4)_{2i-1} = \frac{3}{16}(p_{i-2}^2)_{2i-1} + \frac{10}{16}(p_{i-1}^2)_{2i-1} + \frac{3}{16}(p_i^2)_{2i-1}.$$

Definimos nuestra aproximación a \bar{f}_{2i-1}^{k+1} como:

$$\hat{f}_{2i-1}^{k+1} = \omega_{i-2}(p_{i-2}^2)_{2i-1} + \omega_{i-1}(p_{i-1}^2)_{2i-1} + \omega_i(p_i^2)_{2i-1}, \quad (2.11)$$

siendo $\omega_{i-2}, \omega_{i-1}, \omega_i$:

$$\begin{aligned} \omega_{i-2} &= \frac{3 + |\bar{f}_{i+1}^k - \bar{f}_i^k|^2 + |\bar{f}_{i+2}^k - \bar{f}_{i+1}^k|^2}{16 + 2(|\bar{f}_{i-1}^k - \bar{f}_{i-2}^k|^2 + |\bar{f}_{i+2}^k - \bar{f}_{i+1}^k|^2) + |\bar{f}_i - \bar{f}_{i-1}|^2 + |\bar{f}_{i+1} - \bar{f}_i|^2}, \\ \omega_{i-1} &= \frac{10 + |\bar{f}_{i-1}^k - \bar{f}_{i-2}^k|^2 + |\bar{f}_{i+2}^k - \bar{f}_{i+1}^k|^2}{16 + 2(|\bar{f}_{i-1}^k - \bar{f}_{i-2}^k|^2 + |\bar{f}_{i+2}^k - \bar{f}_{i+1}^k|^2) + |\bar{f}_i - \bar{f}_{i-1}|^2 + |\bar{f}_{i+1} - \bar{f}_i|^2}, \\ \omega_i &= \frac{3 + |\bar{f}_{i-1}^k - \bar{f}_{i-2}^k|^2 + |\bar{f}_i^k - \bar{f}_{i-1}^k|^2}{16 + 2(|\bar{f}_{i-1}^k - \bar{f}_{i-2}^k|^2 + |\bar{f}_{i+2}^k - \bar{f}_{i+1}^k|^2) + |\bar{f}_i - \bar{f}_{i-1}|^2 + |\bar{f}_{i+1} - \bar{f}_i|^2}. \end{aligned} \quad (2.12)$$

En el caso de que la función sea suave en los intervalos $\{I_{i+l}^k\}_{l=-2}^2$, se verifican las siguientes igualdades:

$$\begin{aligned} (p_{i-2}^4)_{2i-1} &= \hat{f}_{2i-1}^{k+1} + O(h_k^5), \\ (p_{i-2}^2)_{2i-1} &= \hat{f}_{2i-1}^{k+1} + O(h_k^3), \\ (p_{i-1}^2)_{2i-1} &= \hat{f}_{2i-1}^{k+1} + O(h_k^3), \\ (p_i^2)_{2i-1} &= \hat{f}_{2i-1}^{k+1} + O(h_k^3), \end{aligned}$$

Además, teniendo en cuenta que $|\bar{f}_{l+1}^k - \bar{f}_l^k|^2 = O(h_k^2)$ y, por tanto,

$$\omega_{i-2} = \frac{3}{16} + O(h_k^2), \quad \omega_{i-1} = \frac{10}{16} + O(h_k^2) \text{ y } \omega_i = \frac{3}{16} + O(h_k^2),$$

podemos obtener el orden de nuestra aproximación:

$$\begin{aligned} \sum_{l=-2}^0 (p_{i+l}^2)_{2i-1} - (p_{i-2}^4)_{2i-1} &= \sum_{l=-2}^0 \omega_{i+l} p_{i+l}^2 - \sum_{l=-2}^0 C_{i+l} (p^2)_{2i-1} = \\ \sum_{l=-2}^0 (p_{i+l}^2)_{2i-1} - \sum_{l=-2}^0 C_{i+l} (p^2)_{2i-1} + \bar{f}_{2i-1}^{k+1} \sum_{l=-2}^0 C_{i+l} - \bar{f}_{2i-1}^{k+1} \sum_{l=-2}^0 \omega_{i+l} &= \\ \sum_{l=-2}^0 \omega_{i+l} ((p_{i+l}^2)_{2i-1} - \bar{f}_{2i-1}^{k+1}) - \sum_{l=-2}^0 C_{i+l} ((p_{i+l}^2)_{2i-1} - \bar{f}_{2i-1}^{k+1}) &= \\ \sum_{l=-2}^0 (\omega_{i+l} - C_{i+l}) \cdot ((p_{i+l}^2)_{2i-1} - \bar{f}_{2i-1}^{k+1}) &= O(h_k^2) \cdot O(h_k^3) = O(h_k^5), \end{aligned}$$

siendo $C_{i-2} = C_i = \frac{3}{16}$, $C_{i-1} = \frac{10}{16}$.

Finalmente,

$$\begin{aligned} \sum_{l=-2}^0 \omega_{i+l} (p_{i+l}^2)_{2i-1} - \bar{f}_{2i-1}^{k+1} &= \left(\sum_{l=-2}^0 \omega_{i+l} (p_{i+l}^2)_{2i-1} - ((p_{i-2}^4)_{2i-1}) \right) \\ &\quad + ((p_{i-2}^4)_{2i-1} - \bar{f}_{2i-1}^{k+1}) \\ &= O(h_k^5) + O(h_k^5) = O(h_k^5). \end{aligned}$$

Por tanto, nuestra aproximación es de orden cinco, siendo este el orden óptimo que podemos alcanzar al involucrar a cinco intervalos.

2.3

Ejemplo numérico

A continuación, repetimos el experimento realizado en la sección 1.7 considerando ahora los datos, ver figura 1.4, como las

medias de una cierta función $f(x)$ en las siguientes discretizaciones del intervalo $[0, 1]$:

$$I_i^2 = [x_{i-1}^2, x_i^2], \quad i = 0, \dots, 2^5,$$

$$I_i^1 = [x_{i-1}^1, x_i^1], \quad i = 0, \dots, 2^{10},$$

donde x_i^1 y x_i^2 son definidos del mismo modo que en la sección 1.7. Para ello hemos aplicado recursivamente las técnicas interpolatorias siguientes:

LIN3 : interpolación lineal de orden 3 (3 intervalos).

LIN5 : interpolación lineal de orden 5 (5 intervalos).

ENO3 : interpolación ENO de orden 3.

WENO3 : interpolación WENO de orden 3.

RAT3 : interpolación racional con dos polinomios de primer grado (Sección 2.2).

RAT5 : interpolación racional con tres polinomios de segundo grado (Sección 2.2).

La figura 2.2 muestra los resultados obtenidos, y la tabla 2.1 recoge los errores en $\|\cdot\|_2$ cometidos por cada técnica. En ellas podemos ver como el fenómeno de Gibbs presente en las técnicas lineales desaparece con las técnicas ENO3, WENO3 y con las dos modificaciones propuestas de la interpolación racional. Esto último es debido a que los pesos introducidos en dichas aproximaciones hacen que la contribución del polinomio interpolador que cruza la discontinuidad sea pequeña.

Si observamos los errores cometido por cada técnica podemos ver que: de las técnicas interpolatorias de orden 3, el error cometido por ENO3, WENO3 y RAT3 es similar e inferior al cometido al emplear la interpolación lineal LIN3. Respecto a las técnicas interpolatorias de orden 5, la modificación racional produce un error menor que al emplear interpolación lineal de dicho orden.

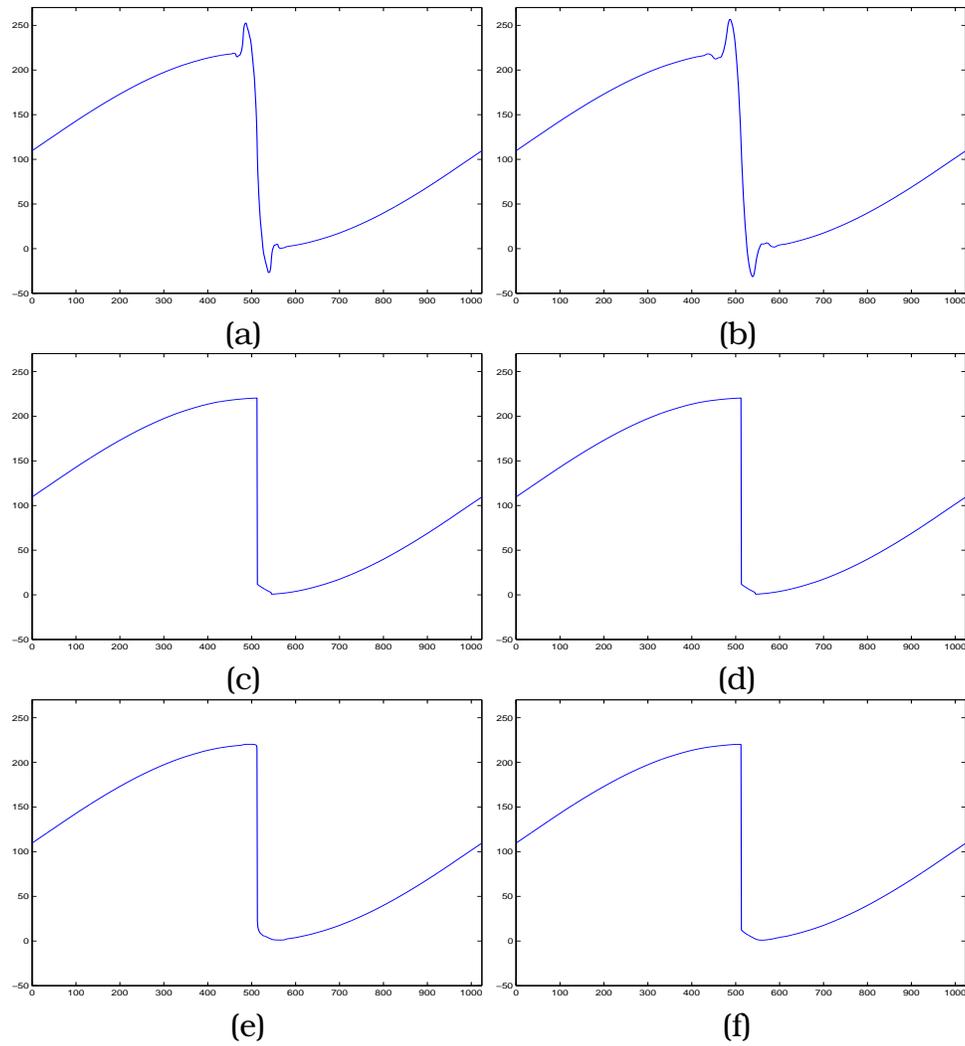


Figura 2.2: (a) LIN3 (b) LIN5 (c) ENO3 (d) WENO3 (e) RAT3 (f) RAT5

LIN3	9.1817
LIN5	10.5362
ENO3	6.6341
WENO3	6.6930
RAT3	6.2942
RAT5	6.6140

Tabla 2.1: Errores cometidos en $\|\cdot\|_2$

2.4

Interpolación 2D para medias en celda

Supongamos en este caso, que los datos $\{\bar{f}_{i,j}^k\}_{i,j=1}^{N_k}$, representan las medias de una función $f(x, y)$ en las celdas $C_{i,j}^k$, donde éstas son una discretización uniforme de $[0, 1] \times [0, 1]$ para una resolución k , i.e.,

$$C_{i,j}^k = [x_{i-1}^k, x_i^k] \times [y_{j-1}^k, y_j^k], \quad x_i^k = i \cdot h_k, \quad y_j^k = j \cdot h_k, \quad i, j = 0, \dots, N_k, \quad h_k = 1/N_k, \quad (2.13)$$

luego

$$\bar{f}_{i,j}^k = \frac{1}{h_k^2} \int_{C_{i,j}^k} f(x, y) \, dx \, dy.$$

En el caso de considerar un nivel de resolución mayor, $k + 1$, de modo que $h_{k+1} = h_k/2$, el objetivo es obtener una aproximación a las medias de $f(x, y)$ en las celdas $C_{i,j}^{k+1}$ definidas de igual modo que en (2.13). Notar que así definidas se verifica la relación siguiente entre las celdas de dos niveles de resolución consecutivos (ver figura 2.3):

$$C_{i,j}^k = C_{2i-1,2j-1}^{k+1} \cup C_{2i-1,2j}^{k+1} \cup C_{2i,2j-1}^{k+1} \cup C_{2i,2j}^{k+1}. \quad (2.14)$$

A partir de la aditividad de la integral, y teniendo en cuenta (2.14), obtenemos la siguiente relación entre las medias de dos

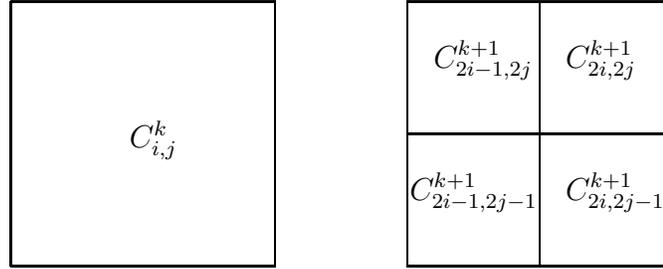


Figura 2.3: Izq. $C_{i,j}^k$. Der. $C_{2i-1,2j-1}^{k+1} \cup C_{2i-1,2j}^{k+1} \cup C_{2i,2j-1}^{k+1} \cup C_{2i,2j}^{k+1}$

niveles de resolución consecutivos:

$$\begin{aligned}
 \bar{f}_{i,j}^k &= \frac{1}{h_k^2} \int_{C_{i,j}^k} f(x,y) dx dy \\
 &= \frac{1}{h_k^2} \left(\int_{C_{2i,2j}^{k+1}} f(x,y) dx dy + \int_{C_{2i-1,2j}^{k+1}} f(x,y) dx dy \right. \\
 &\quad \left. + \int_{C_{2i,2j-1}^{k+1}} f(x,y) dx dy + \int_{C_{2i-1,2j-1}^{k+1}} f(x,y) dx dy \right) \\
 &= \frac{1}{4} (\bar{f}_{2i,2j}^{k+1} + \bar{f}_{2i-1,2j}^{k+1} + \bar{f}_{2i,2j-1}^{k+1} + \bar{f}_{2i-1,2j-1}^{k+1}). \tag{2.15}
 \end{aligned}$$

De igual modo que en el caso unidimensional (ver (2.8)), las aproximaciones a las medias de $f(x)$ en las celdas $C_{2i-1,2j-1}^{k+1}$, $C_{2i-1,2j}^{k+1}$, $C_{2i,2j-1}^{k+1}$, $C_{2i,2j}^{k+1}$, verificarán:

$$4\bar{f}_{i,j}^k = \hat{f}_{2i-1,2j-1}^{k+1} + \hat{f}_{2i-1,2j}^{k+1} + \hat{f}_{2i,2j-1}^{k+1} + \hat{f}_{2i,2j}^{k+1}. \tag{2.16}$$

La aplicación tensorial de las técnicas interpolatorias unidimensionales para medias en celda, nos permite obtener una aproximación a $\bar{f}_{2i-1,2j-1}^{k+1}$, $\bar{f}_{2i-1,2j}^{k+1}$, $\bar{f}_{2i,2j-1}^{k+1}$ y $\bar{f}_{2i,2j}^{k+1}$.

En caso de querer aplicar una interpolación polinómica puramente dos-dimensional, el caso más sencillo es emplear una interpolación polinómica lineal, para ello dado $s \geq 0$, consideramos para cada celda el siguiente polinomio

$$p_{i,j}^s = \sum_{0 \leq l,m \leq 2s} a_{l,m} x^l y^m,$$

de modo que los coeficientes $a_{l,m}$ son calculados a partir de las siguientes $(2s + 1)^2$ ecuaciones:

$$\frac{1}{h_k^2} \int_{x_{i+l-1}^k}^{x_{i+l}^k} \int_{y_{j+m-1}^k}^{y_{j+m}^k} p_{i,j}^s(x, y) dx dy = \bar{f}_{i+l,j+m}^k, \quad l, m = -s, \dots, s.$$

Las aproximaciones a las medias de $f(x, y)$ en las celdas $C_{2i-1,2j-1}^{k+1}$, $C_{2i-1,2j}^{k+1}$, $C_{2i,2j-1}^{k+1}$ y $C_{2i,2j}^{k+1}$, son obtenidas como la media del polinomio $p_{i,j}^s(x, y)$ en la celda correspondiente, es decir:

$$\begin{aligned} \hat{f}_{2i-1,2j-1}^{k+1} &= \frac{1}{h_{k+1}^2} \int_{x_{2i-2}^{k+1}}^{x_{2i-1}^{k+1}} \int_{y_{2j-2}^{k+1}}^{y_{2j-1}^{k+1}} p_{i,j}^s(x, y) dx dy, \\ \hat{f}_{2i-1,2j}^{k+1} &= \frac{1}{h_{k+1}^2} \int_{x_{2i-2}^{k+1}}^{x_{2i-1}^{k+1}} \int_{y_{2j-1}^{k+1}}^{y_{2j}^{k+1}} p_{i,j}^s(x, y) dx dy, \\ \hat{f}_{2i,2j-1}^{k+1} &= \frac{1}{h_{k+1}^2} \int_{x_{2i-1}^{k+1}}^{x_{2i}^{k+1}} \int_{y_{2j-2}^{k+1}}^{y_{2j-1}^{k+1}} p_{i,j}^s(x, y) dx dy, \\ \hat{f}_{2i,2j}^{k+1} &= \frac{1}{h_{k+1}^2} \int_{x_{2i-1}^{k+1}}^{x_{2i}^{k+1}} \int_{y_{2j-1}^{k+1}}^{y_{2j}^{k+1}} p_{i,j}^s(x, y) dx dy. \end{aligned}$$

En el caso de tomar $s = 1$, es decir empleando *stencils* formados por 3×3 celdas, las expresiones anteriores son:

$$\begin{aligned} \hat{f}_{2i-1,2j-1}^{k+1} &= \bar{f}_{i,j}^k + \frac{1}{8}(\bar{f}_{i,j-1}^k + \bar{f}_{i-1,j}^k - \bar{f}_{i+1,j}^k - \bar{f}_{i,j+1}^k) \\ &\quad + \frac{1}{64}(\bar{f}_{i+1,j+1}^k + \bar{f}_{i-1,j-1}^k - \bar{f}_{i+1,j-1}^k - \bar{f}_{i-1,j+1}^k), \\ \hat{f}_{2i-1,2j}^{k+1} &= \bar{f}_{i,j}^k + \frac{1}{8}(\bar{f}_{i,j-1}^k - \bar{f}_{i-1,j}^k + \bar{f}_{i+1,j}^k - \bar{f}_{i,j+1}^k) \\ &\quad - \frac{1}{64}(\bar{f}_{i+1,j+1}^k + \bar{f}_{i-1,j-1}^k - \bar{f}_{i+1,j-1}^k - \bar{f}_{i-1,j+1}^k), \\ \hat{f}_{2i,2j-1}^{k+1} &= \bar{f}_{i,j}^k + \frac{1}{8}(-\bar{f}_{i,j-1}^k + \bar{f}_{i-1,j}^k - \bar{f}_{i+1,j}^k + \bar{f}_{i,j+1}^k) \\ &\quad - \frac{1}{64}(\bar{f}_{i+1,j+1}^k + \bar{f}_{i-1,j-1}^k - \bar{f}_{i+1,j-1}^k - \bar{f}_{i-1,j+1}^k), \\ \hat{f}_{2i,2j}^{k+1} &= \bar{f}_{i,j}^k + \frac{1}{8}(-\bar{f}_{i,j-1}^k - \bar{f}_{i-1,j}^k + \bar{f}_{i+1,j}^k + \bar{f}_{i,j+1}^k) \\ &\quad + \frac{1}{64}(\bar{f}_{i+1,j+1}^k + \bar{f}_{i-1,j-1}^k - \bar{f}_{i+1,j-1}^k - \bar{f}_{i-1,j+1}^k). \quad (2.17) \end{aligned}$$

Para $s \geq 1$ podemos expresar las aproximaciones anteriores (ver [11, 16]) del siguiente modo:

$$\begin{aligned}
\hat{f}_{2i-1,2j-1}^{k+1} &= \bar{f}_{i,j}^k + Q_x(i, j; \bar{f}^k) + Q_y(i, j; \bar{f}^k) + Q_{xy}(i, j; \bar{f}^k), \\
\hat{f}_{2i,2j-1}^{k+1} &= \bar{f}_{i,j}^k - Q_x(i, j; \bar{f}^k) + Q_y(i, j; \bar{f}^k) - Q_{xy}(i, j; \bar{f}^k), \\
\hat{f}_{2i-1,2j}^{k+1} &= \bar{f}_{i,j}^k + Q_x(i, j; \bar{f}^k) - Q_y(i, j; \bar{f}^k) - Q_{xy}(i, j; \bar{f}^k), \\
\hat{f}_{2i,2j}^{k+1} &= \bar{f}_{i,j}^k - Q_x(i, j; \bar{f}^k) - Q_y(i, j; \bar{f}^k) + Q_{xy}(i, j; \bar{f}^k),
\end{aligned} \tag{2.18}$$

siendo

$$\begin{aligned}
Q_x(i, j; \bar{f}^k) &= \sum_{l=1}^s \beta_l (\bar{f}_{i+l,j}^k - \bar{f}_{i-l,j}^k), \\
Q_y(i, j; \bar{f}^k) &= \sum_{m=1}^s \beta_m (\bar{f}_{i,j+m}^k - \bar{f}_{i,j-m}^k), \\
Q_{xy}(i, j; \bar{f}^k) &= \sum_{l=1}^s \beta_l \sum_{m=1}^s \beta_m (\bar{f}_{i+l,j+m}^k - \bar{f}_{i-l,j+m}^k - \bar{f}_{i+l,j-m}^k + \bar{f}_{i-l,j-m}^k),
\end{aligned} \tag{2.19}$$

y los coeficientes β_i son;

$$\begin{cases} s = 1 \Rightarrow \beta_1 = -\frac{1}{8} \\ s = 2 \Rightarrow \beta_1 = -\frac{22}{128}, \beta_2 = -\frac{3}{128}. \end{cases}$$

En estos casos se obtienen aproximaciones de orden $2s + 1$.

2.5

Interpolación ENO 2D para medias en celda

De igual modo que en el caso unidimensional, es posible realizar una interpolación no lineal eligiendo para cada celda $C_{i,j}^k$ aquel *stencil* que sea más adecuado para interpolar. Así, si consideramos

stencils cuadrados formados por 3×3 celdas, existen 9 posibles *stencils*, $S_{i,j}^k$ que contienen la celda $C_{i,j}^k$ (ver figura 2.4):

$$S_{i,j}^{k;u,v} = \{C_{i-2+n+u,j-2+m+v}^k\}_{0 \leq n,m \leq 2, 0 \leq u,v \leq 2}.$$

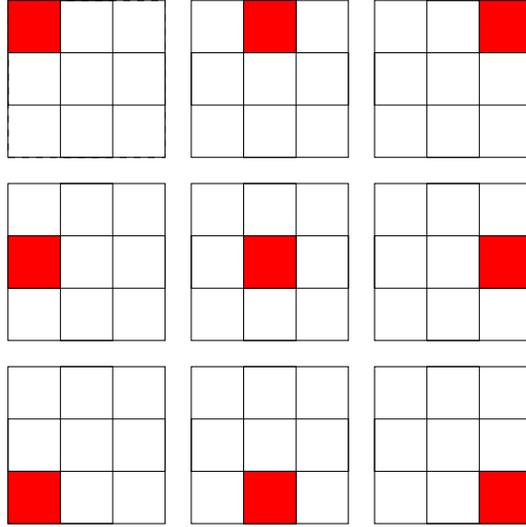


Figura 2.4: Stencils posibles. La celda coloreada representa a $C_{i,j}^k$.

La elección del *stencil* adecuado para la interpolación se realiza a partir de los *stencils* seleccionados de un modo no jerárquico en la dirección horizontal y vertical, pues estos nos determinan un único *stencil* de los nueve anteriores. Una vez el elegido el *stencil*, $S_{i,j}^{ENO}$, se determina el polinomio $p_{i,j}^{ENO}$ que verifica:

$$\frac{1}{h_k^2} \int_{C_{n,m}^k} p_{i,j}^{ENO} dx dy = \bar{f}_{n,m}^k, \quad \forall C_{n,m}^k \in S_{i,j}^{ENO},$$

y definimos las medias de f en el nivel de resolución $k+1$ como las medias del polinomio $p_{i,j}^{ENO}$ en las celdas $C_{2i-1,2j-1}^{k+1}$, $C_{2i,2j-1}^{k+1}$, $C_{2i-1,2j}^{k+1}$ y $C_{2i,2j}^{k+1}$. A continuación mostramos las aproximaciones obtenidas al emplear los diferentes *stencils* 3×3 que contienen a la celda $C_{i,j}^k$.

- *Stencil formado por las celdas* $C_{i-2,j-2}^k, C_{i-1,j-2}^k, C_{i,j-2}^k, C_{i-2,j-1}^k, C_{i-1,j-1}^k, C_{i,j-1}^k, C_{i-2,j}^k, C_{i-1,j}^k$ *y* $C_{i,j}^k$:

$$\begin{aligned}
 \hat{f}_{2i-1,2j-1}^{k+1} &= \frac{25}{64} \bar{f}_{i,j}^k - \frac{5}{64} \bar{f}_{i,j-2}^k + \frac{5}{16} \bar{f}_{i-1,j}^k + \frac{1}{64} \bar{f}_{i-2,j-2}^k - \frac{5}{64} \bar{f}_{i-2,j}^k \\
 &+ \frac{1}{4} \bar{f}_{i-1,j-1}^k + \frac{5}{16} \bar{f}_{i,j-1}^k - \frac{1}{16} \bar{f}_{i-1,j-2}^k - \frac{1}{16} \bar{f}_{i-2,j-1}^k, \\
 \hat{f}_{2i,2j-1}^{k+1} &= \frac{55}{64} \bar{f}_{i,j}^k - \frac{11}{64} \bar{f}_{i,j-2}^k - \frac{5}{16} \bar{f}_{i-1,j}^k - \frac{1}{64} \bar{f}_{i-2,j-2}^k + \frac{5}{64} \bar{f}_{i-2,j}^k \\
 &- \frac{1}{4} \bar{f}_{i-1,j-1}^k + \frac{11}{16} \bar{f}_{i,j-1}^k + \frac{1}{16} \bar{f}_{i-1,j-2}^k + \frac{1}{16} \bar{f}_{i-2,j-1}^k, \\
 \hat{f}_{2i-1,2j}^{k+1} &= \frac{55}{64} \bar{f}_{i,j}^k + \frac{5}{64} \bar{f}_{i,j-2}^k + \frac{11}{16} \bar{f}_{i-1,j}^k - \frac{1}{64} \bar{f}_{i-2,j-2}^k - \frac{11}{64} \bar{f}_{i-2,j}^k \\
 &- \frac{1}{4} \bar{f}_{i-1,j-1}^k - \frac{5}{16} \bar{f}_{i,j-1}^k + \frac{1}{16} \bar{f}_{i-1,j-2}^k + \frac{1}{16} \bar{f}_{i-2,j-1}^k, \\
 \hat{f}_{2i,2j}^{k+1} &= \frac{121}{64} \bar{f}_{i,j}^k + \frac{11}{64} \bar{f}_{i,j-2}^k - \frac{11}{16} \bar{f}_{i-1,j}^k + \frac{1}{64} \bar{f}_{i-2,j-2}^k + \frac{11}{64} \bar{f}_{i-2,j}^k \\
 &+ \frac{1}{4} \bar{f}_{i-1,j-1}^k - \frac{11}{16} \bar{f}_{i,j-1}^k - \frac{1}{16} \bar{f}_{i-1,j-2}^k - \frac{1}{16} \bar{f}_{i-2,j-1}^k.
 \end{aligned}$$

- *Stencil formado por las celdas* $C_{i-2,j-1}^k, C_{i-1,j-1}^k, C_{i,j-1}^k, C_{i-2,j}^k, C_{i-1,j}^k, C_{i,j}^k, C_{i-2,j+1}^k, C_{i-1,j+1}^k$ *y* $C_{i,j+1}^k$:

$$\begin{aligned}
 \hat{f}_{2i-1,2j-1}^{k+1} &= -\frac{5}{64} \bar{f}_{i,j+1}^k + \frac{5}{64} \bar{f}_{i,j-1}^k - \frac{1}{16} \bar{f}_{i-1,j+1}^k - \frac{1}{64} \bar{f}_{i-2,j-1}^k \\
 &+ \frac{1}{64} \bar{f}_{i-2,j+1}^k + \frac{1}{2} \bar{f}_{i-1,j}^k + \frac{5}{8} \bar{f}_{i,j}^k + \frac{1}{16} \bar{f}_{i-1,j}^k - \frac{1}{8} \bar{f}_{i-2,j}^k, \\
 \hat{f}_{2i,2j-1}^{k+1} &= -\frac{11}{64} \bar{f}_{i,j+1}^k + \frac{11}{64} \bar{f}_{i,j-1}^k + \frac{1}{16} \bar{f}_{i-1,j+1}^k + \frac{1}{64} \bar{f}_{i-2,j-1}^k \\
 &- \frac{1}{64} \bar{f}_{i-2,j+1}^k - \frac{1}{2} \bar{f}_{i-1,j}^k + \frac{11}{8} \bar{f}_{i,j}^k - \frac{1}{16} \bar{f}_{i-1,j}^k + \frac{1}{8} \bar{f}_{i-2,j}^k, \\
 \hat{f}_{2i-1,2j}^{k+1} &= \frac{5}{64} \bar{f}_{i,j+1}^k - \frac{5}{64} \bar{f}_{i,j-1}^k + \frac{1}{16} \bar{f}_{i-1,j+1}^k + \frac{1}{64} \bar{f}_{i-2,j-1}^k \\
 &- \frac{1}{64} \bar{f}_{i-2,j+1}^k + \frac{1}{2} \bar{f}_{i-1,j}^k + \frac{5}{8} \bar{f}_{i,j}^k - \frac{1}{16} \bar{f}_{i-1,j}^k - \frac{1}{8} \bar{f}_{i-2,j}^k, \\
 \hat{f}_{2i,2j}^{k+1} &= \frac{11}{64} \bar{f}_{i,j+1}^k - \frac{11}{64} \bar{f}_{i,j-1}^k - \frac{1}{16} \bar{f}_{i-1,j+1}^k - \frac{1}{64} \bar{f}_{i-2,j-1}^k \\
 &+ \frac{1}{64} \bar{f}_{i-2,j+1}^k - \frac{1}{2} \bar{f}_{i-1,j}^k + \frac{11}{8} \bar{f}_{i,j}^k + \frac{1}{16} \bar{f}_{i-1,j}^k + \frac{1}{8} \bar{f}_{i-2,j}^k.
 \end{aligned}$$

- *Stencil formado por las celdas* $C_{i-2,j}^k, C_{i-1,j}^k, C_{i,j}^k, C_{i-2,j+1}^k, C_{i-1,j+1}^k, C_{i,j+1}^k, C_{i-2,j+2}^k, C_{i-1,j+2}^k$ *y* $C_{i,j+2}^k$:

$$\begin{aligned}
\hat{f}_{2i-1,2j-1}^{k+1} &= \frac{5}{64}\bar{f}_{i,j+2}^k + \frac{55}{64}\bar{f}_{i,j}^k + \frac{1}{8}\bar{f}_{i-1,j+2}^k - \frac{11}{64}\bar{f}_{i-2,j}^k - \frac{1}{64}\bar{f}_{i-2,j+2}^k \\
&\quad - \frac{1}{4}\bar{f}_{i-1,j+1}^k - \frac{5}{16}\bar{f}_{i,j+1}^k + \frac{11}{16}\bar{f}_{i-1,j}^k + \frac{1}{16}\bar{f}_{i-2,j+1}^k, \\
\hat{f}_{2i,2j-1}^{k+1} &= -\frac{5}{64}\bar{f}_{i,j+2}^k + \frac{25}{64}\bar{f}_{i,j}^k - \frac{1}{8}\bar{f}_{i-1,j+2}^k - \frac{5}{64}\bar{f}_{i-2,j}^k + \frac{1}{64}\bar{f}_{i-2,j+2}^k \\
&\quad + \frac{1}{4}\bar{f}_{i-1,j+1}^k + \frac{5}{16}\bar{f}_{i,j+1}^k + \frac{5}{16}\bar{f}_{i-1,j}^k - \frac{1}{16}\bar{f}_{i-2,j+1}^k, \\
\hat{f}_{2i-1,2j}^{k+1} &= -\frac{11}{64}\bar{f}_{i,j+2}^k + \frac{55}{64}\bar{f}_{i,j}^k + \frac{1}{8}\bar{f}_{i-1,j+2}^k + \frac{5}{64}\bar{f}_{i-2,j}^k - \frac{1}{64}\bar{f}_{i-2,j+2}^k \\
&\quad - \frac{1}{4}\bar{f}_{i-1,j+1}^k + \frac{11}{16}\bar{f}_{i,j+1}^k - \frac{5}{16}\bar{f}_{i-1,j}^k + \frac{1}{16}\bar{f}_{i-2,j+1}^k, \\
\hat{f}_{2i,2j}^{k+1} &= \frac{11}{64}\bar{f}_{i,j+2}^k + \frac{121}{64}\bar{f}_{i,j}^k - \frac{1}{8}\bar{f}_{i-1,j+2}^k + \frac{11}{64}\bar{f}_{i-2,j}^k + \frac{1}{64}\bar{f}_{i-2,j+2}^k \\
&\quad + \frac{1}{4}\bar{f}_{i-1,j+1}^k - \frac{11}{16}\bar{f}_{i,j+1}^k - \frac{11}{16}\bar{f}_{i-1,j}^k - \frac{1}{16}\bar{f}_{i-2,j+1}^k.
\end{aligned}$$

- *Stencil formado por las celdas* $C_{i-1,j-2}^k, C_{i,j-2}^k, C_{i+1,j-2}^k, C_{i-1,j-1}^k, C_{i,j-1}^k, C_{i+1,j-1}^k, C_{i-1,j}^k, C_{i,j}^k$ *y* $C_{i+1,j}^k$:

$$\begin{aligned}
\hat{f}_{2i-1,2j-1}^k &= -\frac{5}{64}\bar{f}_{i+1,j}^k + \frac{1}{64}\bar{f}_{i+1,j-2}^k + \frac{5}{4}\bar{f}_{i,j}^k - \frac{1}{64}\bar{f}_{i-1,j-2}^k + \frac{5}{64}\bar{f}_{i-1,j}^k \\
&\quad + \frac{1}{2}\bar{f}_{i,j-1}^k - \frac{1}{16}\bar{f}_{i+1,j-1}^k - \frac{1}{8}\bar{f}_{i,j-2}^k + \frac{1}{16}\bar{f}_{i-1,j-1}^k, \\
\hat{f}_{2i,2j-1}^k &= \frac{5}{64}\bar{f}_{i+1,j}^k - \frac{1}{64}\bar{f}_{i+1,j-2}^k + \frac{5}{4}\bar{f}_{i,j}^k + \frac{1}{64}\bar{f}_{i-1,j-2}^k - \frac{5}{64}\bar{f}_{i-1,j}^k \\
&\quad + \frac{1}{2}\bar{f}_{i,j-1}^k + \frac{1}{16}\bar{f}_{i+1,j-1}^k - \frac{1}{8}\bar{f}_{i,j-2}^k - \frac{1}{16}\bar{f}_{i-1,j-1}^k, \\
\hat{f}_{2i-1,2j}^k &= -\frac{11}{64}\bar{f}_{i+1,j}^k - \frac{1}{64}\bar{f}_{i+1,j-2}^k + \frac{11}{4}\bar{f}_{i,j}^k + \frac{1}{64}\bar{f}_{i-1,j-2}^k + \frac{11}{64}\bar{f}_{i-1,j}^k \\
&\quad - \frac{1}{2}\bar{f}_{i,j-1}^k + \frac{1}{16}\bar{f}_{i+1,j-1}^k + \frac{1}{8}\bar{f}_{i,j-2}^k - \frac{1}{16}\bar{f}_{i-1,j-1}^k, \\
\hat{f}_{2i,2j}^k &= \frac{11}{64}\bar{f}_{i+1,j}^k + \frac{1}{64}\bar{f}_{i+1,j-2}^k + \frac{11}{4}\bar{f}_{i,j}^k - \frac{1}{64}\bar{f}_{i-1,j-2}^k - \frac{11}{64}\bar{f}_{i-1,j}^k \\
&\quad - \frac{1}{2}\bar{f}_{i,j-1}^k - \frac{1}{16}\bar{f}_{i+1,j-1}^k + \frac{1}{8}\bar{f}_{i,j-2}^k + \frac{1}{16}\bar{f}_{i-1,j-1}^k.
\end{aligned}$$

- *Stencil formado por las celdas* $C_{i-1,j-1}^k, C_{i,j-1}^k, C_{i+1,j-1}^k, C_{i-1,j}^k, C_{i,j}^k, C_{i+1,j}^k, C_{i-1,j+1}^k, C_{i,j+1}^k, C_{i+1,j+1}^k$ *y* $C_{i+1,j+1}^k$:

$$\begin{aligned}
 \hat{f}_{2i-1,2j-1}^k &= \frac{1}{64} \bar{f}_{i+1,j+1}^k - \frac{1}{64} \bar{f}_{i+1,j-1}^k - \frac{1}{4} \bar{f}_{i,j+1}^k + \frac{1}{64} \bar{f}_{i-1,j-1}^k \\
 &\quad - \frac{1}{64} \bar{f}_{i-1,j+1}^k + \bar{f}_{i,j}^k - \frac{1}{4} \bar{f}_{i+1,j}^k + \frac{1}{4} \bar{f}_{i,j-1}^k + \frac{1}{4} \bar{f}_{i-1,j}^k, \\
 \hat{f}_{2i,2j-1}^k &= -\frac{1}{64} \bar{f}_{i+1,j+1}^k + \frac{1}{64} \bar{f}_{i+1,j-1}^k - \frac{1}{4} \bar{f}_{i,j+1}^k - \frac{1}{64} \bar{f}_{i-1,j-1}^k \\
 &\quad + \frac{1}{64} \bar{f}_{i-1,j+1}^k + \bar{f}_{i,j}^k + \frac{1}{4} \bar{f}_{i+1,j}^k + \frac{1}{4} \bar{f}_{i,j-1}^k - \frac{1}{4} \bar{f}_{i-1,j}^k, \\
 \hat{f}_{2i-1,2j}^k &= -\frac{1}{64} \bar{f}_{i+1,j+1}^k + \frac{1}{64} \bar{f}_{i+1,j-1}^k + \frac{1}{4} \bar{f}_{i,j+1}^k - \frac{1}{64} \bar{f}_{i-1,j-1}^k \\
 &\quad + \frac{1}{64} \bar{f}_{i-1,j+1}^k + \bar{f}_{i,j}^k - \frac{1}{4} \bar{f}_{i+1,j}^k - \frac{1}{4} \bar{f}_{i,j-1}^k + \frac{1}{4} \bar{f}_{i-1,j}^k, \\
 \hat{f}_{2i,2j}^k &= \frac{1}{64} \bar{f}_{i+1,j+1}^k - \frac{1}{64} \bar{f}_{i+1,j-1}^k + \frac{1}{4} \bar{f}_{i,j+1}^k + \frac{1}{64} \bar{f}_{i-1,j-1}^k \\
 &\quad - \frac{1}{64} \bar{f}_{i-1,j+1}^k + \bar{f}_{i,j}^k + \frac{1}{4} \bar{f}_{i+1,j}^k + \frac{1}{4} \bar{f}_{i,j-1}^k - \frac{1}{4} \bar{f}_{i-1,j}^k.
 \end{aligned}$$

- *Stencil formado por las celdas* $C_{i-1,j}^k, C_{i,j}^k, C_{i+1,j}^k, C_{i-1,j+1}^k, C_{i,j+1}^k, C_{i+1,j+1}^k, C_{i-1,j+2}^k, C_{i,j+2}^k$ *y* $C_{i+1,j+2}^k$:

$$\begin{aligned}
 \hat{f}_{2i-1,2j-1}^k &= -\frac{1}{64} \bar{f}_{i+1,j+2}^k - \frac{11}{64} \bar{f}_{i+1,j}^k + \frac{1}{4} \bar{f}_{i,j+2}^k + \frac{11}{64} \bar{f}_{i-1,j}^k + \frac{1}{64} \bar{f}_{i-1,j+2}^k \\
 &\quad - \frac{1}{2} \bar{f}_{i,j+1}^k + \frac{1}{8} \bar{f}_{i+1,j+1}^k + \frac{11}{4} \bar{f}_{i,j}^k - \frac{1}{64} \bar{f}_{i-1,j+1}^k, \\
 \hat{f}_{2i,2j-1}^k &= \frac{1}{64} \bar{f}_{i+1,j+2}^k + \frac{11}{64} \bar{f}_{i+1,j}^k + \frac{1}{4} \bar{f}_{i,j+2}^k - \frac{11}{64} \bar{f}_{i-1,j}^k - \frac{1}{64} \bar{f}_{i-1,j+2}^k \\
 &\quad - \frac{1}{2} \bar{f}_{i,j+1}^k - \frac{1}{8} \bar{f}_{i+1,j+1}^k + \frac{5}{4} \bar{f}_{i,j}^k + \frac{1}{64} \bar{f}_{i-1,j+1}^k, \\
 \hat{f}_{2i-1,2j}^k &= \frac{1}{64} \bar{f}_{i+1,j+2}^k - \frac{5}{64} \bar{f}_{i+1,j}^k - \frac{1}{4} \bar{f}_{i,j+2}^k + \frac{5}{64} \bar{f}_{i-1,j}^k - \frac{1}{64} \bar{f}_{i-1,j+2}^k \\
 &\quad + \frac{1}{2} \bar{f}_{i,j+1}^k - \frac{1}{8} \bar{f}_{i+1,j+1}^k + \frac{5}{4} \bar{f}_{i,j}^k + \frac{1}{64} \bar{f}_{i-1,j+1}^k, \\
 \hat{f}_{2i,2j}^k &= -\frac{1}{64} \bar{f}_{i+1,j+2}^k + \frac{5}{64} \bar{f}_{i+1,j}^k - \frac{1}{4} \bar{f}_{i,j+2}^k - \frac{5}{64} \bar{f}_{i-1,j}^k + \frac{1}{64} \bar{f}_{i-1,j+2}^k \\
 &\quad + \frac{1}{2} \bar{f}_{i,j+1}^k + \frac{1}{8} \bar{f}_{i+1,j+1}^k + \frac{5}{4} \bar{f}_{i,j}^k - \frac{1}{64} \bar{f}_{i-1,j+1}^k,
 \end{aligned}$$

- *Stencil formado por las celdas* $C_{i,j-2}^k, C_{i+1,j-2}^k, C_{i+2,j-2}^k, C_{i,j-1}^k, C_{i+1,j-1}^k, C_{i+2,j-1}^k, C_{i,j}^k, C_{i+1,j}^k$ *y* $C_{i+2,j}^k$:

$$\begin{aligned}
\hat{f}_{2i-1,2j-1}^k &= \frac{5}{64}\bar{f}_{i+2,j}^k - \frac{1}{64}\bar{f}_{i+2,j-2}^k - \frac{5}{8}\bar{f}_{i+1,j}^k - \frac{11}{64}\bar{f}_{i,j-2}^k + \frac{55}{64}\bar{f}_{i,j}^k \\
&\quad - \frac{1}{4}\bar{f}_{i+1,j-1}^k + \frac{1}{16}\bar{f}_{i+2,j-1}^k + \frac{1}{16}\bar{f}_{i+1,j-2}^k + \frac{11}{16}\bar{f}_{i,j-1}^k, \\
\hat{f}_{2i,2j-1}^k &= -\frac{5}{64}\bar{f}_{i+2,j}^k + \frac{1}{64}\bar{f}_{i+2,j-2}^k + \frac{5}{8}\bar{f}_{i+1,j}^k - \frac{5}{64}\bar{f}_{i,j-2}^k + \frac{25}{64}\bar{f}_{i,j}^k \\
&\quad + \frac{1}{4}\bar{f}_{i+1,j-1}^k - \frac{1}{16}\bar{f}_{i+2,j-1}^k - \frac{1}{16}\bar{f}_{i+1,j-2}^k + \frac{5}{16}\bar{f}_{i,j-1}^k, \\
\hat{f}_{2i-1,2j}^k &= \frac{11}{64}\bar{f}_{i+2,j}^k + \frac{1}{64}\bar{f}_{i+2,j-2}^k - \frac{11}{8}\bar{f}_{i+1,j}^k + \frac{11}{64}\bar{f}_{i,j-2}^k + \frac{121}{64}\bar{f}_{i,j}^k \\
&\quad + \frac{1}{4}\bar{f}_{i+1,j-1}^k - \frac{1}{16}\bar{f}_{i+2,j-1}^k - \frac{1}{16}\bar{f}_{i+1,j-2}^k - \frac{11}{16}\bar{f}_{i,j-1}^k, \\
\hat{f}_{2i,2j}^k &= -\frac{11}{64}\bar{f}_{i+2,j}^k - \frac{1}{64}\bar{f}_{i+2,j-2}^k + \frac{11}{8}\bar{f}_{i+1,j}^k + \frac{5}{64}\bar{f}_{i,j-2}^k + \frac{55}{64}\bar{f}_{i,j}^k \\
&\quad - \frac{1}{4}\bar{f}_{i+1,j-1}^k + \frac{1}{16}\bar{f}_{i+2,j-1}^k + \frac{1}{16}\bar{f}_{i+1,j-2}^k - \frac{5}{16}\bar{f}_{i,j-1}^k.
\end{aligned}$$

- *Stencil formado por las celdas* $C_{i,j-1}^k, C_{i+1,j-1}^k, C_{i+2,j-1}^k, C_{i,j}^k, C_{i+1,j}^k, C_{i+2,j}^k, C_{i,j+1}^k, C_{i+1,j+1}^k$ *y* $C_{i+2,j+1}^k$:

$$\begin{aligned}
\hat{f}_{2i-1,2j-1}^k &= -\frac{1}{64}\bar{f}_{i+2,j+1}^k + \frac{1}{64}\bar{f}_{i+2,j-1}^k + \frac{1}{16}\bar{f}_{i+1,j+1}^k + \frac{11}{64}\bar{f}_{i,j-1}^k \\
&\quad - \frac{11}{64}\bar{f}_{i,j+1}^k - \frac{1}{2}\bar{f}_{i+1,j}^k + \frac{1}{8}\bar{f}_{i+2,j}^k - \frac{1}{16}\bar{f}_{i+1,j-1}^k + \frac{11}{8}\bar{f}_{i,j}^k, \\
\hat{f}_{2i,2j-1}^k &= \frac{1}{64}\bar{f}_{i+2,j+1}^k - \frac{1}{64}\bar{f}_{i+2,j-1}^k - \frac{1}{16}\bar{f}_{i+1,j+1}^k + \frac{5}{64}\bar{f}_{i,j-1}^k \\
&\quad - \frac{5}{64}\bar{f}_{i,j+1}^k + \frac{1}{2}\bar{f}_{i+1,j}^k - \frac{1}{8}\bar{f}_{i+2,j}^k + \frac{1}{16}\bar{f}_{i+1,j-1}^k + \frac{5}{8}\bar{f}_{i,j}^k, \\
\hat{f}_{2i-1,2j}^k &= \frac{1}{64}\bar{f}_{i+2,j+1}^k - \frac{1}{64}\bar{f}_{i+2,j-1}^k - \frac{1}{16}\bar{f}_{i+1,j+1}^k - \frac{11}{64}\bar{f}_{i,j-1}^k \\
&\quad + \frac{11}{64}\bar{f}_{i,j+1}^k - \frac{1}{2}\bar{f}_{i+1,j}^k + \frac{1}{8}\bar{f}_{i+2,j}^k + \frac{1}{16}\bar{f}_{i+1,j-1}^k + \frac{11}{8}\bar{f}_{i,j}^k, \\
\hat{f}_{2i,2j}^k &= -\frac{1}{64}\bar{f}_{i+2,j+1}^k + \frac{1}{64}\bar{f}_{i+2,j-1}^k + \frac{1}{16}\bar{f}_{i+1,j+1}^k - \frac{5}{64}\bar{f}_{i,j-1}^k \\
&\quad - \frac{5}{64}\bar{f}_{i,j+1}^k + \frac{1}{2}\bar{f}_{i+1,j}^k - \frac{1}{8}\bar{f}_{i+2,j}^k - \frac{1}{16}\bar{f}_{i+1,j-1}^k + \frac{5}{8}\bar{f}_{i,j}^k.
\end{aligned}$$

- *Stencil formado por las celdas $C_{i,j}^k, C_{i+1,j}^k, C_{i+2,j}^k, C_{i,j+1}^k, C_{i+1,j+1}^k, C_{i+2,j+1}^k, C_{i,j+2}^k, C_{i+1,j+2}^k$ y $C_{i+2,j+2}^k$:*

$$\begin{aligned}
\hat{f}_{2i-1,2j-1}^k &= \frac{1}{64} \bar{f}_{i+2,j+2}^k + \frac{11}{64} \bar{f}_{i+2,j}^k - \frac{1}{16} \bar{f}_{i+1,j+2}^k + \frac{121}{64} \bar{f}_{i,j}^k + \frac{11}{64} \bar{f}_{i,j+2}^k \\
&+ \frac{1}{4} \bar{f}_{i+1,j+1}^k - \frac{1}{4} \bar{f}_{i+2,j+1}^k - \frac{11}{16} \bar{f}_{i+1,j}^k - \frac{11}{16} \bar{f}_{i,j+1}^k, \\
\hat{f}_{2i,2j-1}^k &= -\frac{1}{64} \bar{f}_{i+2,j+2}^k - \frac{11}{64} \bar{f}_{i+2,j}^k + \frac{1}{16} \bar{f}_{i+1,j+2}^k + \frac{55}{64} \bar{f}_{i,j}^k + \frac{5}{64} \bar{f}_{i,j+2}^k \\
&- \frac{1}{4} \bar{f}_{i+1,j+1}^k + \frac{1}{4} \bar{f}_{i+2,j+1}^k + \frac{11}{16} \bar{f}_{i+1,j}^k - \frac{5}{16} \bar{f}_{i,j+1}^k, \\
\hat{f}_{2i-1,2j}^k &= -\frac{1}{64} \bar{f}_{i+2,j+2}^k + \frac{5}{64} \bar{f}_{i+2,j}^k + \frac{1}{16} \bar{f}_{i+1,j+2}^k + \frac{55}{64} \bar{f}_{i,j}^k - \frac{11}{64} \bar{f}_{i,j+2}^k \\
&- \frac{1}{4} \bar{f}_{i+1,j+1}^k + \frac{1}{4} \bar{f}_{i+2,j+1}^k - \frac{5}{16} \bar{f}_{i+1,j}^k + \frac{11}{16} \bar{f}_{i,j+1}^k, \\
\hat{f}_{2i,2j}^k &= \frac{1}{64} \bar{f}_{i+2,j+2}^k - \frac{5}{64} \bar{f}_{i+2,j}^k - \frac{1}{16} \bar{f}_{i+1,j+2}^k + \frac{25}{64} \bar{f}_{i,j}^k - \frac{5}{64} \bar{f}_{i,j+2}^k \\
&+ \frac{1}{4} \bar{f}_{i+1,j+1}^k - \frac{1}{4} \bar{f}_{i+2,j+1}^k + \frac{5}{16} \bar{f}_{i+1,j}^k + \frac{5}{16} \bar{f}_{i,j+1}^k.
\end{aligned}$$

2.6

Resultados numéricos

Veamos los resultados obtenidos al aplicar las siguientes técnicas interpolatorias a una imagen geométrica y una imagen real.

LIN2D : interpolación lineal 2D con *stencils* de 3×3 celdas.

ENO2D : interpolación ENO 2D (ver figura 2.4).

WENO3 : interpolación WENO3 (producto tensorial).

SR : interpolación ENO-Subcell Resolution (producto tensorial).

RAT3 : interpolación racional con dos polinomios de primer grado (producto tensorial).

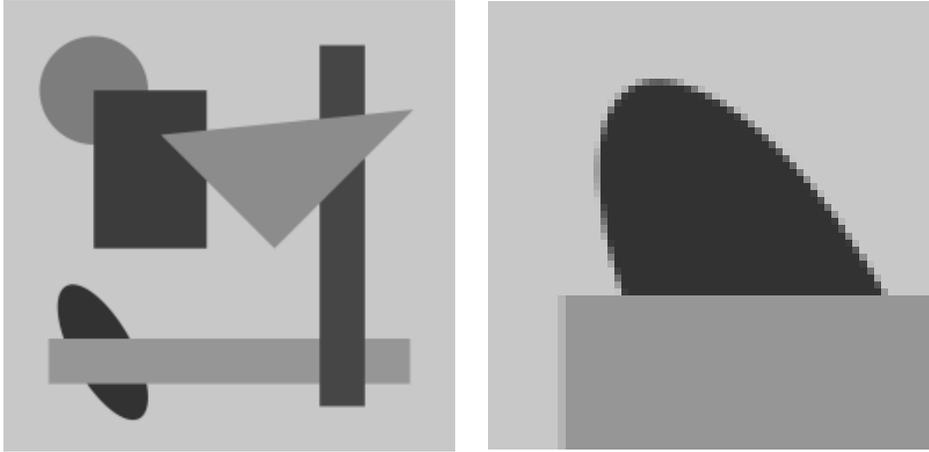


Figura 2.5: Izquierda: Imagen Original. Derecha: Ventana de 64×64 píxeles

RAT5 : interpolación racional con tres polinomios de segundo grado (producto tensorial).

Notar que mientras las técnicas LIN2D, ENO2D son puramente dos-dimensionales, SR, WENO3, RAT3 y RAT5 se han aplicado tensorialmente hasta obtener la imagen resultado.

2.6.1

Ampliación de una imagen geométrica

A partir de una ventana de 64×64 píxeles de la imagen geométrica (Figura 2.5) hemos ampliado hasta obtener una imagen final de 512×512 píxeles (Figura 2.6). La técnica lineal es la que nos ofrece una mayor difusión en los perfiles de la imagen, mientras que con la interpolación ENO-SR obtenemos una mayor definición en éstos. Las técnicas ENO2D, WENO3, RAT3 y RAT5 disminuyen significativamente la difusión presente en los contornos con respecto a LIN2D. Notar además que las ampliaciones conseguidas con WENO3 y RAT3 son similares, mientras que RAT5 consigue unos perfiles más definidos y una disminución del efecto escalera respecto WENO3 y RAT3. Además RAT5 consigue unos resultados más naturales en las zonas *suaves* de la imagen.

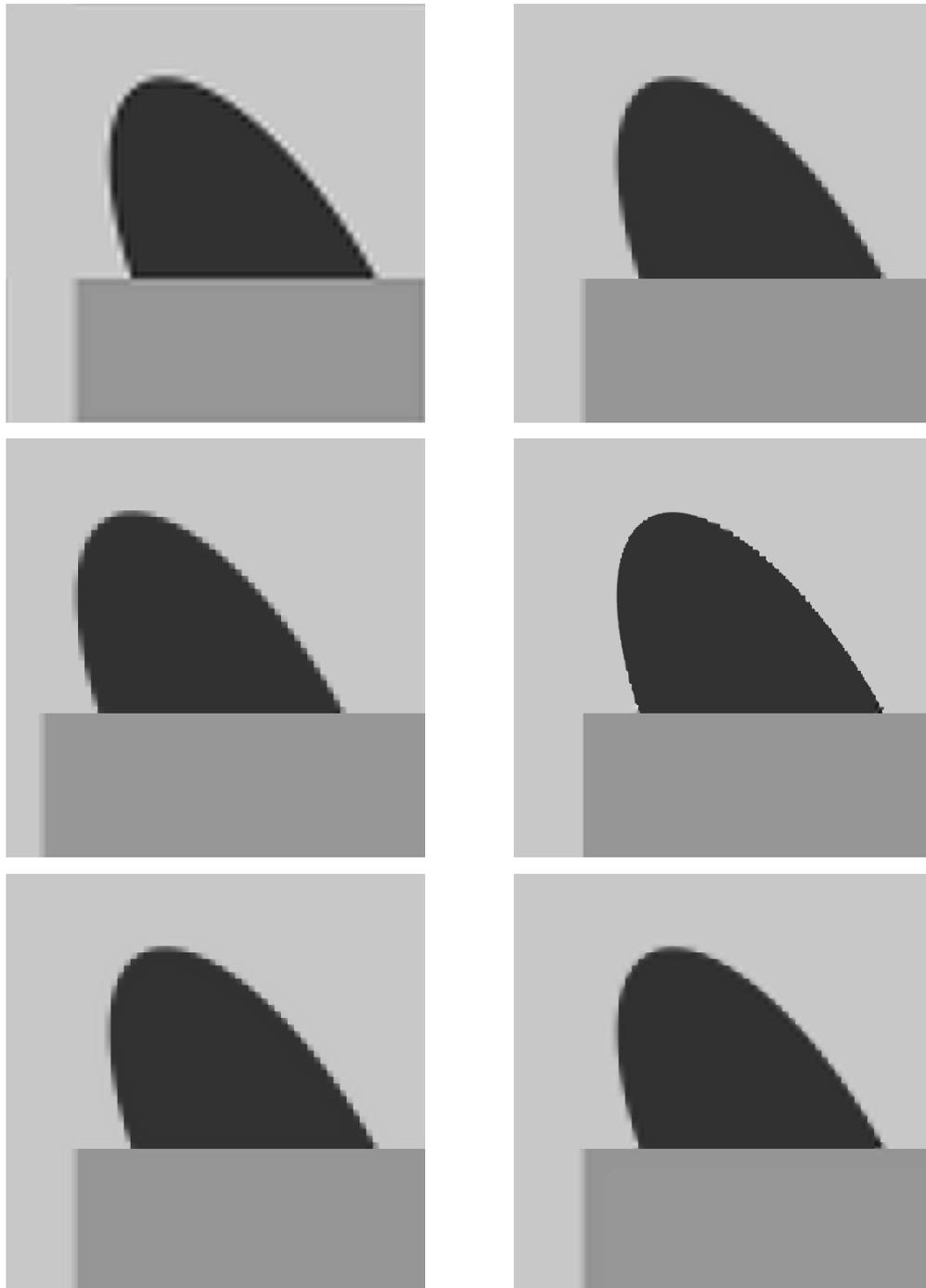


Figura 2.6: Arriba: LIN2D. ENO2D Centro: WENO3. SR Abajo: RAT3. RAT5.

2.6.2

Ampliación de una imagen real

En este caso realizamos el mismo experimento con la imagen de la figura 2.7. De igual modo, las dimensiones de la imagen original y la ventana son 256×256 y 64×64 píxeles respectivamente, ampliando esta última hasta obtener una ventana de 512×512 píxeles.



Figura 2.7: Izquierda: Imagen Original. Derecha: Ventana de 64×64 píxeles

Llegamos a las mismas conclusiones que en el caso de la imagen geométrica.

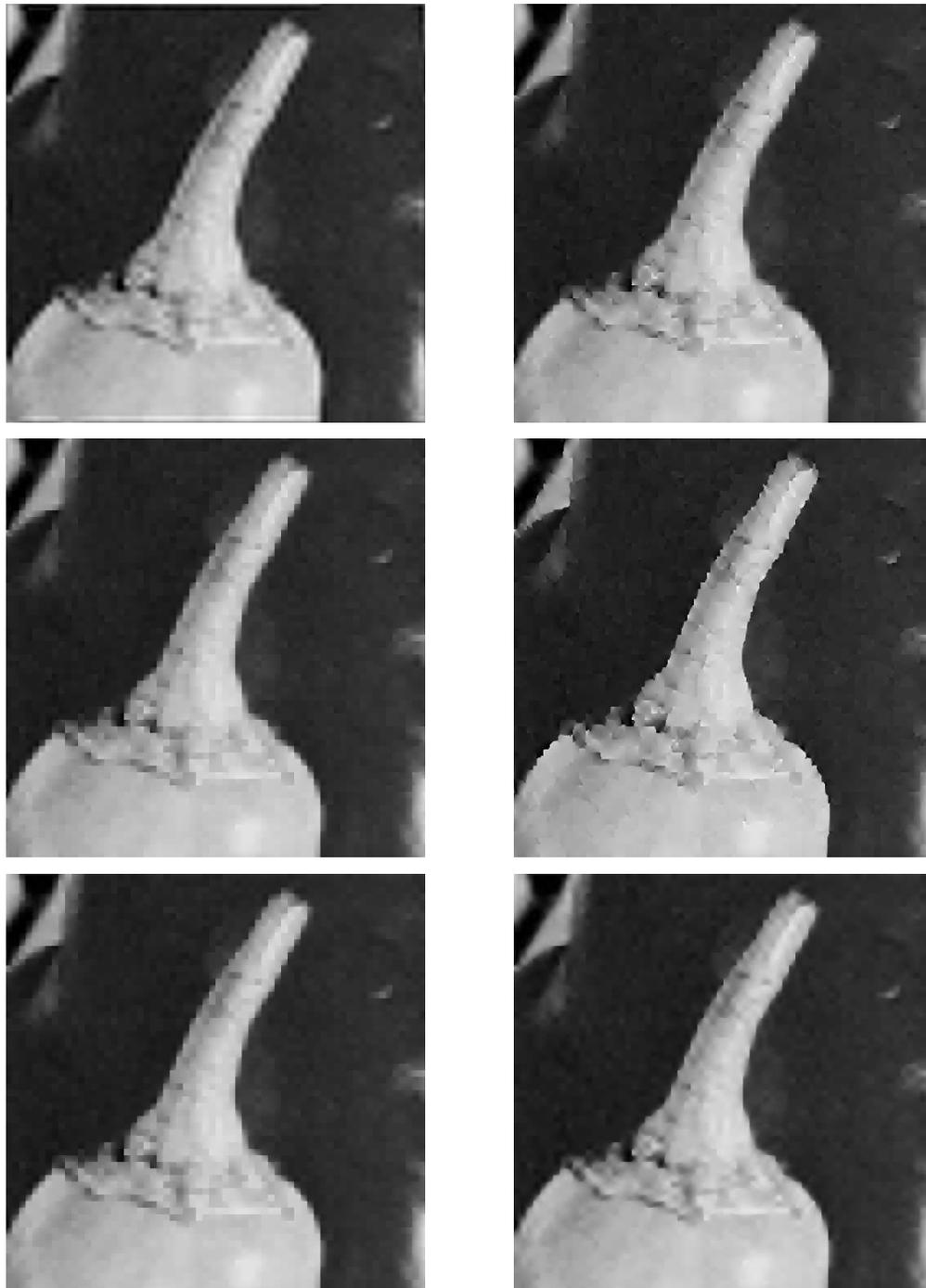


Figura 2.8: Arriba: LIN2D. ENO2D Centro: WENO3. SR. Abajo: RAT3. RAT5.

3

Multirresolución de Harten

3.1

Introducción

La multirresolución de Harten (ver [2, 10]), nos permite obtener una descomposición multiescala de un conjunto de datos discretos, de modo que dicha reordenación tenga unas propiedades mejores que la inicial. En nuestro caso estamos interesados en la mejor compresión de los datos transformados. El resultado obtenido tras aplicar el algoritmo de multirresolución a unos datos dados, 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 los datos iniciales.

Podemos considerar que los datos originales pertenecientes a cierto espacio V^k , son el resultado de aplicar un *operador discretización* a una función f , donde k indica el nivel de resolución (al crecer el valor de k , más resolución). En el contexto de valores puntuales podemos considerar que los datos discretos son los valores puntuales de f en una malla dada.

La multiresolución de Harten está fundamentada en los operadores decimación y predicción que relacionan dos niveles de resolución consecutivos, así el operador decimación, D_k^{k-1} , nos permite pasar del nivel de resolución k al nivel de resolución $k - 1$, mientras que el operador predicción, P_{k-1}^k , nos permite transitar entre el nivel de resolución $k - 1$ y el nivel de resolución k , es decir:

$$D_k^{k-1} : V^k \longrightarrow V^{k-1}, \quad P_{k-1}^k : V^{k-1} \longrightarrow V^k.$$

Los operadores anteriores son definidos a partir del operador discretización antes mencionado, y un operador reconstrucción. Para definir estos dos operadores, consideremos \mathcal{F} un espacio funcional:

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

Definimos el operador discretización, \mathcal{D}_k , de modo que dada $f \in \mathcal{F}$, $\mathcal{D}_k(f) = f^k$ pertenece al espacio V^k , es decir:

$$\mathcal{D}_k : \mathcal{F} \longrightarrow V^k = \mathcal{D}_k(\mathcal{F}).$$

El operador \mathcal{D}_k debe ser un operador lineal y sobreyectivo.

El operador reconstrucción es definido de modo que si $v^k \in V^k$, $\mathcal{R}_k(v^k) \in \mathcal{F}$, i.e.

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

Los operadores decimación y reconstrucción deben verificar una condición de consistencia. Dicho condición no es nada más que la reconstrucción de un conjunto discreto de datos debe contener, exactamente, la misma información que el conjunto inicial de datos, i.e.

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

Notar que el recíproco no es cierto, es decir $\mathcal{R}_k \mathcal{D}_k v^k$ constituye una aproximación a v^k , pero no tiene porque coincidir con v^k .

Dada una secuencia de operadores discretización $\{\mathcal{D}_k\}$, y reconstrucción $\{\mathcal{R}_k\}$, verificando la condición de consistencia (3.1), se definen los operadores decimación y predicción como sigue:

$$D_k^{k-1} = \mathcal{D}_{k-1}\mathcal{R}_k, \quad P_{k-1}^k = \mathcal{D}_k\mathcal{R}_{k-1}. \quad (3.2)$$

La figura 3.1 nos muestra la relación existente entre los operadores discretización, reconstrucción, decimación y predicción.

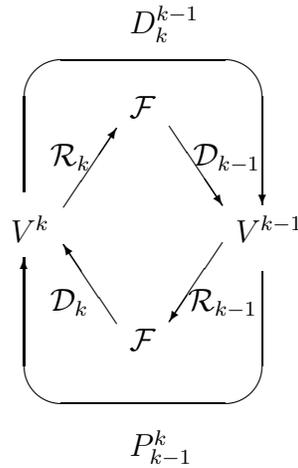


Figura 3.1: Relación entre los diferentes operadores.

Diremos que una sucesión de operadores discretización, $\{\mathcal{D}_k\}$, es *ennidada* si verifica la siguiente relación:

$$\mathcal{D}_k f = 0 \Rightarrow \mathcal{D}_{k-1} f = 0, \quad \forall f \in \mathcal{F}. \quad (3.3)$$

Notar que si $\{\mathcal{D}_k\}$ es ennidada, entonces el operador D_k^{k-1} es independiente respecto del operador reconstrucción empleado. Para demostrar esta afirmación, consideremos $\{\mathcal{R}_k\}$ y $\{\mathcal{R}'_k\}$ dos secuencias de operadores reconstrucción verificando (3.1), entonces se verifica:

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

De este modo si consideramos que $\{\mathcal{D}_k\}$ es ennidada, empleando (3.2) y (3.3), podemos observar que la definición del operador decimación es independiente del operador reconstrucción:

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

Por tanto, si tomamos $\{\mathcal{D}_k\}$ ennidada, el operador decimación será también un operador lineal, independientemente de la linealidad del operador reconstrucción. Además D_k^{k-1} será también un operador sobreyectivo, siendo así mayor la información contenida en un nivel de resolución mayor que en un nivel de resolución inferior.

Haciendo uso de las relaciones (3.1) y (3.2), obtenemos una relación análoga a la condición de consistencia entre los operadores decimación y predicción:

$$D_k^{k-1} P_{k-1}^k = \mathcal{D}_{k-1} \mathcal{R}_k \mathcal{D}_k \mathcal{R}_{k-1} = I_{V^{k-1}}, \quad (3.4)$$

la cual es inmediata a partir de $\mathcal{D}_m \mathcal{R}_k \mathcal{D}_k$ si $m < k$. Para probar esta relación tomemos $f \in \mathcal{F}$, entonces:

$$\mathcal{D}_k \mathcal{R}_k \mathcal{D}_k f = (\mathcal{D}_k \mathcal{R}_k) \mathcal{D}_k f = \mathcal{D}_k f,$$

aprovechando la linealidad de \mathcal{D}_k :

$$\mathcal{D}_k (\mathcal{R}_k \mathcal{D}_k f - f) = 0,$$

y debido al hecho de que \mathcal{D}_k es ennidada llegamos a:

$$\mathcal{D}_{k-1} (\mathcal{R}_k \mathcal{D}_k f - f) = 0 \Rightarrow \mathcal{D}_{k-1} \mathcal{R}_k \mathcal{D}_k f = \mathcal{D}_{k-1} f, \quad (3.5)$$

y empleando (3.5) recursivamente obtenemos la relación que queremos probar:

$$\mathcal{D}_m \mathcal{R}_k \mathcal{D}_k, \quad \forall m < k. \quad (3.6)$$

Sea $v^k \in V^k$, entonces dado que $P_{k-1}^k D_k^{k-1} v^k$ constituye una aproximación a v^k , definimos el error cometido en dicha aproximación como:

$$e^k = v^k - P_{k-1}^k D_k^{k-1} v^k = (I_{V^k} - P_{k-1}^k D_k^{k-1}) v^k =: Q_k v^k \in V^k. \quad (3.7)$$

Así, conocido $v^{k-1} = D_k^{k-1} v^k \in V^{k-1}$ y e^k , podemos recuperar v^k , siendo entonces igual la información que contiene v^k que la información del conjunto $\{v^{k-1}, e^k\}$, i.e.

$$v^k \equiv \{v^{k-1}, e^k\},$$

pues haciendo uso de la expresión de v^k es inmediata la siguiente relación $v^k = P_{k-1}^k v^{k-1} + e^k$.

Notar que el conjunto $\{v^{k-1}, e^k\}$ posee información redundante, pues si $N_k = \dim V^k$, resulta que $\{v^{k-1}, e^k\}$ consta de $N_{k-1} + N_k$ elementos, teniendo $\{v^{k-1}, e^k\}$ y v^k la misma información. Para eliminar la información redundante de $\{v^{k-1}, e^k\}$, aprovechamos el hecho de que e^k pertenece al núcleo del operador D_k^{k-1} , ya que:

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

Por tanto, $e^k \in \mathcal{N}(D_k^{k-1}) = \{v^k \in V^k : D_k^{k-1}v^k = 0\}$. Sea $\{\mu_i^k\}$ una base del núcleo de D_k^{k-1} cuya dimensión es $\dim V^k - \dim V^{k-1} = N_k - N_{k-1}$, y expresemos e^k en función de esta base:

$$e^k = \sum_i d_i^k \mu_i^k.$$

Sea G_k el operador que dado $e^k \in \mathcal{N}(D_k^{k-1})$ le asocia sus coordenadas $\{d_i^k\}$ correspondientes a la base $\{\mu_i^k\}$, y sea E_k el operador que dada una serie de coeficientes $\{d_i^k\}$ le asocie $\sum_i d_i^k \mu_i^k$. Tenemos entonces la siguiente relación:

$$v^k \equiv \{v^{k-1}, d^k\},$$

donde, ahora sí, ambos conjuntos tienen igual cantidad de elementos pues el número de elementos de $\{v^{k-1}, d^k\}$ es igual a:

$$\dim V^{k-1} + \dim \mathcal{N}(D_k^{k-1}) = N_{k-1} + (N_k - N_{k-1}) = N_k = \dim V^k.$$

Notar que podemos pasar de v^k a $\{v^{k-1}, d^k\}$ mediante las siguientes expresiones:

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

y viceversa

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

relación que es inmediata a partir de $e^k = E_k d^k$.

Si reiteramos el proceso anterior en el que hemos establecido la equivalencia $v^k \equiv \{v^{k-1}, d^k\}$, con los distintos niveles de resolución v^n , $n < k$, obtenemos la descomposición multiescala de v^k . Así si

consideramos que los datos discretos originales están en un nivel de resolución L , podemos llegar a establecer la siguiente equivalencia:

$$v^L \equiv \{v^0, d^1, d^2, \dots, d^L\}.$$

$$\begin{array}{ccccccc} v^L & \rightarrow & v^{L-1} & \rightarrow & v^{L-2} & \rightarrow & \dots \\ & & \searrow & & \searrow & & \searrow \\ & & d^L & & d^{L-1} & & \dots \end{array}$$

Figura 3.2: Descomposición multiescala.

Los algoritmos para obtener la transformación multiescala y su transformación inversa son los siguientes:

Algoritmo. 3.1. Transformación directa

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

Algoritmo. 3.2. Transformación Inversa

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

A partir de $v^k = \mathcal{D}_k f$, y considerando $\{\mathcal{D}_k\}$ una secuencia de operadores discretización ennidada, podemos expresar el error de predicción (3.7) a partir de los operadores discretización y predicción:

$$\begin{aligned} e^k &= (I - P_{k-1}^k D_k^{k-1})v^k = (I - \mathcal{D}_k \mathcal{R}_{k-1} \mathcal{D}_{k-1} \mathcal{R}_k) \mathcal{D}_k f \\ &= (\mathcal{D}_k - \mathcal{D}_k \mathcal{R}_{k-1} \mathcal{D}_{k-1} \mathcal{R}_k \mathcal{D}_k) f = (\mathcal{D}_k - \mathcal{D}_k \mathcal{R}_{k-1} \mathcal{D}_{k-1}) f \\ &= \mathcal{D}_k (I - \mathcal{R}_{k-1} \mathcal{D}_{k-1}) f. \end{aligned}$$

La relación anterior nos muestra que el problema de encontrar un buen operador predicción para la multirresolución de Harten, se puede plantear como conocida $\mathcal{D}_{k-1}f = v^{k-1}$, $v \in \mathcal{F}$, encontrar una “buena aproximación” a $\mathcal{D}_k f$, siendo este último un problema de teoría de la aproximación.

3.2

Multirresolución de Harten para valores puntuales

Consideremos el espacio funcional de las funciones acotadas definidas sobre un dominio acotado Ω , i.e. $\mathcal{F} = \mathcal{B}(\Omega)$. Sea $X^k = \{x_i^k\}_{i=0}^{J_k}$, $x_i^k \in \Omega$, y definamos $f^k = \mathcal{D}_k f$ del modo siguiente:

$$f_i^k = (\mathcal{D}_k f)_i = f(x_i^k), \quad f^k = \{f_i^k\}_{i=0}^{J_k}. \quad (3.8)$$

Así definida, f_i^k es una *discretización por valores puntuales*. Notar que en este caso el espacio V^k es un espacio de dimensión finita, con $\dim V^k = N_k = J_k + 1$.

La sucesión de los operadores decimación, $\{\mathcal{D}_k\}$ es ennidada, ver (3.3), si y solo si $X^{k-1} \subset X^k$, $\forall k$. En este caso la decimación consiste en eliminar de f^k las componentes $f_i^k = f(x_i^k)$ tales que $x_i^k \notin X^{k-1}$, y por tanto su definición es independiente del operador reconstrucción.

El operador reconstrucción será un operador \mathcal{R}_k verificando:

$$\mathcal{R}_k : V^k \longrightarrow \mathcal{B}(\Omega), \quad \mathcal{D}_k \mathcal{R}_k f^k = f^k,$$

es decir, $(\mathcal{R}_k f^k)(x_i^k) = f_i^k = f(x_i^k)$. Por tanto, $\mathcal{R}_k f^k(x)$ debe pertenecer al espacio $\mathcal{B}(\Omega)$ e interpolar a la función f en los nodos $\{x_i^k\}$. De este modo, podemos emplear como operador reconstrucción una técnica interpolatoria. Si definimos

$$(\mathcal{R}_k f^k)(x) := \mathcal{I}(x; f^k).$$

a partir de (3.2), tenemos:

$$(\mathcal{P}_{k-1}^k f^{k-1})_i = (\mathcal{D}_k \mathcal{I}(\cdot; f^{k-1}))_i = \mathcal{I}(x_i^k; f^{k-1}).$$

Para los valores $x_i^k \in X^{k-1}$, la predicción consiste en tomar el valor de la función en dichos nodos, mientras que para los valores $x_i^k \notin X^{k-1}$ se calculará el valor interpolado.

El error de predicción $e_i^k = f_i^k - \mathcal{I}(x_i^k; f^{k-1})$ es igual a 0 para todo $x_i^k \in X^{k-1}$. De este modo se definen los coeficientes $d^k = \{d_j^k\}$ como el error de interpolación en los puntos $x_j^k \in X^k - X^{k-1}$.

3.2.1

Multirresolución en $[0, 1]$

Consideremos ahora el caso particular en el que los datos discretos son los valores que toma una función f perteneciente al espacio funcional $\mathcal{B}[0, 1]$, i.e. el espacio de las funciones acotadas en $I = [0, 1]$. Sea X^L una discretización uniforme de I :

$$X^L = \{x_i^L\}_{i=0}^{J_L}, \quad x_i^L = i \cdot J_L, \quad h_L = \frac{1}{J_L}, \quad J_L = 2^L J_0, \quad (3.9)$$

donde L indica el nivel de resolución y J_0 es un número entero dado. Las discretizaciones correspondientes a niveles de resolución inferior, $X^k = \{x_i^k\}_{i=0}^{J_k}$, $k = 0, \dots, L - 1$ son definidas del siguiente modo:

$$x_i^{k-1} = x_{2i}^k, \quad i = 0, \dots, J_{k-1} = J_k/2. \quad (3.10)$$

Así la discretización del nivel de resolución $k - 1$ está formada por los nodos pares del nivel de resolución k (ver figura 3.3).

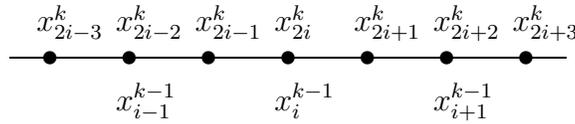


Figura 3.3: Discretizaciones X^k y X^{k-1}

Definimos entonces el operador decimación en el contexto de valores puntuales en el intervalo $[0, 1]$ como

$$f_i^{k-1} = (D_k^{k-1} f^k)_i = f_{2i}^k, \quad 0 \leq i \leq J_k \quad (3.11)$$

Notar que de este modo el operador predicción debe verificar $(P_{k-1}^k f^{k-1})_{2i} = f_{2i}^k$, y por tanto los operadores decimación y predicción satisfacen la condición de consistencia (3.4). Teniendo en cuenta lo anterior, se comprueba fácilmente que el error cometido en los nodos pares, x_{2i}^k , es nulo, i.e.,

$$e_{2i}^k = f_{2i}^k - (P_{k-1}^k f^{k-1})_i = 0.$$

Así, los algoritmos de la transformación directa e inversa para el contexto de los valores puntuales en el intervalo $[0, 1]$ son los siguientes:

Algoritmo. 3.3. *Transformación Directa (valores puntuales en $[0, 1]$)*

$$f^L \rightarrow (f^0, d^1, \dots, d^L)$$

for $k = L, \dots, 1$

$$\begin{aligned} f_i^{k-1} &= f_{2i}^k, & 0 \leq i \leq J_{k-1} \\ d_i^k &= f_{2i-1}^k - \mathcal{I}(x_{2i-1}^k; f^{k-1}), & 1 \leq i \leq J_{k-1} \end{aligned}$$

end

Algoritmo. 3.4. *Transformación Inversa (valores puntuales en $[0, 1]$)*

$$(f^0, d^1, \dots, d^L) \rightarrow f^L$$

for $k = 1, \dots, L$

$$\begin{aligned} f_{2i}^k &= f_i^{k-1}, & 0 \leq i \leq J_{k-1} \\ f_{2i-1}^k &= \mathcal{I}(x_{2i-1}^k; f^{k-1}) + d_i^k, & 1 \leq i \leq J_{k-1} \end{aligned}$$

end

En los algoritmos anteriores, el operador \mathcal{I} puede ser cualquier operador interpolación. Si empleamos un interpolante polinómico a trozos de grado r , dado que los coeficientes d^k son los errores de interpolación cometidos en los nodos impares del nivel de resolución k , tendremos que $d_i^k = O(h_k^{r+1})$ en aquellas zonas donde

la función sea lo suficientemente suave. En caso de que el *stencil* empleado cruce alguna discontinuidad de la derivada p -ésima tendremos entonces que $d_i^k = O(h_k^p)$, si $p \leq r$. Así en presencia de singularidades, tal y como vimos en la sección 1.1, el número de nodos donde la predicción no es óptima, y por tanto el número de coeficientes d_i^k de una magnitud mayor, puede ser reducido evitando tomar *stencils* conteniendo singularidades, lo cual es posible en ciertos casos empleando técnicas interpolatorias no lineales.

3.2.2

Multirresolución en $[0, 1] \times [0, 1]$

En el caso dos-dimensional, los datos discretos $\{f_{i,j}^L\}_{i,j=0}^{J_L}$, representan los valores que toma una función $f : [0, 1] \times [0, 1] \rightarrow \mathbb{R}$ en la partición uniforme $X^L = \{(x_i^L, x_j^L)\}_{i,j=0}^{J_L}$, donde $x_i^L = i \cdot h_L$, $y_j^L = j \cdot h_L$, $i, j = 0, \dots, J_L$, $h_L = 1/J_L$, donde L indica el nivel de resolución. De forma análoga al caso de una dimensión, definimos las mallas correspondiente al nivel $k - 1$ a partir de los nodos de índice par-par correspondientes al nivel de resolución k (ver figura 3.4):

$$X^{k-1} = \{(x_i^{k-1}, y_j^{k-1})\}_{i,j=0}^{J_{k-1}}, \quad J_{k-1} = \frac{J_k}{2} \quad (x_i^{k-1}, y_j^{k-1}) = (x_{2i}^k, y_{2j}^k).$$

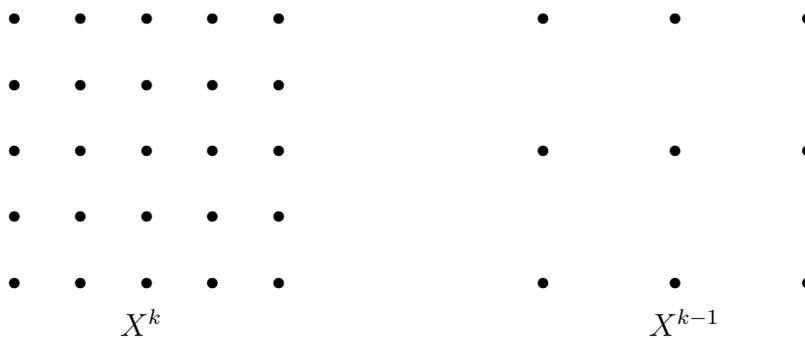


Figura 3.4: Mallas X^k y X^{k-1}

El operador decimación se define en el caso dos-dimensional de

la siguiente forma:

$$f_{i,j}^{k-1} = (D_k^{k-1} f)_{i,j} = f_{2i,2j}^k, \quad 0 \leq i, j \leq J_{k-1}.$$

Con el objetivo de que los operadores decimación y predicción sean consistentes, i.e. verifiquen (3.4), el operador predicción debe satisfacer

$$(P_{k-1}^k f)_{2i,2j} = f_{i,j}^{k-1}, \quad 0 \leq i, j \leq J_{k-1}.$$

Así, conocidos los errores cometidos por el operador predicción en aquellos nodos que no están presentes en dos niveles de resolución consecutivos, obtenemos una biyección entre los datos discretos en el nivel de resolución k y los datos en el nivel de resolución $k-1$ más los errores correspondientes. Sean entonces $d_{i,j;1}^k$, $d_{i,j;2}^k$ y $d_{i,j;3}^k$ dichos errores,

$$\begin{aligned} d_{i,j;1}^k &= f_{2i+1,2j}^k - (P_{k-1}^k f)_{2i+1,2j}, & 0 \leq i < J_{k-1}, \quad 0 \leq j \leq J_{k-1}, \\ d_{i,j;2}^k &= f_{2i,2j+1}^k - (P_{k-1}^k f)_{2i,2j+1}, & 0 \leq i \leq J_{k-1}, \quad 0 \leq j < J_{k-1}, \\ d_{i,j;3}^k &= f_{2i+1,2j+1}^k - (P_{k-1}^k f)_{2i+1,2j+1}, & 0 \leq i < J_{k-1}, \quad 0 \leq j < J_{k-1}. \end{aligned}$$

Empleando entonces f^{k-1} y $d^k = (d_1^k, d_2^k, d_3^k)$ es inmediato establecer la equivalencia $\{f^k\} \equiv \{f^{k-1}, d^k\}$. Si consideramos que los datos originales están en un nivel de resolución L , y reiteramos el proceso anterior hasta un nivel de resolución e , $e < L$, obtenemos

$$\{f^L\} \equiv \{f^e, d^{e+1}, \dots, d^L\}.$$

El algoritmo 3.5 muestra como obtener la descomposición multiescala de unos datos discretos, y el algoritmo 3.6 muestra como recuperar los datos originales a partir de la descomposición multiescala de éstos.

Algoritmo. 3.5. *Transformación Directa (valores puntuales en $[0, 1] \times [0, 1]$)* $f^L \rightarrow (f^e, d^{e+1}, \dots, d^L)$

for $k = L, \dots, e + 1$
for $i, j = 0, \dots, J_{k-1}$

```

 $f_{2i,2j}^{k-1} = f_{i,j}^k$ 
end
for  $i, j = 0, \dots, J_{k-1} - 1$ 
 $d_{i,j;1}^k = f_{2i+1,2j}^k - (P_{k-1}^k f^{k-1})_{2i+1,2j}$ 
 $d_{i,j;2}^k = f_{2i,2j+1}^k - (P_{k-1}^k f^{k-1})_{2i,2j+1}$ 
 $d_{i,j;3}^k = f_{2i+1,2j+1}^k - (P_{k-1}^k f^{k-1})_{2i+1,2j+1}$ 
end
for  $l = 0, \dots, J_{k-1} - 1$ 
 $d_{l,2^{k-1};1}^k = f_{2l+1,2^{k-1}}^k - (P_{k-1}^k f^{k-1})_{2l+1,2^{k-1}}$ 
 $d_{2^{k-1},l;2}^k = f_{2^{k-1},2l+1}^k - (P_{k-1}^k f^{k-1})_{2^{k-1},2l+1}$ 
end

```

Algoritmo. 3.6. *Transformación Inversa (valores puntuales en $[0, 1] \times [0, 1]$)* $(f^e, d^{e+1}, \dots, d^L) \rightarrow f^L$

```

for  $k = e + 1, \dots, L$ 
for  $i, j = 0, \dots, J_{k-1}$ 
 $f_{2i,2j}^k = f_{i,j}^{k-1}$ 
end
for  $i, j = 0, \dots, J_{k-1} - 1$ 
 $f_{2i+1,2j}^k = d_{i,j;1}^k + (P_{k-1}^k f^{k-1})_{2i+1,2j}$ 
 $f_{2i,2j+1}^k = d_{i,j;2}^k + (P_{k-1}^k f^{k-1})_{2i,2j+1}$ 
 $f_{2i+1,2j+1}^k = d_{i,j;3}^k + (P_{k-1}^k f^{k-1})_{2i+1,2j+1}$ 
end
for  $l = 0, \dots, J_{k-1} - 1$ 
 $f_{l,2^{k-1}}^k = d_{l,2^{k-1};1}^k + (P_{k-1}^k f^{k-1})_{2l+1,2^{k-1}}$ 
 $f_{2^{k-1},l}^k = d_{2^{k-1},l;2}^k + (P_{k-1}^k f^{k-1})_{2^{k-1},2l+1}$ 
end

```

Una de las aplicaciones de la multirresolución es la compresión de datos. Así, si a los datos transformados $\{f^e, d^{e+1}, \dots, d^L\}$ los modificamos ligeramente $\{\hat{f}^e, \hat{d}^{e+1}, \dots, \hat{d}^L\}$, podemos esperar que al aplicar el algoritmo de multirresolución inverso a éstos últimos obtengamos una aproximación f^L . A. Harten con el objetivo de asegurar la estabilidad de los algoritmos de multirresolución al

emplear operadores de predicción no lineales introdujo la técnica de control del error (EC), ver [29], en 1D, la cual nos permite establecer una cota *a priori* del error cometido en función del error que cometamos al modificar los coeficientes. Más detalles acerca de la técnica de control del error en 1D se pueden consultar en [1, 10, 29].

Entre las modificaciones que podemos aplicar a los datos para facilitar su compresión destacamos dos operaciones, el truncamiento y la cuantización. El truncamiento consiste en dado un número $\varepsilon > 0$, entonces

$$\text{tr}[x; \varepsilon] = \begin{cases} 0, & \text{si } |x| \leq \varepsilon \\ x, & \text{si } |x| > \varepsilon, \end{cases} \quad (3.12)$$

es decir, al truncar un número con parámetro ε éste será codificado como 0 si su valor absoluto es menor o igual que el parámetro empleado. Si por el contrario cuantizamos los datos con un parámetro ε entonces,

$$\text{qu}[x; \varepsilon] = 2\varepsilon \text{round}\left(\frac{x}{2\varepsilon}\right), \quad (3.13)$$

donde $\text{round}[\cdot]$ es el entero obtenido por redondeo. De este modo si cuantizamos con un parámetro $\varepsilon = 0,5$ las datos codificados serán números enteros, mientras que si $\varepsilon = 1, 2, 3 \dots$ estaremos codificando múltiplos enteros de 2, 4, 6...

El algoritmo 3.7 muestra el algoritmo de control del error en 2D. En este caso en el nivel de resolución k se ha empleado el parámetro de cuantización ε_k .

Algoritmo. 3.7. *Transformación Directa EC*

(valores puntuales en $[0, 1] \times [0, 1]$)

$f^L \rightarrow (\hat{f}^e, \hat{d}^{e+1}, \dots, \hat{d}^L)$

for $k = L, \dots, e + 1$

for $i, j = 0, \dots, 2^{k-1}$

$f_{2i, 2j}^{k-1} = f_{i, j}^k$

end

end

$\hat{f}^e = \text{qu}(f^e, \varepsilon_e)$

```

for  $k = e + 1, \dots, L$ 
  for  $i = 0, \dots, 2^{k-1} - 1, j = 0, \dots, 2^{k-1}$ 
     $d_{i,j;1}^k = f_{2i+1,2j}^k - (P_{k-1}^k \hat{f}^{k-1})_{2i+1,2j}$ 
  end
   $\hat{d}_1 = \mathbf{qu}(d_1, \varepsilon_k)$ 
  for  $i = 0, \dots, 2^{k-1} - 1, j = 0, \dots, 2^{k-1}$ 
     $\hat{f}_{2i+1,2j}^k = \hat{d}_{i,j;1}^k + (P_{k-1}^k \hat{f}^{k-1})_{2i+1,2j}$ 
  end
  for  $i = 0, \dots, 2^{k-1}, j = 0, \dots, 2^{k-1} - 1$ 
     $d_{i,j;2}^k = f_{2i,2j+1}^k - (P_{k-1}^k \hat{f}^{k-1})_{2i,2j+1}$ 
  end
   $\hat{d}_2 = \mathbf{qu}(d_2, \varepsilon_k)$ 
  for  $i = 0, \dots, 2^{k-1}, j = 0, \dots, 2^{k-1} - 1$ 
     $\hat{f}_{2i,2j+1}^k = \hat{d}_{i,j;2}^k + (P_{k-1}^k \hat{f}^{k-1})_{2i,2j+1}$ 
  end
  for  $i = 0, \dots, 2^{k-1} - 1, j = 0, \dots, 2^{k-1} - 1$ 
     $d_{i,j;3}^k = f_{2i+1,2j+1}^k - (P_{k-1}^k \hat{f}^{k-1})_{2i+1,2j+1}$ 
  end
   $\hat{d}_3 = \mathbf{qu}(d_3, \varepsilon_k)$ 
  for  $i = 0, \dots, 2^{k-1} - 1, j = 0, \dots, 2^{k-1} - 1$ 
     $\hat{f}_{2i+1,2j+1}^k = \hat{d}_{i,j;3}^k + (P_{k-1}^k \hat{f}^{k-1})_{2i+1,2j+1}$ 
  end
  for  $i, j = 0, \dots, 2^{k-1}$ 
     $\hat{f}_{2i,2j}^k = \hat{f}_{i,j}^{k-1}$ 
  end
end

```

Procediendo de forma análoga al caso 1D, [1, 10, 29], se obtienen las siguientes cotas del error:

$$\|f^L - \hat{f}^L\|_p \leq \varepsilon = \max \varepsilon_k, \quad p = 1, 2, \infty, \quad (3.14)$$

3.3

Modificación del algoritmo de control del error

A continuación proponemos una modificación del algoritmo de control del error en 2D (algoritmo 3.7). Los resultados obtenidos han sido presentados en el congreso Cedyá 2005 [8]. Dicha modificación consiste en emplear la información obtenida tras corregir la predicción de los nodos (x_{2i+1}^k, y_{2j}^k) , (x_{2i}^k, y_{2j+1}^k) al sumarle los detalles correspondientes, para así obtener una mejor aproximación a $f_{2i+1,2j+1}^k$. Así para calcular $(P_{k-1}^k f)_{2i+1,2j+1}$ consideraremos los valores $\{f_{i,j}^{k-1}\}$, $\{\hat{f}_{2i+1,2j}^k\}$ y $\{\hat{f}_{2i,2j+1}^k\}$.

Algoritmo. 3.8. *Modificación codificación algoritmo EC*

(valores puntuales en $[0, 1] \times [0, 1]$)

$f^L \rightarrow (\hat{f}^e, \hat{d}^{e+1}, \dots, \hat{d}^L)$

```

for  $k = L, \dots, e + 1$ 
  for  $i, j = 0, \dots, 2^{k-1}$ 
     $f_{2i,2j}^{k-1} = f_{i,j}^k$ 
  end
end
 $\hat{f}^e = \mathbf{qu}(f^e, \varepsilon_e)$ 
for  $k = e + 1, \dots, L$ 
  for  $i = 0, \dots, 2^{k-1} - 1, j = 0, \dots, 2^{k-1}$ 
     $d_{i,j;1}^k = f_{2i+1,2j}^k - (P_{k-1}^k \hat{f}^{k-1})_{2i+1,2j}$ 
  end
  for  $i = 0, \dots, 2^{k-1}, j = 0, \dots, 2^{k-1} - 1$ 
     $d_{i,j;2}^k = f_{2i,2j+1}^k - (P_{k-1}^k \hat{f}^{k-1})_{2i,2j+1}$ 
  end
   $\hat{d}_1 = \mathbf{qu}(d_1, \varepsilon_k), \hat{d}_2 = \mathbf{qu}(d_2, \varepsilon_k)$ 
  for  $i = 0, \dots, 2^{k-1} - 1, j = 0, \dots, 2^{k-1}$ 
     $\hat{f}_{2i+1,2j}^k = \hat{d}_{i,j;1}^k + (P_{k-1}^k \hat{f}^{k-1})_{2i+1,2j}$ 
  end
for  $i = 0, \dots, 2^{k-1}, j = 0, \dots, 2^{k-1} - 1$ 

```

```

       $\hat{f}_{2i,2j+1}^k = \hat{d}_{i,j;2}^k + (P_{k-1}^k \hat{f}^{k-1})_{2i,2j+1}$ 
end
for  $i = 0, \dots, 2^{k-1} - 1, j = 0, \dots, 2^{k-1} - 1$ 
       $d_{i,j;3}^k = f_{2i+1,2j+1}^k - (\hat{P}_{k-1}^k \hat{f}^{k-1})_{2i+1,2j+1}$ 
end
       $\hat{d}_3 = \mathbf{qu}(d_3, \varepsilon_k)$ 
for  $i = 0, \dots, 2^{k-1} - 1, j = 0, \dots, 2^{k-1} - 1$ 
       $\hat{f}_{2i+1,2j+1}^k = \hat{d}_{i,j;3}^k + (\hat{P}_{k-1}^k \hat{f}^{k-1})_{2i+1,2j+1}$ 
end
for  $i, j = 0, \dots, 2^{k-1}$ 
       $\hat{f}_{2i,2j}^k = \hat{f}_{i,j}^{k-1}$ 
end
end

```

Notar que en algoritmo 3.8, el operador \hat{P}_{k-1}^k , no es nada más que un operador predicción actuando sobre $\{f_{i,j}^{k-1}\}$, $\{\hat{f}_{2i+1,2j}^k\}$ y $\{\hat{f}_{2i,2j+1}^k\}$. En nuestro caso hemos utilizado

$$(\hat{P}_{k-1}^k f^{k-1})_{2i+1,2j+1} = \frac{1}{4}(\hat{f}_{2i+1,2j}^k + \hat{f}_{2i,2j+1}^k + \hat{f}_{2i+1,2j+2}^k + \hat{f}_{2i+2,2j+1}^k), \quad (3.15)$$

ya que en caso de emplear polinomios interpoladores con *stencils* 2×2 tendríamos,

$$(P_{k-1}^k f^{k-1})_{2i+1,2j+1} = \frac{1}{4}(f_{i,j}^{k-1} + f_{i+1,j}^{k-1} + f_{i,j+1}^{k-1} + f_{i+1,j+1}^{k-1}),$$

y si tenemos en cuenta las relaciones siguientes:

$$\begin{aligned} (P_{k-1}^k f^{k-1})_{2i+1,2j} &= \frac{1}{2}(f_{i,j}^{k-1} + f_{i+1,j}^{k-1}), \\ (P_{k-1}^k f^{k-1})_{2i,2j+1} &= \frac{1}{2}(f_{i,j}^{k-1} + f_{i,j+1}^{k-1}), \\ (P_{k-1}^k f^{k-1})_{2i+2,2j+1} &= \frac{1}{2}(f_{i+1,j}^{k-1} + f_{i+1,j+1}^{k-1}), \\ (P_{k-1}^k f^{k-1})_{2i+1,2j+2} &= \frac{1}{2}(f_{i,j+1}^{k-1} + f_{i+1,j+1}^{k-1}), \end{aligned} \quad (3.16)$$

tenemos

$$\begin{aligned} (P_{k-1}^k f^{k-1})_{2i+1,2j+1} &= \frac{1}{4} \left((P_{k-1}^k f^{k-1})_{2i+1,2j} + (P_{k-1}^k f^{k-1})_{2i,2j+1} + \right. \\ &\quad \left. + (P_{k-1}^k f^{k-1})_{2i+1,2j+2} + (P_{k-1}^k f^{k-1})_{2i+2,2j+1} \right), \end{aligned}$$

y dado que el error cometido por $\hat{f}_{2i+1,2j}^k, \hat{f}_{2i,2j+1}^k, \hat{f}_{2i+2,2j+1}^k, \hat{f}_{2i+1,2j+2}^k$ es inferior al cometido por el operador P_{k-1}^k , obtenemos la aproximación dada por 3.15.

3.4

Resultados numéricos

Las subsecciones 3.4.1 y 3.4.2 muestran los resultados obtenidos tras aplicar diferentes técnicas interpolatorias como operador predicción a dos imágenes test, una de carácter geométrico y la segunda una imagen real.

Los operadores predicción empleados son LIN4, ENO4, WENO4, RAT42, RAT41 y RATIE2 descritos en la sección 1.7.

En todas las pruebas se ha utilizado el algoritmo de control del error (Algoritmo 3.7).

Las tablas recogen el error cometido en términos de $\|\cdot\|_\infty, \|\cdot\|_2$, PSNR (peak-signal-to-noise-ratio) definido como

$$PSNR = 10 \log_{10} \left(\frac{256^2}{\|f^L - \hat{f}^L\|_2} \right),$$

el número de elementos no nulos (Nnz) tras aplicar el operador cuantización, número de bytes utilizados para comprimir la imagen y la tasa de compresión conseguida en términos de bits por pixel definida como $\text{bpp} = \frac{N}{513^2}$, donde N representa el número de bits necesarios para guardar la imagen y el denominador el número de píxeles de la imagen original.

3.4.1

Imagen geométrica

La tabla 3.1 muestra los resultados obtenidos al aplicar las diferentes técnicas a una imagen geométrica. En ellos podemos observar que los métodos derivados de la interpolación racional superan ligeramente a las técnicas LIN4 y ENO4 tanto en términos de PSNR como en tasa de compresión. Observar que los resultados obtenidos por RAT42, RAT41 y RATIE2 son similares a los conseguidos por WENO4 en estos términos.

Las figuras 3.5 y 3.6 muestran las reconstrucciones obtenidas y un detalle de éstas. El parámetro de cuantización empleado para obtener dichas reconstrucciones es $\varepsilon = 16$.

Podemos observar la mejora obtenida por las técnicas no lineales frente a LIN4 en la definición de los perfiles. Notar que con las técnicas WENO4, RAT41, RAT42 y RATIE2 se consigue una mayor calidad visual. No obstante, la reconstrucción de estas técnicas resulta superior a la obtenida por ENO4.

3.4.2

Imagen real

Los resultados obtenidos al aplicar las diferentes técnicas a una imagen real están recogidos en la tabla 3.2. En ella podemos observar que las técnicas WENO4 y RAT42 son las que obtienen mejores resultados en términos de PSNR y tasa de compresión.

En cuanto a la calidad visual de las reconstrucciones, las figuras 3.7 y 3.8 muestran el resultado obtenido con $\varepsilon = 32$. En dichas figuras se observa una mejora de las técnicas no lineales respecto a LIN4. Cabe destacar la calidad de la reconstrucción RATIE2, siendo esta la que obtiene una mayor definición en los perfiles. A su vez notar que la reconstrucción obtenida por RAT41 vuelve a presentar una serie de artefactos en las proximidades de los contornos.

	$\ \cdot\ _\infty$	$\ \cdot\ _2$	PSNR	Nnz	bytes	bpp
LIN4	1,00	0,32	58,16	23529	27433	0,83
	2,00	0,59	52,69	22751	22552	0,69
	4,00	1,28	46,03	21710	18188	0,55
	8,00	2,56	40,02	17633	13538	0,41
	16,00	5,76	32,96	12140	9916	0,30
ENO4	1,00	0,21	61,59	10195	21292	0,65
	2,00	0,41	55,81	10195	17939	0,55
	4,00	0,71	51,09	10060	14756	0,45
	8,00	1,77	43,22	9997	11639	0,35
	16,00	3,60	37,03	9153	8796	0,27
WENO4	1,00	0,15	64,38	8855	19669	0,60
	2,00	0,29	58,98	8852	16540	0,50
	4,00	0,77	50,40	8918	13537	0,41
	8,00	1,35	45,53	8702	10595	0,32
	16,00	2,82	39,16	8441	7975	0,24
RAT42	1,00	0,17	63,57	9464	19698	0,60
	2,00	0,29	58,83	9468	16453	0,50
	4,00	0,80	50,08	9834	13877	0,42
	8,00	1,45	44,93	9412	10676	0,32
	16,00	3,07	38,41	8747	8191	0,25
RAT41	1,00	0,17	63,80	9095	18978	0,58
	2,00	0,26	59,91	8782	15805	0,48
	4,00	0,63	52,23	8933	13054	0,40
	8,00	1,20	46,59	8468	10169	0,31
	16,00	2,56	40,01	7663	7558	0,23
RATIE2	1,00	0,16	64,02	8164	19036	0,58
	2,00	0,28	59,08	8182	16406	0,50
	4,00	0,76	50,54	8680	14006	0,43
	8,00	1,39	45,30	7792	10575	0,32
	16,00	2,89	38,95	6753	8476	0,26

Tabla 3.1: Resultados de una imagen geométrica.

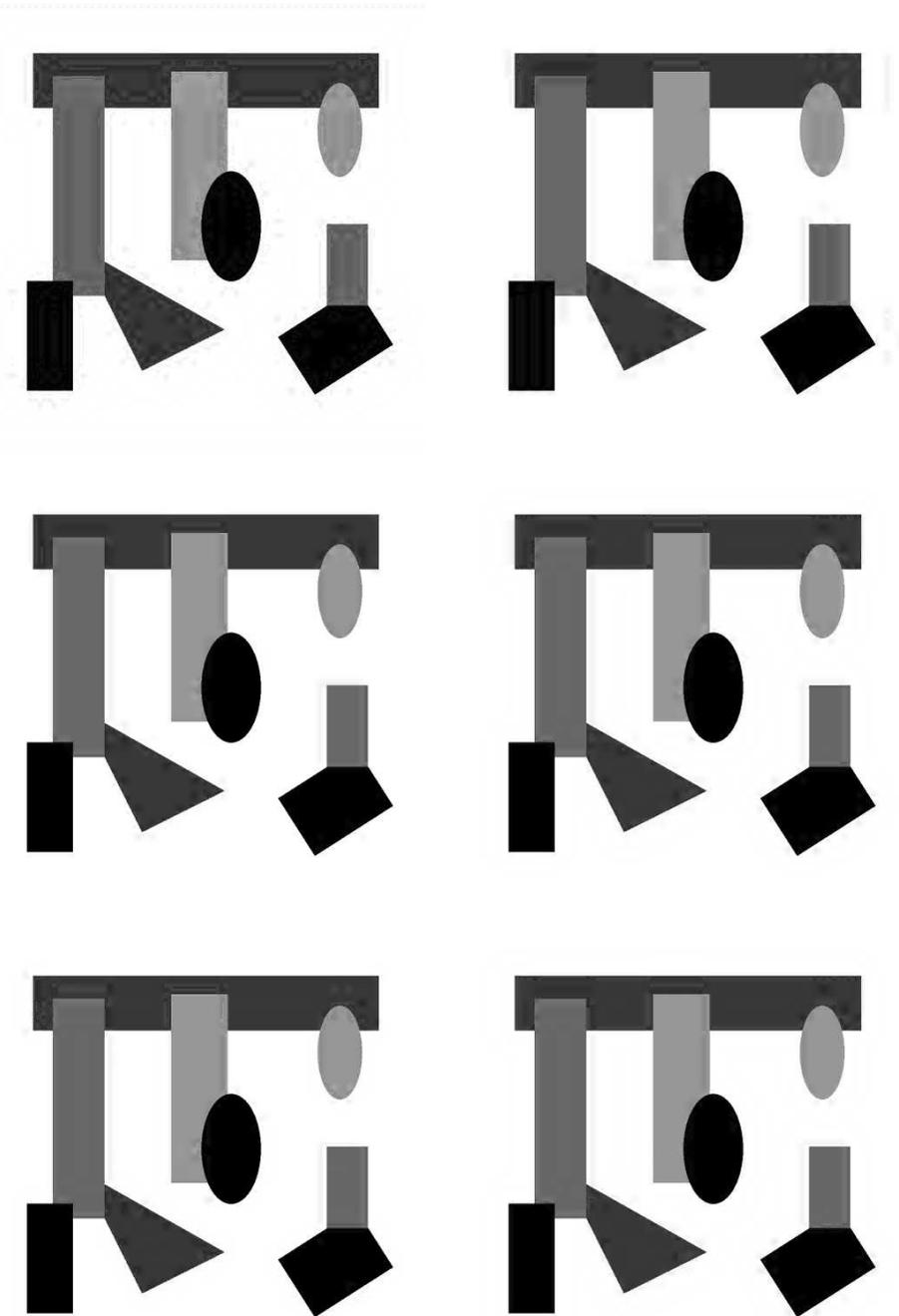


Figura 3.5: Reconstrucciones de una imagen geométrica. Arriba: LIN4, *bpp* 0,30, PSNR 32,96. ENO4, *bpp* 0,27, PSNR 37,03. Centro: WENO4, *bpp* 0,24, PSNR 39,16 RAT42, *bpp* 0,25, PSNR 38,41. Abajo: RAT41, *bpp* 0,23, PSNR 40,01. RATIE2, *bpp* 0,26, PSNR 38,95.

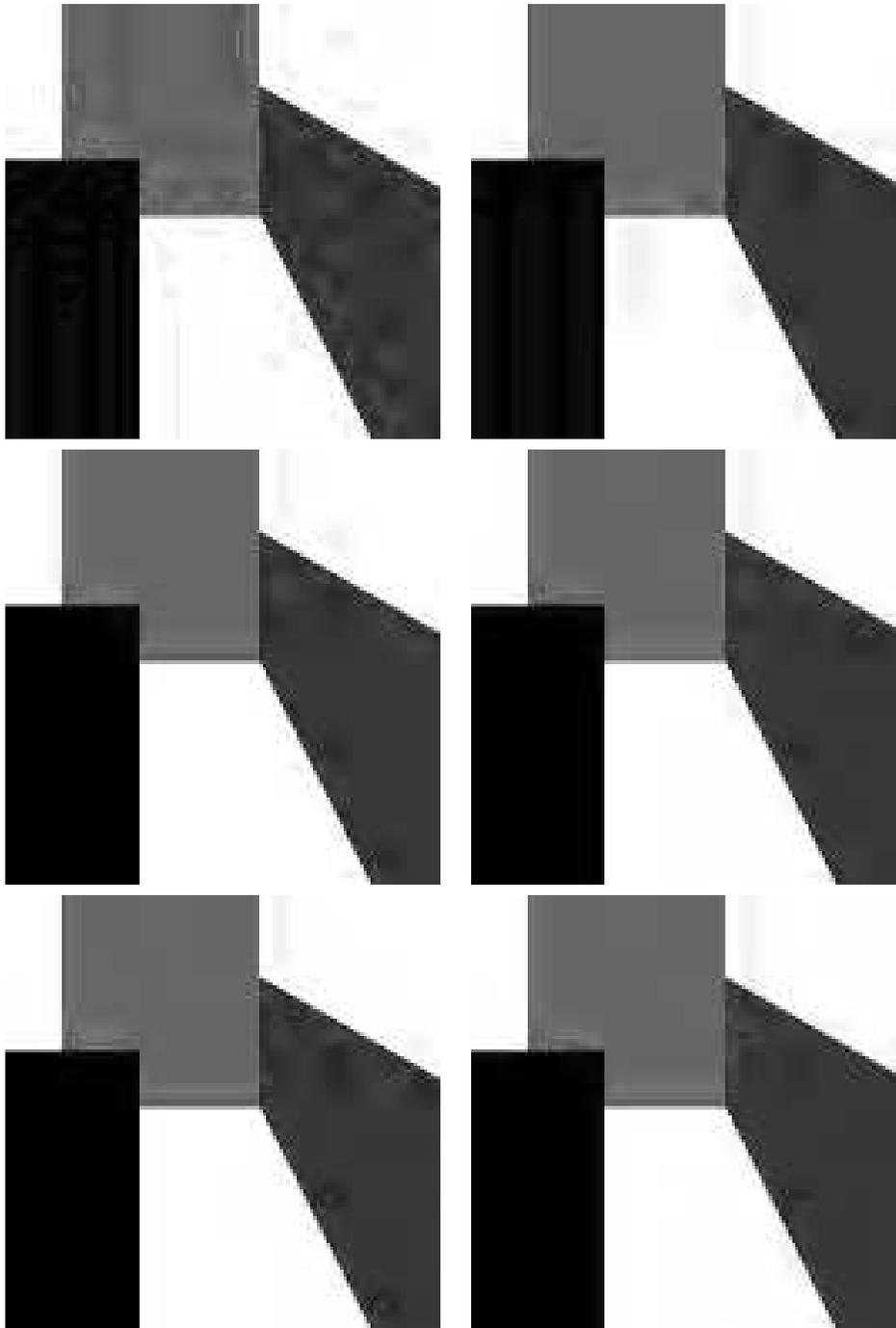


Figura 3.6: *Detalle de las reconstrucciones de la imagen geométrica. Arriba: LIN4, ENO4. Centro: WENO4, RAT42. Abajo: RAT41, RATIE2.*

	$\ \cdot\ _\infty$	$\ \cdot\ _2$	PSNR	Nnz	bytes	bpp
LIN4	1,00	0,57	53,01	165846	102646	3,12
	2,00	1,12	47,16	119171	73531	2,24
	4,00	2,15	41,50	73841	48987	1,49
	8,00	3,96	36,22	38928	29110	0,88
	16,00	6,99	31,28	18034	15395	0,47
ENO4	1,00	0,57	52,99	173395	107993	3,28
	2,00	1,13	47,08	126576	78210	2,38
	4,00	2,17	41,42	78076	52225	1,59
	8,00	3,95	36,23	40675	30634	0,93
	16,00	6,77	31,55	18988	15663	0,48
WENO4	1,00	0,57	53,03	165767	102809	3,13
	2,00	1,12	47,20	116621	72964	2,22
	4,00	2,11	41,68	69141	47310	1,44
	8,00	3,80	36,58	34878	26994	0,82
	16,00	6,42	32,01	16316	13490	0,41
RAT42	1,00	0,57	53,02	165023	101583	3,09
	2,00	1,12	47,18	116122	72157	2,19
	4,00	2,13	41,61	69071	46957	1,43
	8,00	3,83	36,49	34474	26850	0,82
	16,00	6,50	31,90	15759	13497	0,41
RAT41	1,00	0,57	53,03	167721	102991	3,13
	2,00	1,12	47,18	120713	74103	2,25
	4,00	2,15	41,53	74546	49416	1,50
	8,00	3,95	36,23	39470	30209	0,92
	16,00	6,76	31,57	18521	16056	0,49
RATIE2	1,00	0,57	53,02	173432	111420	3,39
	2,00	1,13	47,12	127324	81683	2,48
	4,00	2,17	41,43	81221	55938	1,70
	8,00	3,97	36,18	43599	34244	1,04
	16,00	6,79	31,53	21470	18856	0,57

Tabla 3.2: Resultados de una imagen real.



Figura 3.7: Reconstrucciones de una imagen real. Arriba. LIN4, *bpp* 0,23, PSNR 26,71. ENO4, *bpp* 0,24, PSNR 27,25. Centro. WENO4, *bpp* 0,21, PSNR 27,41. RAT42, *bpp* 0,21, PSNR 27,37. RAT41, *bpp* 0,27, PSNR 27,03. Abajo. RATIE2, *bpp* 0,32, PSNR 27,07.

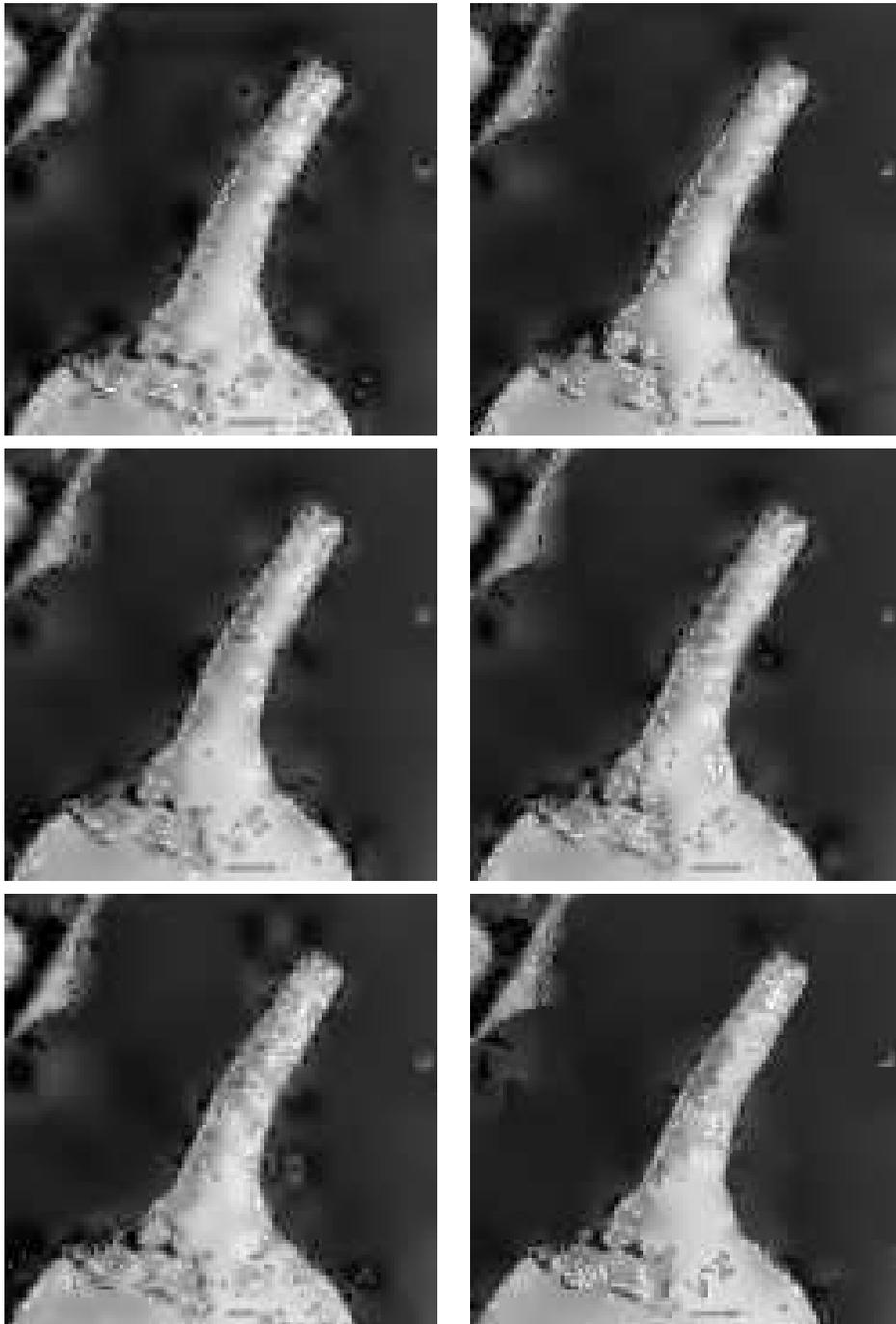


Figura 3.8: *Detalle de las reconstrucciones de la imagen real. Arriba: LIN4, ENO4. Centro: WENO4, RAT42. Abajo: RAT41, Abajo: RATIE2*

3.4.3

Modificación del algoritmo de control del error

En esta sección comparamos los resultados obtenidos al aplicar el algoritmo de control del error y la modificación propuesta en la sección 3.3. La comparación se realiza en términos de $\|\cdot\|_2$ y tasa de compresión (bpp). También comparamos los resultados con los obtenidos por JPEG-LS [48]. En este caso se han empleado técnicas interpolatorias dos-dimensionales (ver sección 1.8):

LIN2 : interpolación lineal con *stencils* cuadrados 2×2 .

LIN4 : interpolación lineal con *stencils* cuadrados 4×4 .

T6 : interpolación ENO con *stencils* triangulares de 6 nodos.

T4 : interpolación ENO con *stencils* triangulares de 10 nodos.

En todos los experimentos, los coeficientes correspondientes al nivel de resolución inferior han sido redondeados al entero más próximo mientras que los detalles han sido cuantizados por un parámetro ε fijo (ver ecuación (3.13)). La codificación de los datos se ha realizado mediante el algoritmo SPIHT [43] (Set Partitioning In Hierarchical Trees). El tamaño de las imágenes test es de 513×513 píxeles, y el nivel de resolución inferior es $k = 5$, i.e. 33×33 píxeles.

La tabla 3.3 muestra los errores en $\|\cdot\|_2$ y las tasas de compresión obtenidas con las diferentes técnicas empleando el algoritmo de control del error ([1, 10, 29]), y los resultados obtenidos con JPEG-LS ([48]). Los resultados conseguidos con la modificación del algoritmo de control del error [8] (EC2) vienen recogidos en la tabla 3.4. Se puede observar que los resultados obtenidos, por todas las técnicas, para parámetros de cuantización $\varepsilon = 4, 8, 16$, utilizando el algoritmo de control del error sin modificar, son comparables a los obtenidos por JPEG-LS. Notar además, que la modificación EC2 produce una mejora, tanto en términos de $\|\cdot\|_2$ como en tasas de compresión, en todas las técnicas aplicadas, siendo en este caso

los parámetros $\varepsilon = 1, 2$ los únicos que no consiguen mejorar los resultados obtenidos por JPEG-LS.

La figura 3.9 muestra las reconstrucciones con las técnicas LIN2, LIN4 y T6 con el algoritmo de control del error modificado, así como la reconstrucción de JPEG-LS. En ésta podemos observar como la reconstrucciones obtenidas por LIN2, LIN4 y T6 son mejores, visualmente, que la obtenida por JPEG-LS.

ε	LIN2		LIN4		T6		T10		JPEG-LS	
	$\ \cdot\ _2$	bpp								
1	0,57	3,28	0,57	3,22	0,57	3,33	0,57	3,36	0,81	2,16
2	1,11	2,32	1,12	2,30	1,12	2,38	1,13	2,43	1,39	1,73
4	2,10	1,50	2,14	1,51	2,14	1,55	2,16	1,61	2,48	1,47
8	3,79	0,84	3,94	0,89	3,88	0,90	3,94	0,94	4,51	1,02
16	6,41	0,43	6,89	0,47	6,59	0,47	6,72	0,49	8,30	0,66

Tabla 3.3: Resultados obtenidos empleando el algoritmo EC con diferentes operadores predicción.

ε	LIN2		LIN4		T6		T10	
	$\ \cdot\ _2$	bpp						
1	0,57	3,07	0,57	3,03	0,57	3,10	0,57	3,12
2	1,10	2,13	1,10	2,12	1,11	2,17	1,11	2,20
4	2,06	1,34	2,09	1,36	2,08	1,38	2,09	1,41
8	3,69	0,74	3,78	0,77	3,74	0,77	3,78	0,79
16	6,25	0,37	6,51	0,39	6,36	0,39	6,42	0,41

Tabla 3.4: Resultados obtenidos empleando el algoritmo EC2 con diferentes operadores predicción.



Figura 3.9: Reconstrucciones de una imagen real con el algoritmo EC2, $\varepsilon = 16$. Arriba: LIN2, LIN4. Abajo: T6. JPEG-LS.

3.5

Multirresolución de Harten para medias en celda

Para la multirresolución de medias en celda consideraremos $\mathcal{F} = L^1(\Omega)$, y sea $C^k = \{c_i^k\}$ un recubrimiento finito de celdas disjuntas de Ω , es decir:

$$\overline{\bigcup c_i^k} = \overline{\Omega}, \quad c_i^k \cap c_j^k = \emptyset, \quad i \neq j. \quad (3.17)$$

El operador discretización en este contexto, en el nivel de resolución k , se define como:

$$\mathcal{D}_k : L^1(\Omega) \Rightarrow V^k, \quad (\mathcal{D}_k f)_i = \frac{1}{|c_i^k|} \int_{c_i^k} f(x) dx,$$

donde $|c_i^k| = \int_{c_i^k} dx$ y V^k es el espacio de sucesiones cuya dimensión coincide con el número de celdas del recubrimiento C^k . Notar que $(\mathcal{D}_k f)_i$ no es más que la media de la función $f(x)$ en la celda c_i^k .

Sea ahora $\{C^k\}_{k=0}^L$ una sucesión de refinamientos del recubrimiento anterior, de modo que C^k se obtiene dividiendo las celdas de C^{k-1} en un número finito q de celdas disjuntas $\{c_{il}^k\}_{l=1}^q$, $\overline{\bigcup_{l=1}^q c_{il}^k} = \overline{c_i^{k-1}}$.

Empleando la aditividad de la integral, tenemos:

$$(\mathcal{D}_{k-1} f)_i = \frac{1}{|c_i^{k-1}|} \sum_{l=1}^q |c_{il}^k| (\mathcal{D}_k f)_{i_l}, \quad (3.18)$$

de la cual podemos deducir que $\{\mathcal{D}_k\}$ es ennidada, ya que es evidente que si $\mathcal{D}_k f = 0$ entonces $\mathcal{D}_{k-1} f = 0$. Además a partir de la relación anterior definimos también el operador decimación como:

$$D_k^{k-1} f^k := D_k^{k-1} \mathcal{D}_k f = \mathcal{D}_{k-1} f. \quad (3.19)$$

Sea ahora \mathcal{R}_k un operador reconstrucción cualesquiera, debido al hecho de que el error de predicción definido en (3.7) pertenece

al núcleo de D_k^{k-1} , haciendo uso de (3.18) y (3.19) tenemos:

$$D_k^{k-1}e^k = 0 \iff \sum_{l=1}^q |c_{i_l}^k| e_{i_l}^k = 0. \quad (3.20)$$

La relación (3.20) nos indica que los errores de predicción $\{e_{i_l}^k\}_{l=1}^q$ son linealmente dependientes, y por tanto podemos definir los coeficientes d^k tomando $q-1$ combinaciones lineales de $\{e_{i_l}^k\}_{l=1}^q$ en cada celda c_i^{k-1} de forma que junto con (3.20) formen un sistema lineal compatible determinado de q ecuaciones con q incógnitas.

La condición de consistencia (3.1), se traduce en este caso en

$$(\mathcal{D}_k \mathcal{R}_k f^k)_i = \frac{1}{|c_i^k|} \int_{c_i^k} \mathcal{R}_k(x; f^k) dx = f_i^k, \quad \forall f^k \in V^k. \quad (3.21)$$

Un operador reconstrucción que verifique (3.21) puede ser construido empleando una reconstrucción polinómica a trozos. Así, si llamamos $\mathcal{S}_i^k = \{c_{i_m}^k\}_{m=1}^{s(r)}$ al *stencil* formado por $s(r)$ celdas de C^k con $c_i^k \in \mathcal{S}_i^k$, podemos construir un polinomio $p_i(x; \mathcal{D}_k f)$ de grado r en \mathbb{R}^m resolviendo el siguiente sistema lineal:

$$f_{i_m}^k = \frac{1}{|c_{i_m}^k|} \int_{c_{i_m}^k} p_i(x; \mathcal{D}_k f) dx, \quad m = 1, \dots, s(r),$$

siendo $p_i(x; \mathcal{D}_k f)$ el único polinomio de grado r cuyas medias coinciden con $(\mathcal{D}_k f)_{i_m}$ si $c_{i_m}^k \in \mathcal{S}_i^k$. Una vez hallado dicho polinomio, definimos el operador reconstrucción \mathcal{R}_k como:

$$(\mathcal{R}_k f^k)(x) = p_i(x; \mathcal{D}_k f), \quad \forall x \in c_i^k.$$

3.5.1

Multirresolución en $[0, 1]$

Dado el nivel de resolución L , y dado $\{x_i^L\}_{i=0}^{J_L}$, una secuencia de particiones de nodos igualmente espaciados del intervalo $[0, 1]$, de modo que $x_i^L = ih_L$, $h_L = 1/J_L$, $J_L = 2^L J_0$, con $J_0 \in \mathbb{N} \setminus \{0\}$, podemos definir las particiones en un nivel de resolución inferior $k = 0, \dots, L-1$ de la siguiente manera:

$$x_i^{k-1} = x_{2i}^k, \quad i = 0, \dots, J_{k-1} := J_k/2.$$

Con las definiciones anteriores, llamamos $c_i^k = (x_{i-1}^k, x_i^k)$, $i = 1, \dots, J_k$, y denotaremos por $\bar{f}_i^k = (\mathcal{D}_k f)_i$ a la media de f en la celda c_i^k , es decir:

$$\bar{f}_i^k = (\mathcal{D}_k f)_i = \frac{1}{h_k} \int_{x_{i-1}^k}^{x_i^k} f(x) dx. \quad (3.22)$$

Notar que así definidas, $\{c_i^k\}_{i=0}^{J_k}$, $k = 0, \dots, L$, verifican (3.17).

A partir de la aditividad de la integral, podemos definir el operador decimación de la siguiente forma:

$$\begin{aligned} (D_k^{k-1} \bar{f}^k)_i &= \frac{1}{h_{k-1}} \int_{x_{i-1}^{k-1}}^{x_i^{k-1}} f(x) dx = \frac{1}{2h_k} \left(\int_{x_{2i-2}^k}^{x_{2i-1}^k} f(x) dx + \int_{x_{2i-1}^k}^{x_{2i}^k} f(x) dx \right) \\ &= \frac{1}{2} (\bar{f}_{2i-1}^k + \bar{f}_{2i}^k), \end{aligned}$$

la cual nos indica que en el nivel de resolución $k - 1$, podemos obtener \bar{f}_i^{k-1} como la media entre \bar{f}_{2i-1}^k y \bar{f}_{2i}^k . Además, como $e^k \in \mathcal{N}(D_k^{k-1})$, obtenemos la siguiente relación:

$$\frac{1}{2} (e_{2i-1}^k + e_{2i}^k) = 0,$$

que nos permite conocer el error e_{2i}^k a partir de e_{2i-1}^k . Aprovechando este hecho, definimos los coeficientes d_i^k como el error cometido por el operador predicción en la celda c_{2i-1}^k , es decir:

$$d_i^k = e_{2i-1}^k.$$

Es evidente entonces que, conocido d^k , podemos obtener e^k como sigue:

$$\begin{aligned} e_{2i-1}^k &= d_i^k, \\ e_{2i}^k &= -d_i^k, \end{aligned}$$

con $1 \leq i \leq J_{k-1} = J_k/2$.

Para definir los operadores reconstrucción y predicción, utilizamos la reconstrucción vía función primitiva explicada en la sección 2.1. Procediendo de la misma forma que en dicha sección, definiremos la sucesión $\{F_i^k\}$ de modo que

$$F_i^k = h_k \sum_{j=1}^i \bar{f}_j^k = \int_0^{x_i^k} f(x) dx.$$

Definido así, $\{F_i^k\}$ no es más que la discretización de valores puntuales en el nivel de resolución k de la función $F(x) = \int_0^x f(y)dy$. Además tenemos:

$$\bar{f}_i^k = \frac{F_i^k - F_{i-1}^k}{h_k}.$$

Si tomamos $\mathcal{I}(x; F^k)$ un interpolante de $F(x)$, de modo que $\mathcal{I}(x_i^k; F^k) = F(x_i^k)$, podemos definir el operador reconstrucción \mathcal{R}_k para la discretización de medias en celda como:

$$\mathcal{R}_k(x; \bar{f}^k) := \frac{d}{dx} \mathcal{I}(x; F^k).$$

Notar que así definido, es sencillo comprobar que los operadores \mathcal{D}_k y \mathcal{R}_k verifican la condición de consistencia $\mathcal{D}_k \mathcal{R}_k = I_k$:

$$(\mathcal{D}_k(\mathcal{R}_k(x; \bar{f}^k)))_i = \frac{1}{h_k} \int_{x_{i-1}^k}^{x_i^k} \frac{d}{dx} \mathcal{I}(x; F^k) dx = \frac{F_i^k - F_{i-1}^k}{h_k} = \bar{f}_i^k.$$

El operador predicción lo definimos empleando (3.2):

$$\begin{aligned} (P_{k-1}^k \bar{f}^{k-1})_{2i-1} &= (\mathcal{D}_k \mathcal{R}_{k-1} \bar{f}^{k-1})_{2i-1} = \frac{1}{h_k} \int_{x_{2i-2}^k}^{x_{2i-1}^k} \frac{d}{dx} \mathcal{I}(x; F^{k-1}) dx \\ &= \frac{1}{h_k} (\mathcal{I}(x_{2i-1}^k; F^{k-1}) - \mathcal{I}(x_{2i-2}^k; F^{k-1})) \\ &= \frac{1}{h_k} (\mathcal{I}(x_{2i-1}^k; F^{k-1}) - F_{i-1}^{k-1}), \\ (P_{k-1}^k \bar{f}^{k-1})_{2i} &= (\mathcal{D}_k \mathcal{R}_{k-1} \bar{f}^{k-1})_{2i} = \frac{1}{h_k} \int_{x_{2i-1}^k}^{x_{2i}^k} \frac{d}{dx} \mathcal{I}(x; F^{k-1}) dx \\ &= \frac{1}{h_k} (\mathcal{I}(x_{2i}^k; F^{k-1}) - \mathcal{I}(x_{2i-1}^k; F^{k-1})) \\ &= \frac{1}{h_k} (F_i^{k-1} - \mathcal{I}(x_{2i-1}^k; F^{k-1})). \end{aligned}$$

Una vez definidos los operadores necesarios, los algoritmos de multirresolución para la discretización de medias en celda son los siguientes:

Algoritmo. 3.9. *Transformación directa para medias en celda en $[0, 1]$.*

$$\bar{f}^L \rightarrow M\bar{f}^L \begin{cases} \text{for } k = L, \dots, 1 \\ \bar{f}_i^{k-1} = \frac{1}{2}(\bar{f}_{2i-1}^k + \bar{f}_{2i}^k), & 1 \leq i \leq J_{k-1} \\ d_i^k = \bar{f}_{2i-1}^k - (P_{k-1}^k \bar{f}^{k-1})_{2i-1}, & 1 \leq i \leq J_{k-1}, \end{cases}$$

Algoritmo. 3.10. *Transformación inversa para medias en celda en $[0, 1]$.*

$$M\bar{f}^L \rightarrow M^{-1}M\bar{f}^L \begin{cases} \text{for } k = 1, \dots, L \\ \bar{f}_{2i-1}^k = (P_{k-1}^k \bar{f}^{k-1})_{2i-1} + d_i^k, & 1 \leq i \leq J_{k-1}, \\ \bar{f}_{2i}^k = 2\bar{f}_i^{k-1} - \bar{f}_{2i-1}^k, & 1 \leq i \leq J_{k-1}. \end{cases}$$

En los algoritmos anteriores, podemos observar que los coeficientes $d^k(\bar{f})$ no son nada más que el error que cometemos al aproximar la media de f en los intervalos impares en el nivel de resolución k . Veamos la relación existente entre este error y el error cometido al interpolar los valores puntuales de F en los nodos impares en el mismo nivel de resolución:

$$\begin{aligned} d_i^k(\bar{f}) &= \bar{f}_{2i-1}^k - (P_{k-1}^k D_k^{k-1} \bar{f}^k)_{2i-1} = f_{2i-1}^k - (\mathcal{D}_k \mathcal{R}_{k-1} \bar{f}^k)_{2i-1} \\ &= \frac{1}{h_k} (F_{2i-1}^k - F_{2i-2}^k) - \frac{1}{h_k} \int_{x_{2i-2}^k}^{x_{2i-1}^k} \frac{d}{dx} \mathcal{I}(x; F^{k-1}) \\ &= \frac{1}{h_k} [(F_{2i-1}^k - F_{2i-2}^k) - (\mathcal{I}(x_{2i-1}^k; F^{k-1}) - \mathcal{I}(x_{2i-2}^k; F^{k-1}))] \\ &= \frac{1}{h_k} (F_{2i-1}^k - \mathcal{I}(x_{2i-1}^k; F^{k-1})) = \frac{e_i^k(F)}{h_k}. \end{aligned}$$

Así, el coeficiente $d_i^k(\bar{f})$ coincide con el error cometido por el interpolante de la función primitiva en el nodo x_{2i-1}^k , el cual resulta ser el coeficiente de escala $d_i^k(F)$ en el contexto de la multirresolución de valores puntuales (ver [10]), pudiendo establecerse la siguiente relación entre los detalles de la multirresolución de medias en

celda y valores puntuales:

$$d_i^k(\bar{f}) = \frac{d_i^k(F)}{h_k}. \quad (3.23)$$

En el caso de que $\mathcal{I}(x; F^k)$ sea un interpolante polinómico a trozos formado por polinomios de grado $r + 1$, obtendremos que el error cometido en aquellas zonas donde el *stencil* no cruce ninguna singularidad es de orden $O(h_{k-1}^{r+2})$ (ver (1.5)), y por tanto:

$$d_i^k(F) = O(h_{k-1}^{r+2}),$$

siempre que el *stencil* empleado para interpolar en $[x_{i-1}^k, x_i^k]$ no cruce ninguna singularidad. Notar que entonces, $\mathcal{R}_k(x; \bar{f}^{k-1})$ será un polinomio a trozos de grado r , y partir de (3.23) tenemos que:

$$d_i^k(\bar{f}) = O(h_{k-1}^{r+1})$$

Si el *stencil* empleado cruza una discontinuidad de $f^{(p)}$ ($p \leq r$), ésta será una discontinuidad de $F^{(p+1)}$, y por tanto haciendo uso de (1.8) y (3.23) tenemos:

$$d_i^k(F) = O(h_{k-1}^{p+1}) \Rightarrow d_i^k(\bar{f}) = O(h_{k-1}^p).$$

De esta manera si empleamos técnicas interpolatorias no lineales, podemos reducir el error cometido ya que siempre que tengamos el espacio suficiente entre las singularidades, y la discontinuidad no se encuentre en el intervalo en el que queremos interpolar, podremos encontrar un *stencil* que no cruce ninguna singularidad y obtener una mejor aproximación.

3.5.2

Multirresolución en $[0, 1] \times [0, 1]$

En este caso podemos interpretar los datos que tenemos como las medias de una función $f(x, y)$ en una malla dada, que nos determina el nivel de resolución de los datos. Sea $\{X^k\}_{k=0}^L$ una secuencia de discretizaciones de modo que $X^k = \{(x_i^k, y_j^k)\}$, $x_i^k = ih_k$, $y_j^k = jh_k$, $i, j = 0, \dots, N_k$, $N_k h_k = 1$, con $h_k = 2^{-k} h_0$, y sea

100 **3.5. Multirresolución de Harten para medias en celda**

$C_{i,j}^k = [x_{i-1}^k, x_i^k] \times [y_{j-1}^k, y_j^k]$. Teniendo en cuenta esta notación, los datos en el nivel de resolución k , $\bar{f}^k = \{\bar{f}_{i,j}^k\}_{i,j=0}^{N_k}$, pueden ser interpretados como:

$$\bar{f}_{i,j}^k = \frac{1}{|C_{i,j}^k|} \int_{C_{i,j}^k} f(x, y) dx dy = \frac{1}{h_k^2} \int_{x_{i-1}^k}^{x_i^k} \int_{y_{j-1}^k}^{y_j^k} f(x, y) dx dy.$$

A partir de la aditividad de la integral, y observando que la celda $C_{i,j}^{k-1}$ está formada por la unión de las celdas $C_{2i,2j}^k$, $C_{2i-1,2j}^k$, $C_{2i,2j-1}^k$ y $C_{2i-1,2j-1}^k$, obtenemos la siguiente relación entre las medias de dos niveles de resolución consecutivos:

$$\begin{aligned} \bar{f}_{i,j}^{k-1} &= \frac{1}{h_{k-1}^2} \int_{C_{i,j}^{k-1}} f(x, y) dx dy \\ &= \frac{1}{h_{k-1}^2} \left(\int_{C_{2i,2j}^k} f(x, y) dx dy + \int_{C_{2i-1,2j}^k} f(x, y) dx dy \right. \\ &\quad \left. + \int_{C_{2i,2j-1}^k} f(x, y) dx dy + \int_{C_{2i-1,2j-1}^k} f(x, y) dx dy \right) \\ &= \frac{1}{4} (\bar{f}_{2i,2j}^k + \bar{f}_{2i-1,2j}^k + \bar{f}_{2i,2j-1}^k + \bar{f}_{2i-1,2j-1}^k). \end{aligned}$$

Haciendo uso de la relación anterior, definimos el operador decimación de la siguiente forma:

$$\bar{f}_{i,j}^{k-1} = (D_k^{k-1} \bar{f}^k)_{i,j} = \frac{1}{4} (\bar{f}_{2i,2j}^k + \bar{f}_{2i-1,2j}^k + \bar{f}_{2i,2j-1}^k + \bar{f}_{2i-1,2j-1}^k). \quad (3.24)$$

El hecho de que se deba verificar $D_k^{k-1} P_{k-1}^k = I_{V^{k-1}}$, hace que el operador predicción cumpla la siguiente igualdad:

$$\begin{aligned} ((P_{k-1}^k \bar{f}^{k-1})_{2i,2j} + (P_{k-1}^k \bar{f}^{k-1})_{2i-1,2j} \\ + (P_{k-1}^k \bar{f}^{k-1})_{2i,2j-1} + (P_{k-1}^k \bar{f}^{k-1})_{2i-1,2j-1}) = 4 \bar{f}_{i,j}^{k-1}. \end{aligned} \quad (3.25)$$

Si definimos $e_{i,j}^k = \bar{f}_{i,j}^k - (P_{k-1}^k \bar{f}^{k-1})_{i,j}$, es fácil ver que $e^k \in \mathcal{N}(D_k^{k-1})$,

pues:

$$\begin{aligned}
 (D_k^{k-1}e^k)_{i,j} &= \frac{1}{4}(e_{2i,2j}^k + e_{2i-1,2j}^k + e_{2i,2j-1}^k + e_{2i-1,2j-1}^k) \\
 &= \frac{1}{4}(\bar{f}_{2i,2j}^k - (P_{k-1}^k \bar{f}^{k-1})_{2i,2j} + \bar{f}_{2i-1,2j}^k - (P_{k-1}^k \bar{f}^{k-1})_{2i-1,2j} \\
 &+ \bar{f}_{2i,2j-1}^k - (P_{k-1}^k \bar{f}^{k-1})_{2i,2j-1} + \bar{f}_{2i-1,2j-1}^k - (P_{k-1}^k \bar{f}^{k-1})_{2i-1,2j-1}) \\
 &= \bar{f}_{i,j}^{k-1} - \frac{1}{4}((P_{k-1}^k \bar{f}^{k-1})_{2i,2j} + (P_{k-1}^k \bar{f}^{k-1})_{2i-1,2j} \\
 &+ (P_{k-1}^k \bar{f}^{k-1})_{2i,2j-1} + (P_{k-1}^k \bar{f}^{k-1})_{2i-1,2j-1}) = 0.
 \end{aligned}$$

Así, gracias a que el error de predicción e^k pertenece al núcleo de D_k^{k-1} , obtenemos la siguiente relación:

$$e_{2i-1,2j-1}^k + e_{2i,2j-1}^k + e_{2i-1,2j}^k + e_{2i,2j}^k = 0, \quad (3.26)$$

gracias a la cual, podemos eliminar la información redundante de $\{e^k\}$ tomando las siguientes combinaciones lineales de los errores de interpolación:

$$\begin{aligned}
 d_{i,j;a}^k &= \frac{1}{4}(e_{2i-1,2j-1}^k + e_{2i,2j-1}^k - e_{2i-1,2j}^k - e_{2i,2j}^k), \\
 d_{i,j;b}^k &= \frac{1}{4}(e_{2i-1,2j-1}^k - e_{2i,2j-1}^k + e_{2i-1,2j}^k - e_{2i,2j}^k), \\
 d_{i,j;c}^k &= \frac{1}{4}(e_{2i-1,2j-1}^k - e_{2i,2j-1}^k - e_{2i-1,2j}^k + e_{2i,2j}^k). \quad (3.27)
 \end{aligned}$$

De este modo se obtiene una correspondencia unívoca entre la información contenida en el nivel de resolución k y la información en el nivel de resolución $k - 1$ más los detalles:

$$\{\bar{f}_{i,j}^k\}_{i,j=1}^{J_k} \leftrightarrow \{\{\bar{f}_{i,j}^{k-1}\}, \{d_{i,j;a}^k\}, \{d_{i,j;b}^k\}, \{d_{i,j;c}^k\}\}_{i,j=1}^{J_{k-1}}.$$

Notar que las ecuaciones (3.26) y (3.27) pueden ser expresadas matricialmente de la siguiente manera:

$$\begin{bmatrix} 0 \\ d_{i,j;a}^k \\ d_{i,j;b}^k \\ d_{i,j;c}^k \end{bmatrix} = B \begin{bmatrix} e_{2i-1,2j-1}^k \\ e_{2i,2j-1}^k \\ e_{2i-1,2j}^k \\ e_{2i,2j}^k \end{bmatrix},$$

con B la siguiente matriz 4×4 :

$$B = \frac{1}{4} \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{bmatrix}.$$

Para recuperar los errores de predicción es suficiente realizar la siguiente transformación:

$$\begin{bmatrix} e_{2i-1,2j-1}^k \\ e_{2i,2j-1}^k \\ e_{2i-1,2j}^k \\ e_{2i,2j}^k \end{bmatrix} = 4B' \begin{bmatrix} 0 \\ d_{i,j;a}^k \\ d_{i,j;b}^k \\ d_{i,j;c}^k \end{bmatrix}.$$

Los algoritmos directo e inverso para la descomposición multi-escala para medias en celda en $[0, 1] \times [0, 1]$ son los siguientes:

Algoritmo. 3.11. *Transformación directa para medias en celda en $[0, 1] \times [0, 1]$: $\bar{f}^L \rightarrow M\bar{f}^L = (\bar{f}^0, d^1, \dots, d^L)$*

```

for  $k = L, \dots, 1$ 
  for  $i, j = 1, J_{k-1}$ 
     $\bar{f}_{i,j}^{k-1} = \frac{1}{4} \sum_{l,m=0,1} \bar{f}_{2i-l,2j-m}^k$ 
  end
  for  $i, j = 0, J_{k-1}$ 
     $e_{2i-1,2j-1}^k = \bar{f}_{2i-1,2j-1}^k - (P_{k-1}^k \bar{f}^{k-1})_{2i-1,2j-1}$ 
     $e_{2i,2j-1}^k = \bar{f}_{2i,2j-1}^k - (P_{k-1}^k \bar{f}^{k-1})_{2i,2j-1}$ 
     $e_{2i-1,2j}^k = \bar{f}_{2i-1,2j}^k - (P_{k-1}^k \bar{f}^{k-1})_{2i-1,2j}$ 
     $e_{2i,2j}^k = \bar{f}_{2i,2j}^k - (P_{k-1}^k \bar{f}^{k-1})_{2i,2j}$ 
     $d_{i,j;a}^k = \frac{1}{4}(e_{2i-1,2j-1}^k + e_{2i,2j-1}^k - e_{2i-1,2j}^k - e_{2i,2j}^k)$ 
     $d_{i,j;b}^k = \frac{1}{4}(e_{2i-1,2j-1}^k - e_{2i,2j-1}^k + e_{2i-1,2j}^k - e_{2i,2j}^k)$ 
     $d_{i,j;c}^k = \frac{1}{4}(e_{2i-1,2j-1}^k - e_{2i,2j-1}^k - e_{2i-1,2j}^k + e_{2i,2j}^k)$ 
  end
end

```

Algoritmo. 3.12. Transformación inversa para medias en celda en $[0, 1] \times [0, 1]$: $M\bar{f}^L = (\bar{f}^0, d^1, \dots, d^L) \rightarrow M^{-1}M\bar{f}^L$

```

for  $k = L, \dots, 1$ 
  for  $i, j = 0, J_{k-1}$ 
     $\bar{f}_{2i-1, 2j-1}^k = (P_{k-1}^k \bar{f}^{k-1})_{2i-1, 2j-1} + d_{i,j;a}^k + d_{i,j;b}^k + d_{i,j;c}^k$ 
     $\bar{f}_{2i, 2j-1}^k = (P_{k-1}^k \bar{f}^{k-1})_{2i-1, 2j} + d_{i,j;a}^k - d_{i,j;b}^k - d_{i,j;c}^k$ 
     $\bar{f}_{2i-1, 2j}^k = (P_{k-1}^k \bar{f}^{k-1})_{2i, 2j-1} - d_{i,j;a}^k + d_{i,j;b}^k - d_{i,j;c}^k$ 
     $\bar{f}_{2i, 2j}^k = (P_{k-1}^k \bar{f}^{k-1})_{2i, 2j} - d_{i,j;a}^k - d_{i,j;b}^k + d_{i,j;c}^k$ 
  end
end

```

Como podemos observar en los algoritmos 3.11 y 3.12, el operador predicción utilizado es un operador dos-dimensional, que tras ser aplicado al nivel de resolución $k-1$, compuesto por $N_{k-1} \times N_{k-1}$ elementos, obtenemos una aproximación al nivel de resolución k que consta de $N_k \times N_k$ elementos con $N_k = 2N_{k-1}$.

En las secciones 2.4 y 2.5 se muestra como definir operadores predicción dos dimensionales.

3.6

Resultados numéricos

Las secciones 3.6.1 y 3.6.2 recogen varios tests numéricos aplicados dos imágenes: una de carácter geométrico y otra de carácter real. Los operadores predicción empleados son LIN2D, ENO2D, WENO3, RAT3 y RAT5, descritos previamente en la sección 2.6. En el capítulo 6 se muestran los resultados obtenidos con los operadores predicción descritos en dicho capítulo.

De igual modo que en la sección 3.4, las tablas 3.5 y 3.6 recogen los resultados obtenidos tras aplicar los diferentes operadores predicción en términos de $\|\cdot\|_\infty$, $\|\cdot\|_2$, PSNR, número de elementos no nulos (Nnz), número de bytes utilizados para comprimir la imagen y tasa de compresión (bpp).

Se ha empleado el algoritmo de control del error ([1, 10, 11]). Los coeficientes *detalle* correspondientes al nivel de resolución k han sido cuantizados por el parámetro ε_k , i.e.

$$\hat{d}_{i,j}^k = qu(d_{i,j}^k; \varepsilon_k) = 2\varepsilon_k \left[\frac{d_{i,j}^k}{2\varepsilon_k} \right],$$

donde $[\cdot]$ no es más que el entero obtenido por redondeo, mientras que los coeficientes correspondientes el nivel inferior de resolución han sido redondeados al entero más cercano. De este modo fijado el parámetro de cuantización para el nivel más fino, ε_L , definimos los parámetros de cuantización de los niveles de resolución inferiores del siguiente modo:

$$\varepsilon_{k-1} = \begin{cases} \varepsilon_k/2 & \text{si } \varepsilon_k \geq 1/2, \\ 1/2 & \text{otro caso.} \end{cases}$$

Los coeficientes correspondientes a la imagen en el nivel de resolución inferior más los detalles cuantizados han sido codificados con el algoritmo SPIHT [43]. Las imágenes test son de 512×512 píxeles y el nivel de resolución inferior empleado es $k = 5$, i.e. 32×32 píxeles.

3.6.1

Imagen geométrica

La tabla 3.5 muestra los resultados obtenidos aplicando los diferentes operadores predicción. Se pueden observar que los mejores resultados, en términos de error en $\|\cdot\|_2$ y tasa de compresión se obtienen con las técnicas ENO2D y RAT3, si bien cabe observar que WENO3 es la que obtiene un mayor PSNR al emplear el parámetro de cuantización menor. Notar también que los resultados obtenidos por RAT5 son superiores a los obtenidos por el esquema lineal.

Las figuras 3.10 y 3.11 muestran las reconstrucciones obtenidas y un detalle de éstas. El parámetro de cuantización empleado es $\varepsilon_L = 16$. En ellas observamos que WENO3 es la que consigue

una mejor reconstrucción. No obstante ENO2D, RAT3 y RAT5 mejoran la difusión producida en los contornos por LIN2D. Notar que visualmente los resultados obtenidos por RAT5 no son tan satisfactorios como los alcanzados por RAT3.

	$\ \cdot\ _\infty$	$\ \cdot\ _2$	PSNR	Nnz	bytes	bpp
LIN2D	5,04	0,47	54,66	13654	12394	0,38
	7,43	0,64	52,01	12259	10169	0,31
	14,19	1,11	47,23	9755	7763	0,24
	31,39	2,02	42,04	7342	5472	0,17
	58,67	3,73	36,74	3892	3354	0,10
ENO2D	5,14	0,38	56,57	6995	9296	0,28
	7,82	0,48	54,59	6284	7771	0,24
	18,42	0,77	50,48	5739	6254	0,19
	32,32	1,47	44,82	4749	4657	0,14
	67,18	3,00	38,61	3343	3371	0,10
WENO3	4,55	0,31	58,44	6647	9468	0,29
	7,21	0,40	56,03	6152	7973	0,24
	17,30	0,74	50,82	5706	6391	0,20
	31,96	1,39	45,33	4917	4841	0,15
	66,00	2,96	38,74	3703	3525	0,11
RAT3	4,83	0,41	56,01	7040	9589	0,29
	7,93	0,51	54,03	5978	7795	0,24
	14,80	0,74	50,81	5476	6200	0,19
	28,20	1,35	45,54	4581	4614	0,14
	63,86	2,87	39,00	3166	3236	0,10
RAT5	4,64	0,49	54,39	12133	10603	0,32
	7,65	0,64	52,05	9622	8250	0,25
	13,10	1,04	47,85	7725	6355	0,19
	32,98	1,92	42,51	5005	4540	0,14
	62,64	3,25	37,93	2997	2935	0,09

Tabla 3.5: Imagen geométrica. $\|\cdot\|_\infty$, $\|\cdot\|_2$, PSNR, coeficientes no nulos (Nnz), bytes y bits por píxel (bpp)

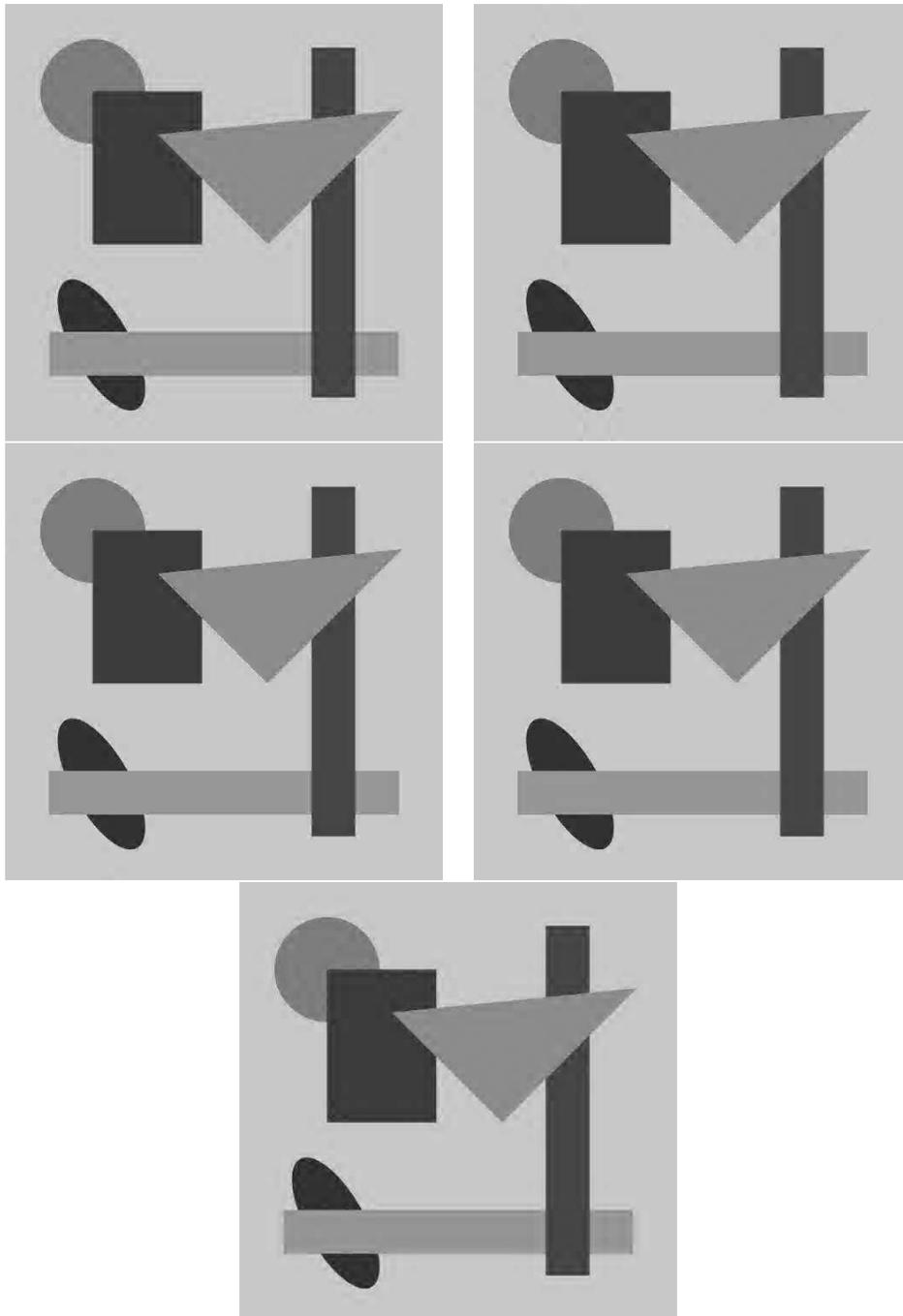


Figura 3.10: Reconstrucciones de una imagen geométrica. Arriba: LIN2D: PSNR 36,74, *bpp* 0,10. ENO2D: PSNR: 38,61, *bpp* 0,10. Centro: WENO3: 38,74, *bpp* 0,11. RAT3: PSNR 39,00, *bpp* 0,10. Abajo: RAT5: PSNR 37,93, *bpp* 0,09.

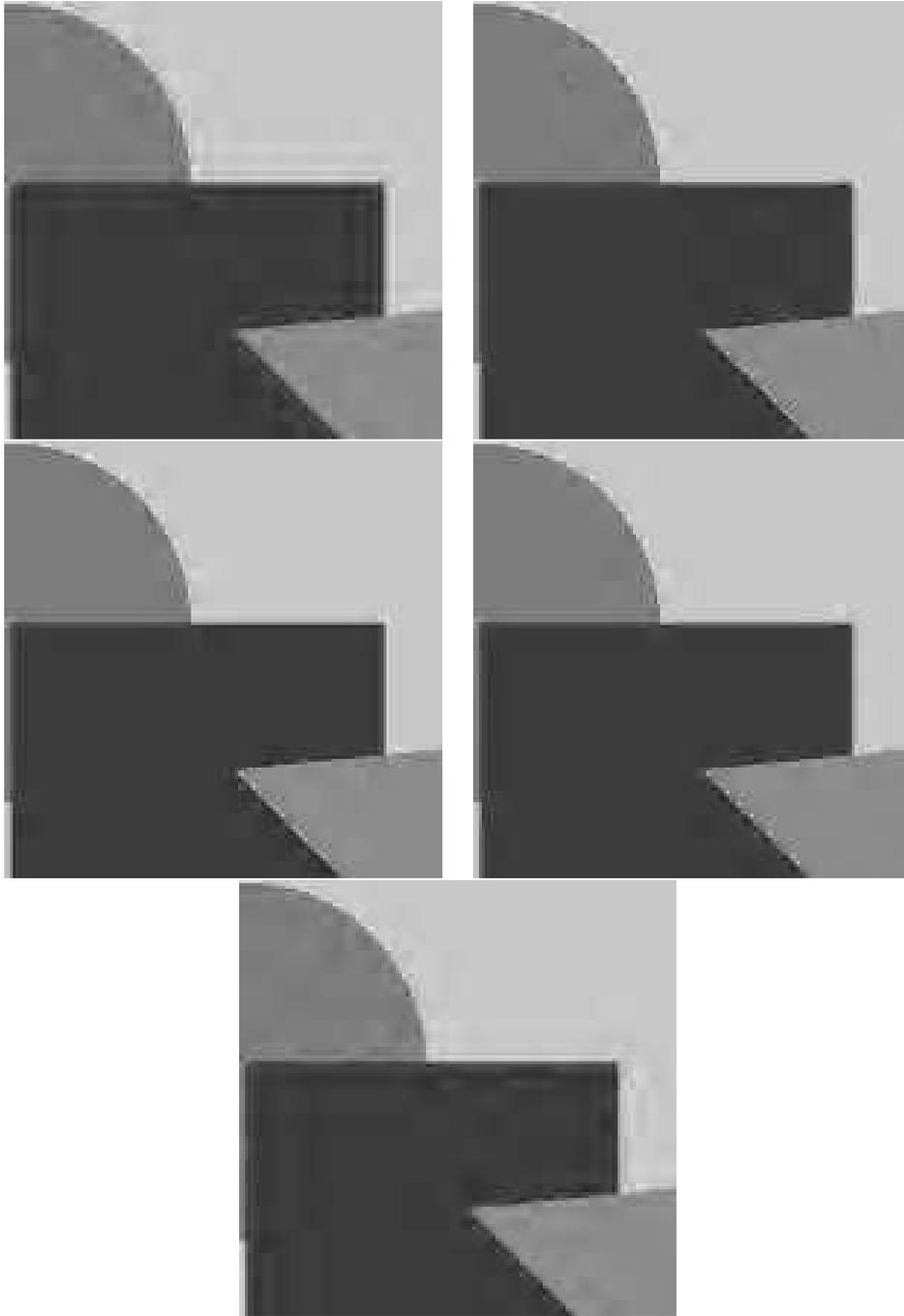


Figura 3.11: *Detalle de la reconstrucción de una imagen geométrica. Arriba: LIN2D, ENO2D. Centro: WENO3, RAT3. Abajo: RAT5*

3.6.2

Imagen real

Los resultados de aplicar los diferentes tests numéricos a una imagen real vienen recogidos en la tabla 3.6. En este caso las técnicas basadas en la interpolación racional, RAT3 y RAT5, consiguen unos resultados ligeramente superiores que el resto, tanto en términos de $\| \cdot \|_2$ como en términos de PSNR. Observar que en este caso los resultados obtenidos por ENO2D y WENO3 son peores que los que ofrece LIN2D.

Respecto a la calidad visual de las reconstrucciones obtenidas, en las figuras 3.12 y 3.13 podemos observar las imágenes obtenidas empleando un parámetro de cuantización $\varepsilon_L = 32$. De igual modo que en la imagen geométrica, las técnicas no lineales consiguen una mayor definición en los contornos, siendo en este caso WENO3 y RAT3 las que obtienen mejores resultados. RAT5 vuelve a obtener un mayor emborronamiento de los perfiles que RAT3 a pesar de mejorar los resultados de LIN2D.

	$\ \cdot\ _\infty$	$\ \cdot\ _2$	PSNR	Nnz	bytes	bpp
LIN2D	5,45	1,26	46,15	124015	73102	2,23
	9,43	2,01	42,12	80221	49033	1,50
	16,41	3,30	37,78	42732	28852	0,88
	29,77	5,08	34,04	20634	15187	0,46
	59,91	7,33	30,86	10353	8043	0,25
ENO2D	5,89	1,29	45,98	131795	78109	2,38
	9,50	2,07	41,85	86353	52872	1,61
	17,27	3,45	37,40	45993	31312	0,96
	31,99	5,28	33,71	21229	16104	0,49
	64,40	7,63	30,51	10582	8510	0,26
WENO3	5,28	1,27	46,06	127978	75893	2,32
	10,21	2,03	42,01	83582	51367	1,57
	17,73	3,37	37,60	43973	30313	0,93
	33,63	5,16	33,92	21017	15951	0,49
	64,91	7,43	30,75	10877	8626	0,26
RAT3	5,86	1,26	46,14	122876	72317	2,21
	8,73	2,01	42,12	78912	48474	1,48
	16,98	3,30	37,79	40432	28266	0,86
	31,34	5,00	34,18	18811	14730	0,45
	61,50	7,16	31,07	9788	7985	0,24
RAT5	6,21	1,26	46,15	121562	70404	2,15
	8,91	2,01	42,12	77382	46484	1,42
	15,96	3,30	37,79	39667	26762	0,82
	34,18	5,02	34,15	17781	13471	0,41
	61,04	7,16	31,07	8641	7041	0,21

Tabla 3.6: Imagen real. $\|\cdot\|_\infty$, $\|\cdot\|_2$, PSNR, coeficientes no nulos (Nnz), bytes y bits por píxel (bpp)



Figura 3.12: Reconstrucciones de una imagen real. Arriba: LIN2D: PSNR 27,69, *bpp* 0,14. ENO2D: PSNR: 27,27, *bpp* 0,15. Centro: WENO3 27,52, *bpp* 0,15. RAT3: PSNR 27,97, *bpp* 0,14. RAT5: PSNR 28,09, *bpp* 0,12.

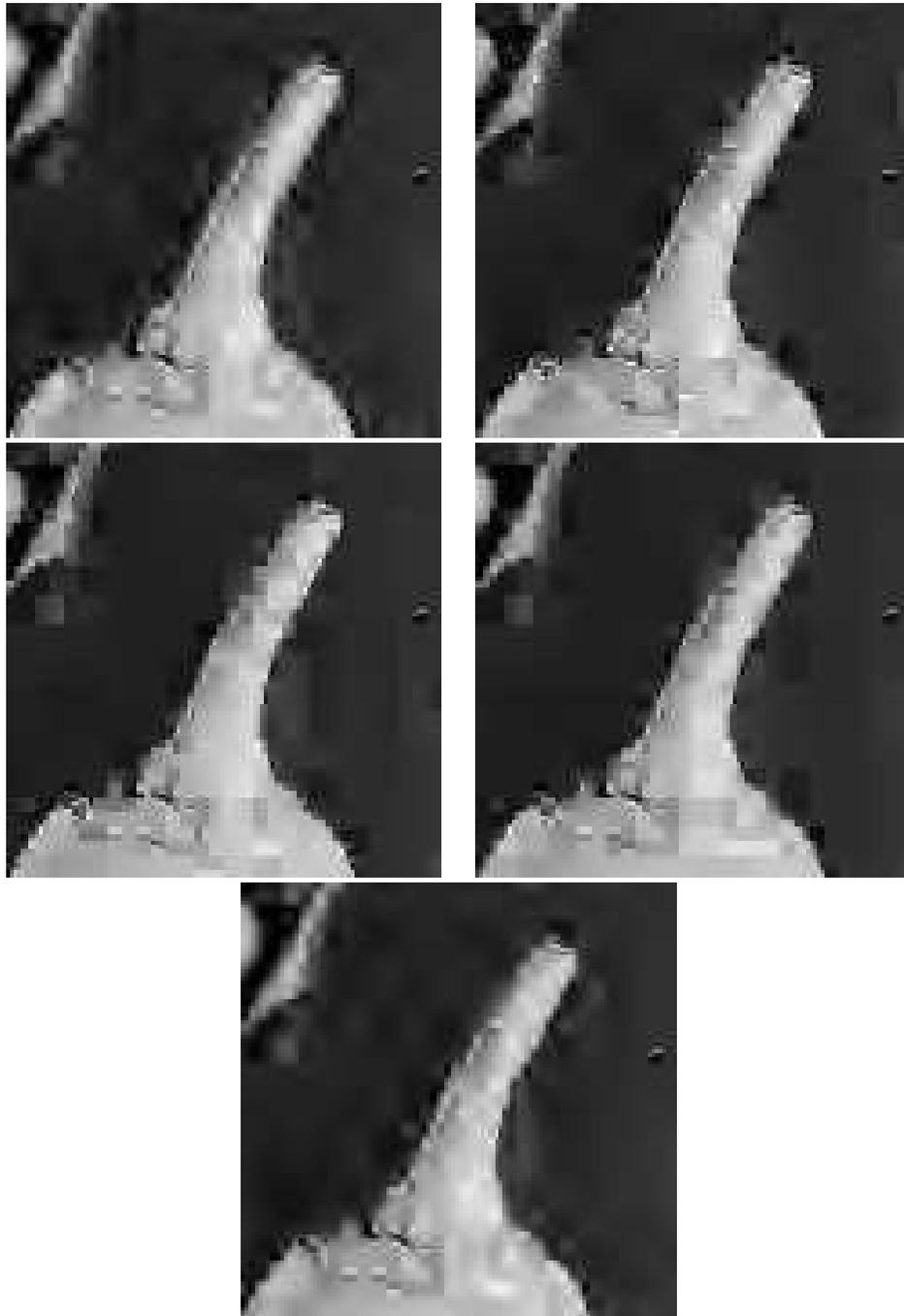


Figura 3.13: *Detalle de la reconstrucción de una imagen real. Arriba: LIN2D, ENO2D. Centro: WENO3, RAT3. Abajo: RAT5*

4

Deteccción de contornos

4.1

Introducción

La localización de los contornos puede resultar de gran utilidad en aplicaciones como la interpolación o compresión de imágenes. En los capítulos 1, 2 y 3, se observa que los resultados menos satisfactorios se producen en las proximidades de los perfiles. En caso de conocer un mapa de los perfiles de una imagen, existe la posibilidad de adaptar la técnica interpolatoria empleada en aquellas zonas donde la calidad de la predicción se vea degradada, mejorando así los resultados obtenidos.

De ahora en adelante, al referirnos a una imagen $n \times m$ en escala de grises, estaremos haciendo referencia a la siguiente aplicación:

$$I : \{1, \dots, n\} \times \{1, \dots, m\} \rightarrow [0, 255],$$

donde $I(n_1, m_1)$ representa el nivel de gris del píxel (n_1, m_1) . Así, los contornos de las imágenes no son más que cambios bruscos en el nivel de gris, aportándonos éstos, información muy importante en el reconocimiento de formas.

La estructura del capítulo es la siguiente: en primer lugar revisaremos diferentes detectores de contornos basados en la mera aplicación de unos filtros determinados. Posteriormente, explicaremos en que consisten el detector de contornos de Canny, el cual halla los perfiles de la imagen determinando aquellos puntos que representan máximos locales en la dirección del gradiente.

En la sección 4.5 explicamos como emplear los algoritmos jerárquico y no jerárquico de selección de *stencil* (ver sección 1.2) para detectar los perfiles de las imágenes. Si la imagen carece de textura, se pueden obtener los contornos de ésta a partir de las celdas aisladas por los *stencils* seleccionados para interpolar en sus celdas vecinas. Sin embargo, en imágenes reales, la presencia de textura provoca que exista un mayor número de celdas aisladas, no siendo posible así obtener una correcta localización de los contornos.

Para solucionar este problema, en la sección 4.6 proponemos un algoritmo de detección de contornos que utiliza técnicas de selección de *stencil* y unos filtros similares a los de Sobel, empleados para determinar la orientación del perfil y eliminar aquellos contornos detectados a causa de la textura. Este algoritmo de detección será aplicado posteriormente en el capítulo 6 con el objetivo de realizar una predicción adaptada a los perfiles de la imagen en el contexto de medias en celda.

Finalmente en la sección 4.7 explicamos el algoritmo de detección de contornos que será aplicado en la predicción multidireccional que expondremos en el capítulo 5 (en la sección 4.7 y capítulo 5 estamos en el contexto de valores puntuales). Para realizar dicha predicción, nos interesa que los contornos posean cierta “*continuidad*”, así que como resultado este algoritmo nos proporcionará los contornos que posean una longitud mínima.

4.2

Detección de contorno mediante filtros

Para detectar aquellos puntos que forman los perfiles de una imagen podemos emplear la siguiente definición:

Definición 4.1. *Un punto (x_0, y_0) de una imagen pertenece a un contorno si en este punto el modulo del gradiente de la intensidad lumínica $|\vec{\nabla}I|$ es máximo.*

Esta definición es la base de los detectores de contornos por gradiente, los cuales provienen de la discretización de las derivadas parciales de orden 1 de la imagen. En esta sección explicaremos la detección de contornos empleando los filtros de Roberts, Sobel y Prewitt. Al emplear cualquiera de estos filtros, se obtiene siempre como resultado final una imagen en la que cada píxel representará el valor absoluto del gradiente, correspondiendo las zonas más claras a los perfiles de la imagen.

4.2.1

Detector de contornos de Roberts

El detector de Roberts emplea las siguientes filtros de máscara para evaluar el gradiente:

$$G_x = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \quad G_y = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}.$$

Estas máscaras son aplicadas, de forma independiente, a la imagen inicial, obteniendo así en cada píxel aproximaciones a las componentes del gradiente. Sean $G_{x;i,j}$ y $G_{y;i,j}$ las aproximaciones obtenidas al aplicar G_x y G_y al píxel (i, j) , respectivamente. Así

tenemos que la aproximación valor absoluto del gradiente en dicho píxel viene dado por:

$$|G_{i,j}| = \sqrt{G_{x;i,j}^2 + G_{y;i,j}^2} \approx |G_{x;i,j}| + |G_{y;i,j}|.$$

La siguiente expresión nos proporciona la orientación del contorno:

$$\theta = \arctan\left(\frac{G_{y;i,j}}{G_{x;i,j}}\right) - \frac{\pi}{4}$$

A continuación mostramos como aplicaríamos las máscaras de convolución anteriores a una ventana 2×2 de la imagen con el objetivo de calcular la aproximación al gradiente en el píxel correspondiente:

$I_{1,2}$	$I_{2,2}$
$I_{1,1}$	$I_{2,1}$

en este caso tenemos $|G| = |I_{1,2} - I_{2,1}| + |I_{2,2} - I_{1,1}|$.

4.2.2

Detector de contornos de Sobel

El detector de contornos de Sobel emplea los siguientes filtros de máscara 3×3 :

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \quad G_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}.$$

Estas máscaras nos darán máximos locales al ser aplicada a píxeles que formen parte de contornos verticales u horizontales respectivamente. Del mismo modo que en el detector de Roberts, tenemos que la aproximación al gradiente en el píxel (i, j) viene dada por:

$$|G_{i,j}| = \sqrt{G_{x;i,j}^2 + G_{y;i,j}^2} \approx |G_{x;i,j}| + |G_{y;i,j}|,$$

y la orientación del perfil viene dada por θ :

$$\theta = \arctan\left(\frac{G_{y;i,j}}{G_{x;i,j}}\right) + \frac{\pi}{2}.$$

Por tanto, la aproximación del gradiente que obtenemos al aplicar las máscaras de Sobel a:

$I_{1,3}$	$I_{2,3}$	$I_{3,3}$
$I_{1,2}$	$I_{2,2}$	$I_{3,2}$
$I_{1,1}$	$I_{2,1}$	$I_{3,1}$

es:

$$|G| = |(I_{1,3}+2I_{2,3}+I_{3,3})-(I_{1,1}+2I_{2,1}+I_{3,1})| + |(I_{1,3}+2I_{1,2}+I_{1,1})-(I_{3,3}+2I_{3,2}+I_{3,1})|.$$

4.2.3

Detector de contornos Prewitt

El detector de contornos Prewitt, se diferencia del detector de contornos de Sobel en los filtros de máscara empleados. En este caso los filtros son:

$$G_x = \begin{bmatrix} 1 & 1 & 1 \\ 0 & 0 & 0 \\ -1 & -1 & -1 \end{bmatrix}, \quad G_y = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}.$$

La aproximación al gradiente obtenida es:

$$|G| = |(I_{1,3})-I_{(1,1)}+(I_{2,3}-I_{2,1})+(I_{3,3}-I_{3,1})| + |(I_{3,3}-I_{1,3})+(I_{3,2}-I_{1,2})+(I_{3,1}-I_{1,1})|.$$

4.3

Detector de contornos de Canny

J. Canny (ver [18]), propone suavizar la imagen aplicando un núcleo de convolución Θ_a (demuestra las ventajas de emplear un núcleo gaussiano; Θ_a es una gaussiana de varianza a). De este modo $I \star \Theta_a$, es diferenciable, y los puntos de los contornos son aquellos que maximizan $\vec{\nabla}(I \star \Theta_a)$, y por tanto a $I \star \vec{\nabla}\Theta_a$.

De este modo J. Canny propone la siguiente definición para los puntos pertenecientes a los contornos:

Definición 4.2. Un punto (x_0, y_0) pertenece a un contorno si el módulo del gradiente de la intensidad lumínica, suavizado por un núcleo Θ_a , $|\vec{\nabla}(I \star \Theta_a)|$, es localmente máximo en la dirección de $\vec{\nabla}(I \star \Theta_a)$

El detector de contornos de Canny, consta de las siguientes fases:

- **Obtención de gradientes y de la magnitud del gradiente.** Tras suavizar la imagen inicial por medio de una Gaussiana, Θ_a . Se obtiene el gradiente $\vec{\nabla}(I \star \Theta_a) = \left(\frac{\partial}{\partial x}(I \star \Theta_a), \frac{\partial}{\partial y}(I \star \Theta_a) \right)$. De este modo, la magnitud del gradiente y la dirección, θ , vienen dados por las expresiones:

$$|\vec{\nabla}(I \star \Theta_a)| = \sqrt{\left(\frac{\partial}{\partial x}(I \star \Theta_a) \right)^2 + \left(\frac{\partial}{\partial y}(I \star \Theta_a) \right)^2}$$

$$\theta = \arctan \frac{\frac{\partial}{\partial x}(I \star \Theta_a)}{\frac{\partial}{\partial y}(I \star \Theta_a)}$$

- **Supresión de no máximos.** Una vez determinado el valor del gradiente en cada punto, los contornos de la imagen estarán dentro del conjunto de puntos que posean una magnitud máxima. En este paso se pretende eliminar los “no máximos” perpendiculares al contorno.
- **Histéresis.** Una forma de obtener los contornos de una imagen consiste en quedarnos con aquellos puntos cuyo gradiente es superior a una cantidad dada. Canny utiliza dos parámetros t_a y t_b llamados parámetros de histéresis alto y bajo, respectivamente. De acuerdo con estos parámetros los puntos se clasifican de la siguiente manera:
 - Si el gradiente en el punto es superior a t_a , es aceptado inmediatamente.
 - Si el gradiente en el punto es inferior a t_b , es rechazado de forma directa.

- Si el valor está comprendido entre t_a y t_b , se marcará de forma temporal como posible integrante de un contorno. Para aceptarlo, debe existir un camino de puntos marcados como integrantes de un perfil desde él hasta un punto aceptado.

4.4

Resultados numéricos

A continuación mostramos los resultados obtenidos tras aplicar los métodos anteriores a las imágenes de la figura 4.1. La imagen *Peppers* es una imagen real que no posee mucha textura, la imagen *Geo* es una imagen puramente geométrica, mientras que *Geobl* es el resultado obtenido tras añadir textura a la imagen *Geo*.

La figura 4.2 muestra los resultados obtenidos tras aplicar los filtros de Sobel a las imágenes test. Para ello hemos fijado un parámetro T , marcando únicamente como puntos pertenecientes a un contorno aquellos que verifican $|G| > T$. En el caso de $T = 0$, el resultado muestra el valor del gradiente obtenido, de modo que un tono de gris más oscuro corresponde a un módulo del gradiente mayor. Notar que para eliminar aquellos píxeles detectados como contornos debido a la textura de la imagen es suficiente aumentar el valor del parámetro T . Hemos optado por omitir los resultados que se obtienen al aplicar los filtros de Roberts y Prewitt puesto que estos son similares.

La figura 4.3 muestra los contornos detectados por el método de Canny a las imágenes test. En ella podemos observar que mientras que en la imagen *Geo* detecta los contornos de modo perfecto, en las imágenes *Peppers* y *Geobl*, detecta píxeles que no pertenecen a los perfiles debido a la textura que poseen dichas imágenes. No obstante, estos contornos ficticios pueden ser eliminado variando los parámetros t_a y t_b .

Nuestro objetivo es utilizar la detección de contornos para realizar posteriormente una interpolación especial en ellos. De este modo nos interesa que los perfiles detectados tengan el menor grosor posible para así utilizarlas en las diferentes predicciones que

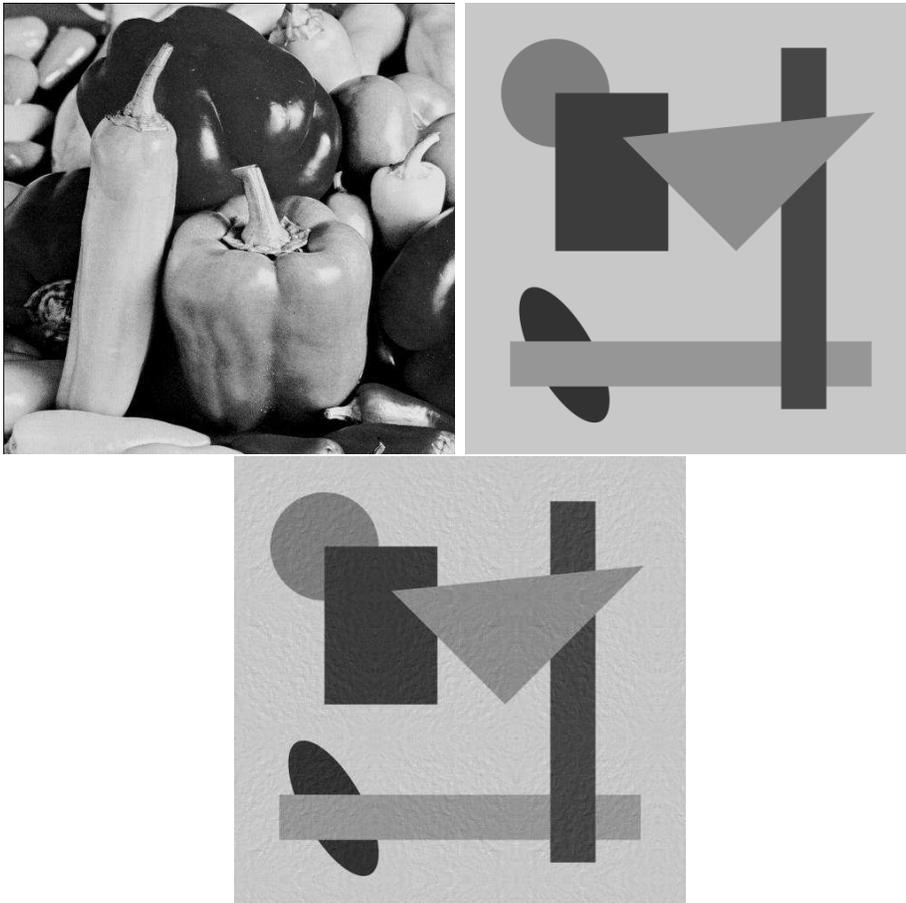


Figura 4.1: Arriba: imágenes Peppers y Geo. Abajo: imagen Geobl.

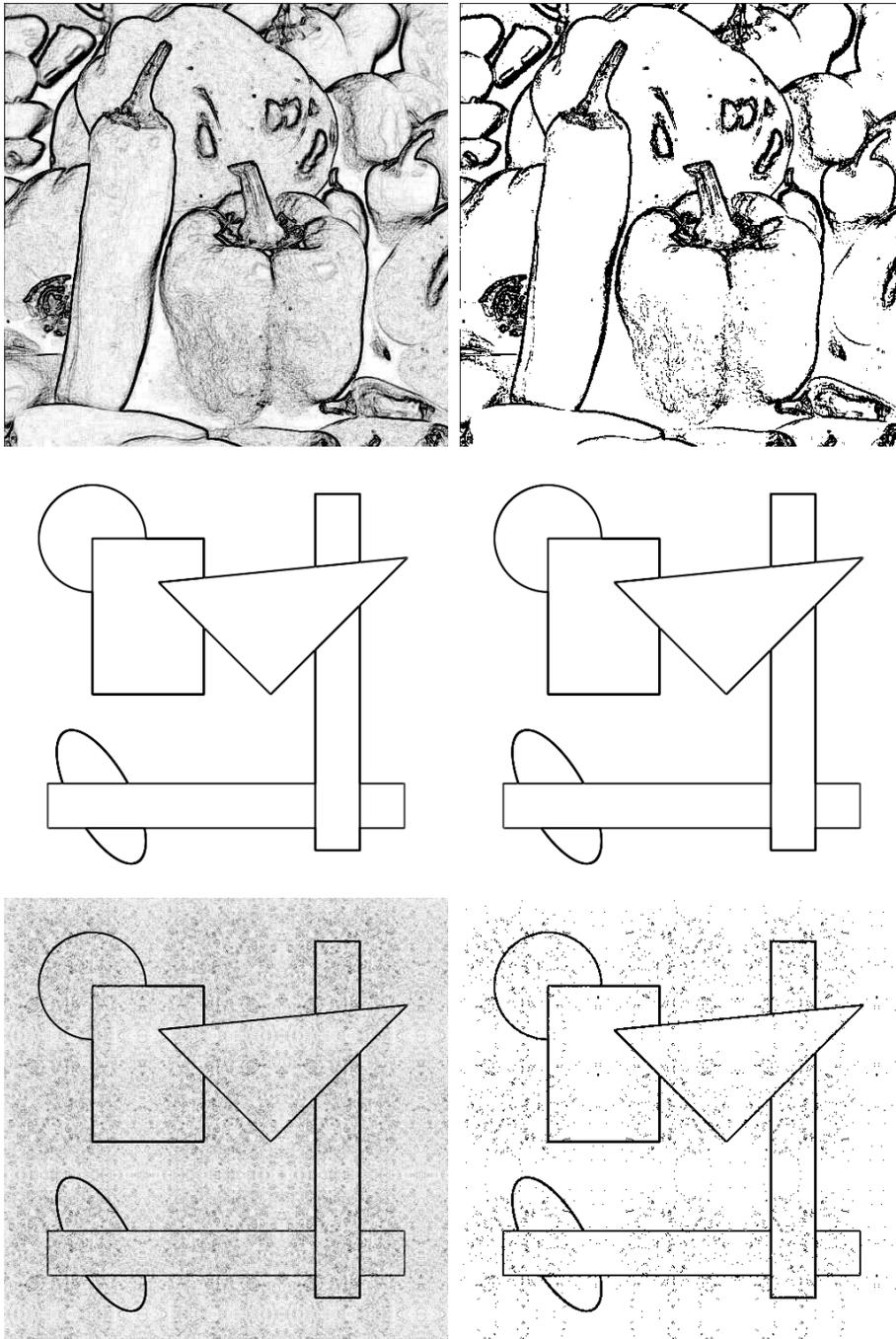


Figura 4.2: Contornos obtenidos filtros de Sobel a las imágenes Peppers, Geo y Geobl. Parámetros empleados: $T = 0, 100$.

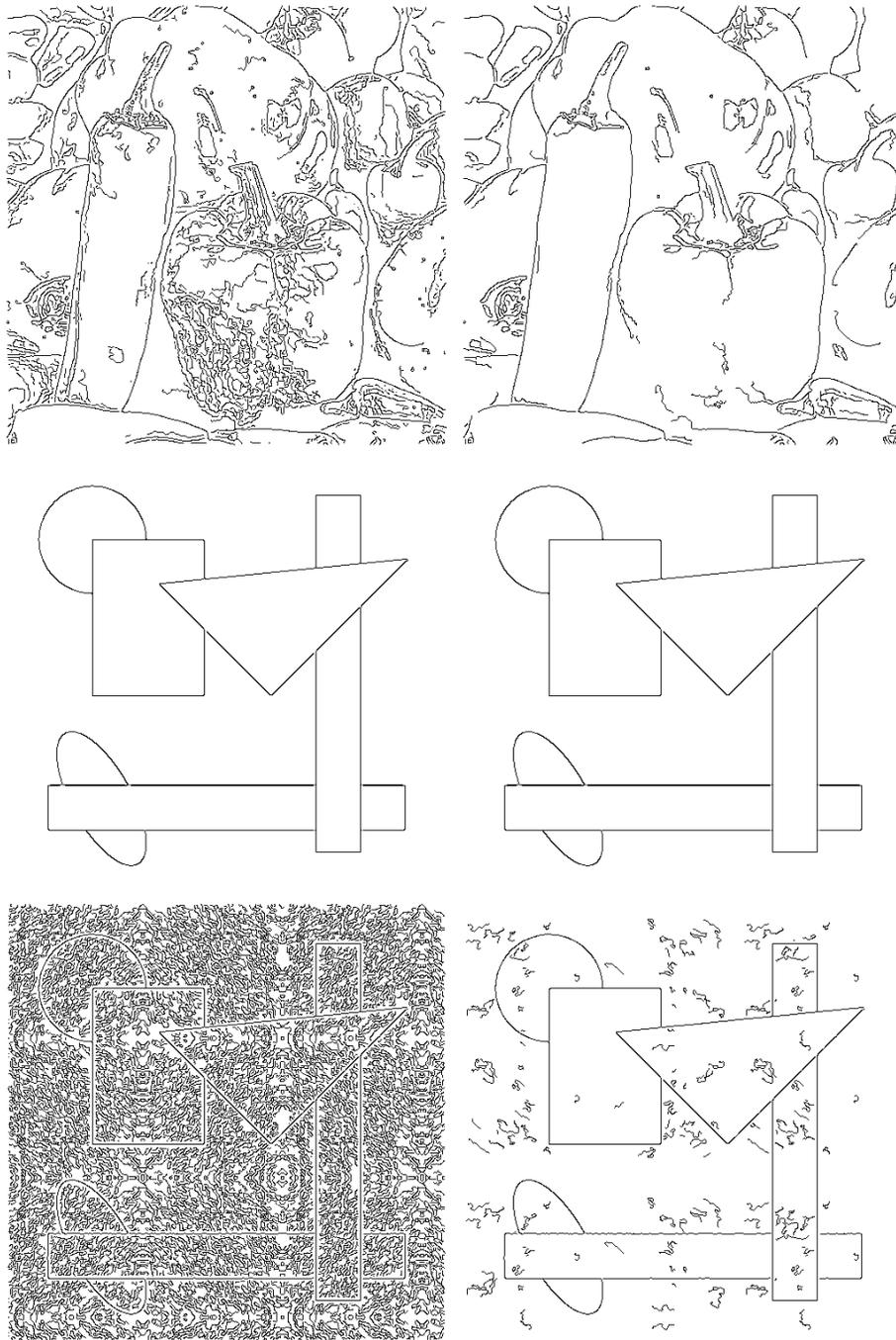


Figura 4.3: De izquierda a derecha. Peppers: $t_a = 1, t_b = 100$ $t_a = 50, t_b = 255$; Geo: $t_a = 1, t_b = 500$ $t_a = 20, t_b = 130$; Geobl: $t_a = 1, t_b = 50$ $t_a = 20, t_b = 130$

describiremos en los capítulos 5 y 6.

Los métodos anteriores presentan los siguientes inconvenientes:

- En caso de poseer las imágenes poca resolución, los perfiles detectados mediante la aplicación de los filtros de Roberts, Prewitt y Sobel, son demasiado gruesos.
- El hecho de que el detector de Canny suavice previamente la imagen, puede causar un desplazamiento de los contornos detectados (este desplazamiento puede evitarse mediante la detección multiescala de contornos, ver [35, 36, 37, 38]).

En la sección siguiente, describiremos como utilizar los Algoritmos 1.1 y 1.2 de selección del *stencil* para marcar los píxeles susceptibles de formar parte de un perfil. Este método, sin embargo, es muy sensible a la textura que posee la imagen. Con el objetivo de no detectar “falsos perfiles”, propondremos un detector de contornos que combinará la elección de *stencil*, junto con unos filtros de máscara similares a los de Sobel.

4.5

Detección utilizando la selección de *stencil*

Los algoritmos de selección de *stencil* 1.1 y 1.2 nos proporcionan otra forma de detectar los perfiles de una imagen. Supongamos que estamos en el caso uno-dimensional, y sean $\{\bar{f}_i\}_{i=1}^n$ las medias de una función, $f(x)$, en los intervalos $I_i = [x_{i-1}, x_i]$, $x_i = ih$, $i = 0, \dots, n$, $h = 1/n$. Debido a que dichos algoritmos intentan seleccionar *stencils* que no posean singularidades, si los *stencils* de los intervalos vecinos a I_i , i.e. I_{i-1} y I_{i+1} , no contienen al intervalo I_i podemos considerar que dicho intervalo posee una singularidad. Del mismo modo, si el único punto en común entre el *stencil* de I_{i-1} y I_i es x_{i-1} , podemos considerar que la singularidad se encuentra en dicho punto. Por tanto si \mathcal{S}_i es el *stencil* elegido para el intervalo

I_i bien con los Algoritmos 1.1 o 1.2, los intervalos que contienen singularidades pueden ser localizados con el siguiente algoritmo:

```
for  $i = 1, \dots, n$ 
  if  $x_i \notin \mathcal{S}_{i-1}$  y  $x_{i-1} \notin \mathcal{S}_{i+1}$ 
    Discontinuidad en  $I_i$ 
  else if  $\mathcal{S}_{i-1} \cap \mathcal{S}_i = \{x_{i-1}\}$ 
    Discontinuidad en  $\{x_{i-1}\}$ 
  end
end
```

Aplicando el algoritmo anterior tensorialmente a una imagen, obtenemos una aproximación a los contornos de ésta. La figura 4.4 muestra el resultado de aplicar este método a las imágenes de la figura 4.1. En ellos observamos que los perfiles son bien detectados si la imagen es puramente geométrica, pero en caso contrario detectamos demasiados "*falsos contornos*".

Recordemos que nuestro objetivo es obtener un mapa de los contornos de la imagen donde estos sean lo más delgados posibles, y además estén lo suficientemente aislados entre sí para asegurarnos el no tomar información de otro perfil en el caso de reconstruir una celda marcada por la detección.

En la siguiente sección proponemos un mecanismo de detección que empleará una modificación de los filtros de Sobel para marcar los contornos relevantes, así como para conocer también la dirección dominante del perfil, y además emplearemos las diferencias divididas de los datos para asegurarnos que los contornos detectados están lo suficientemente aislados los unos de los otros.

4.6

Un método nuevo para la detección de contornos

A continuación proponemos un método para localizar los perfiles de una imagen que consta de tres fases. En primer lugar,

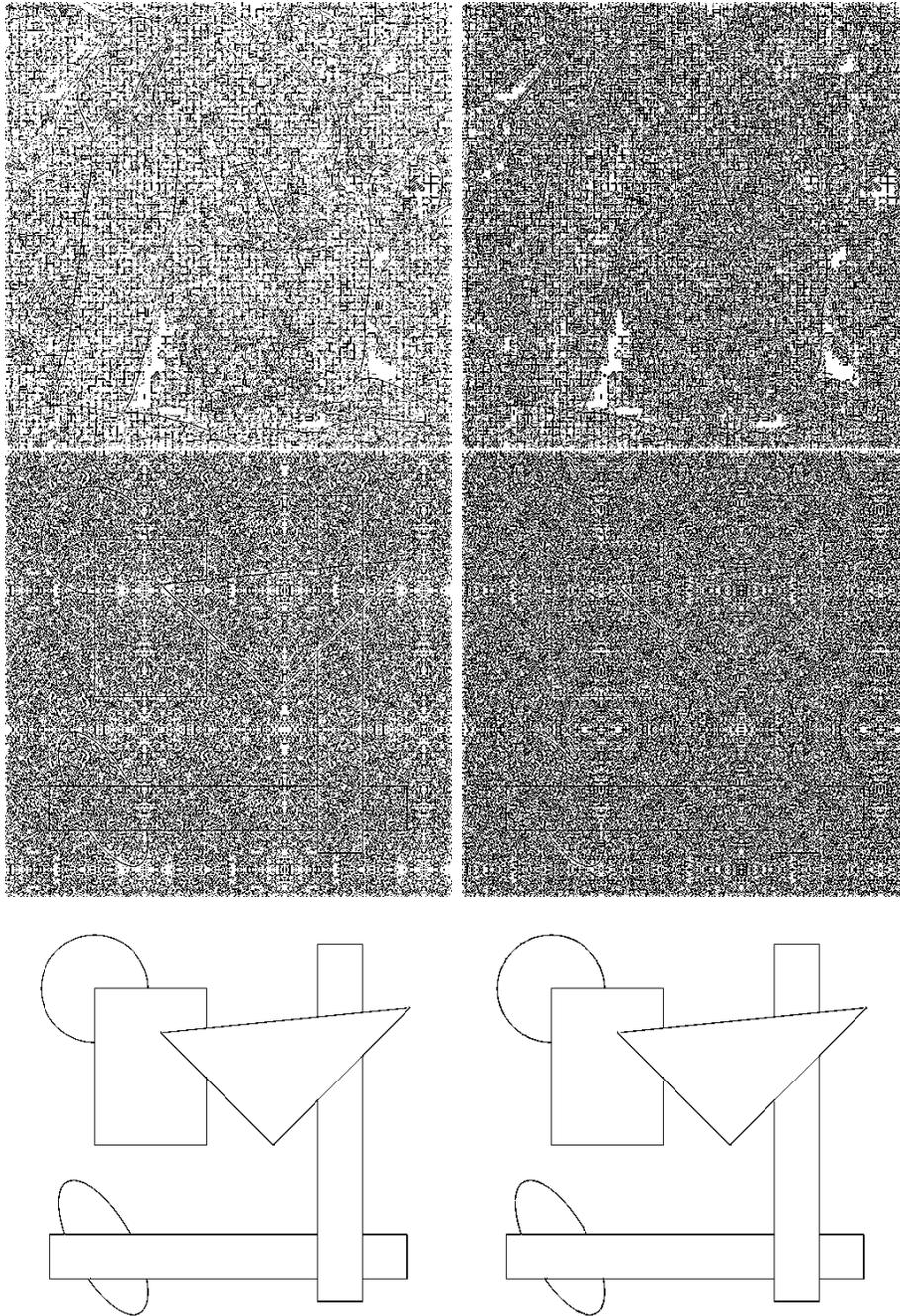


Figura 4.4: Celdas aisladas por los algoritmos de selección de stencil jerárquico y no jerárquico. Arriba: Peppers. Centro: Geo. Abajo: Geobl. Izquierda: Selección jerárquica. Derecha: Selección no jerárquica.

emplearemos las diferencias divididas de los datos para localizar los posibles contornos. Tras esta fase, en imágenes reales, muchas celdas son marcadas debido a la textura. Para descartarlas emplearemos una modificación de los filtros de máscara de Sobel. Para finalizar eliminaremos aquellas celdas marcadas como integrantes de algún perfil que no posean cierta “continuidad”.

Para explicar la primera fase supongamos que $\{\bar{f}_i\}_{i=1}^n$ representan las medias de una cierta función $f(x)$ definida en $[0, 1]$, en los intervalos $I_i = [x_{i-1}, x_i]$, es decir:

$$\bar{f}_i = \frac{1}{h} \int_{x_{i-1}}^{x_i} f(x) dx,$$

donde $x_i = ih$, $i = 0, \dots, n$, $nh = 1$.

Nuestro objetivo es marcar el intervalo $[x_{i-1}, x_i]$ si la discontinuidad se encuentra en el interior de I_i y marcar los intervalos I_{i-1} , I_i si la discontinuidad se encuentra en el nodo x_{i-1} (o próxima a éste).

Para ello veamos que si $f(x)$ es discontinua en $x_d \in [x_{i-1}, x_i] = I_i$, entonces

$$\bar{f}_{i+1} - \bar{f}_i = O([f]) \tag{4.1}$$

donde $[f] = f(x_d^+) - f(x_d^-)$. En efecto, si definimos $F(x) = \int_0^x f(x) dx$ tenemos

$$\bar{f}_{i+1} - \bar{f}_i = \frac{1}{h} \left(\int_{x_i}^{x_{i+1}} f(x) dx - \int_{x_{i-1}}^{x_i} f(x) dx \right) \tag{4.2}$$

$$= \frac{1}{h} \left(F(x_{i+1}) - 2F(x_i) + F(x_{i-1}) \right). \tag{4.3}$$

Si hacemos desarrollo de Taylor en x_d tendremos

$$\begin{aligned} \bar{f}_{i+1} - \bar{f}_i &= \frac{1}{h} \left((F(x_d^+) + (x_{i+1} - x_d)F'(x_d^+) + O(h^2)) \right. \\ &\quad - 2(F(x_d^+) + (x_i - x_d)F'(x_d^+) + O(h^2)) \\ &\quad \left. + ((F(x_d^-) + (x_{i-1} - x_d)F'(x_d^-) + O(h^2))) \right) \\ &= \frac{1}{h} \left((x_d - x_{i-1})(F'(x_d^+) - F'(x_d^-)) + O(h^2) \right) = O([f]). \end{aligned}$$

Por otra parte, si $f(x)$ es continua en $[x_{k-2}, x_k] = I_{k-1} \cup I_k$ no es difícil probar que

$$\bar{f}_k - \bar{f}_{k-1} = O(h) \tag{4.4}$$

Recopilando, podemos afirmar que si $f(x)$ posee una discontinuidad aislada cerca de x_{i-1} entonces $|\bar{f}_i - \bar{f}_{i-1}| > \max\{|\bar{f}_{i-1} - \bar{f}_{i-2}|, |\bar{f}_{i-2} - \bar{f}_{i-3}|, |\bar{f}_{i+2} - \bar{f}_{i+1}|, |\bar{f}_{i+3} - \bar{f}_{i+2}|\}$ y que si I_i posee una discontinuidad en su interior se debe verificar que $|\bar{f}_i - \bar{f}_{i-1}| > \max\{|\bar{f}_{i-1} - \bar{f}_{i-2}|, |\bar{f}_{i-2} - \bar{f}_{i-3}|\}$ y $|\bar{f}_{i+1} - \bar{f}_i| > \max\{|\bar{f}_{i+2} - \bar{f}_{i+1}|, |\bar{f}_{i+3} - \bar{f}_{i+2}|\}$.

Si a las condiciones anteriores, les añadimos condiciones adicionales para que entre dos intervalos marcados como susceptibles de contener una discontinuidad existan al menos tres intervalos no marcados obtenemos nuestro algoritmo de detección:

Algoritmo. 4.1. *Primera fase de detección*

```

 $c_1 = \max\{|\bar{f}_{i-1} - \bar{f}_{i-2}|, \dots, |\bar{f}_{i-3} - \bar{f}_{i-4}|\}$ 
 $c_2 = \max\{|\bar{f}_{i+2} - \bar{f}_{i+1}|, \dots, |\bar{f}_{i+4} - \bar{f}_{i+3}|\}$ 
 $m_1 = \max\{|\bar{f}_{i-1} - \bar{f}_{i-2}|, \dots, |\bar{f}_{i-4} - \bar{f}_{i-5}|\}$ 
 $m_2 = \max\{|\bar{f}_{i+1} - \bar{f}_i|, \dots, |\bar{f}_{i+4} - \bar{f}_{i+3}|\}$ 
If  $|\bar{f}_i - \bar{f}_{i-1}| > c_1$  y  $|\bar{f}_{i+1} - \bar{f}_i| > c_2$ 
  marcamos  $[x_{i-1}, x_i]$ 
elseif  $|\bar{f}_i - \bar{f}_{i-1}| > \max\{m_1, m_2\}$ 
  marcamos  $[x_{i-2}, x_{i-1}]$  y  $[x_{i-1}, x_i]$ 
end

```

Proposición 4.1. *Si con el algoritmo de detección anterior los intervalos $I_j = [x_{j-1}, x_j]$ y $I_{j+k} = [x_{j+k-1}, x_{j+k}]$ han sido marcados como susceptibles de contener una discontinuidad, con $k > 1$, entonces $k \geq 4$.*

Demostración. Usaremos la siguiente notación para diferenciar los dos casos por los que un intervalo puede haber sido marcado:

C1 Si $c_1 < |\bar{f}_i - \bar{f}_{i-1}|$ y $c_2 < |\bar{f}_{i+1} - \bar{f}_i|$, consideraremos que la discontinuidad se encuentra en I_i .

C2 Sino, si $\max\{m_1, m_2\} < |\bar{f}_i - \bar{f}_{i-1}|$, consideraremos que la discontinuidad se encuentra en x_{i-1} .

Supongamos que I_j ha sido marcado debido a que $c_1 < |\bar{f}_i - \bar{f}_{i-1}|$ y $c_2 < |\bar{f}_{i+1} - \bar{f}_i|$, así tenemos:

$$|\bar{f}_{j+1} - \bar{f}_j| > \max\{|\bar{f}_{j+2} - \bar{f}_{j+1}|, \dots, |\bar{f}_{j+4} - \bar{f}_{j+3}|\}. \quad (4.5)$$

En este caso, si el intervalo I_{j+k} ha sido marcado por C1 también, tenemos:

$$|\bar{f}_{j+k} - \bar{f}_{j+k-1}| > \text{máx}\{|\bar{f}_{j+k-1} - \bar{f}_{j+k-2}|, \dots, |\bar{f}_{j+k-3} - \bar{f}_{j+k-4}|\}. \quad (4.6)$$

En esta situación, si $k = 2, 3, 4$, tenemos que (4.5) y (4.6) entran en contradicción, luego $k > 4$. Si el intervalo I_{j+k} ha sido marcado por C2, entonces también está marcado I_{j+k-1} o I_{j+k+1} . Supongamos que han sido marcados I_{j+k-1} y I_{j+k} , obteniendo entonces (4.7),

$$|\bar{f}_{j+k} - \bar{f}_{j+k-1}| > \text{máx}\{|\bar{f}_{j+k-1} - \bar{f}_{j+k-2}|, \dots, |\bar{f}_{j+k-4} - \bar{f}_{j+k-5}|\}, \quad (4.7)$$

y como consecuencia $k > 4$.

Supongamos ahora, que el intervalo I_j ha sido marcado por C2. Si I_{j-1} ha sido también marcado, entonces tenemos

$$|\bar{f}_j - \bar{f}_{j-1}| > \text{máx}\{|\bar{f}_{j+1} - \bar{f}_j|, \dots, |\bar{f}_{j+4} - \bar{f}_{j+3}|\}, \quad (4.8)$$

si I_{j+k} está marcado por C1, obtenemos (4.6), y llegando a contradicción con $k = 2, 3$ y 4. Si I_{j+k} y I_{j+k+1} hubieran sido marcados por C2, tenemos que se debe verificar:

$$|\bar{f}_{j+k+1} - \bar{f}_{j+k}| > \text{máx}\{|\bar{f}_{j+k} - \bar{f}_{j+k-1}|, \dots, |\bar{f}_{j+k-3} - \bar{f}_{j+k-4}|\}, \quad (4.9)$$

entrando (4.9) en contradicción con (4.8) si $k = 1, 2$ y 3. Para finalizar, si I_j ha sido marcado por C2 y también lo ha I_{j+1} , se verifica

$$|\bar{f}_{j+1} - \bar{f}_j| > \text{máx}\{|\bar{f}_{j+2} - \bar{f}_{j+1}|, \dots, |\bar{f}_{j+5} - \bar{f}_{j+4}|\}, \quad (4.10)$$

si I_{j+k} está marcado por C1 se cumplirá (4.6), y por tanto k debe ser distinto de 2, 3 y 4. Si las marcadas son I_{j+k} y I_{j+k+1} (en el caso de ser marcadas I_{j+k-1} y I_{j+k} se haría con $k' = k - 1$), se cumple la relación (4.9), la cual nos obliga a que $k \neq 1, 2, 3, 4$. \square

Esto nos asegura la existencia de tres intervalos no marcados entre dos intervalos susceptibles de contener una discontinuidad, lo cual en el caso uno dimensional, nos permite el extrapolar el valor para un intervalo marcado a partir de tres intervalos que no han sido marcados como susceptibles de contener alguna discontinuidad.

En caso de estar interesados de detectar los perfiles de una imagen, aplicamos el algoritmo anterior de modo tensorial, es decir, si nuestra imagen consta de $n \times m$ píxeles, $\{I_{i,j}\}$ con $1 \leq i \leq n$, $1 \leq j \leq m$, donde $I_{i,j}$ será el valor del píxel (i, j) . Para cada $i \in \{0, \dots, m\}$, aplicamos el algoritmo a $\{I_{i,j}\}_{j=1}^m$. De este modo obtenemos una aproximación a los contornos horizontales de la imagen. Análogamente, para cada $j \in \{1, \dots, m\}$, podemos aplicar el algoritmo anterior a $\{I_{i,j}\}_{i=1}^n$, obteniendo por tanto una aproximación a los perfiles verticales. Nuestra primera aproximación a los contornos de la imagen estará formada por la unión de los contornos detectados fijando primero i y posteriormente j . Notar que los contornos diagonales de la imagen pueden ser detectados tanto al fijar i como j . Consideraremos que el valor $I_{i,j}$ representa la media de una cierta función $I(x, y)$ definida en $[0, 1] \times [0, 1]$ en la celda $C_{i,j} = [x_{i-1}, x_i] \times [y_{j-1}, y_j]$, es decir:

$$I_{i,j} = \frac{1}{|C_{i,j}|} \int_{C_{i,j}} I(x, y) dx dy = \frac{1}{(x_i - x_{i-1})(y_j - y_{j-1})} \int_{x_{i-1}}^{x_i} \int_{y_{j-1}}^{y_j} I(x, y) dx dy,$$

con $x_i = ih_x$, $i = 0, \dots, n$, $nh_x = 1$ y $y_j = jh_y$, $j = 0, \dots, m$, $mh_y = 1$.

Dado que normalmente las imágenes poseen cierta textura, tras aplicar la fase anterior tenemos muchas celdas marcadas. Algunas de estas pueden ser eliminadas aplicando los siguientes filtros de máscara,

$$S_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, S_y = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix}$$

$$S_{d_+} = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 0 & -1 \\ 0 & -1 & -2 \end{bmatrix}, S_{d_-} = \begin{bmatrix} 0 & 1 & 2 \\ -1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix}.$$

(4.11)

Así, si la celda $C_{i,j}$ ha sido marcada por la fase anterior como susceptible de poseer un contorno de la imagen, llamaremos $S_{x;i,j}$, $S_{y;i,j}$, $S_{d_+;i,j}$ y $S_{d_-;i,j}$ a los valores absolutos resultantes de aplicar los filtros de máscara anteriores a dicha celda, siendo dichos valores

los siguientes:

$$\begin{aligned}
 S_{x;i,j} &= |I_{i-1,j+1} + 2I_{i-1,j} + I_{i-1,j-1} - I_{i+1,j+1} - 2I_{i+1,j} - I_{i+1,j-1}|, \\
 S_{y;i,j} &= |I_{i-1,j+1} + 2I_{i,j+1} + I_{i+1,j+1} - I_{i-1,j-1} - 2I_{i,j-1} - I_{i+1,j-1}|, \\
 S_{d+;i,j} &= |I_{i-1,j} + 2I_{i-1,j+1} + I_{i,j+1} - I_{i,j-1} - 2I_{i+1,j-1} - I_{i+1,j}|, \\
 S_{d-;i,j} &= |I_{i-1,j} + 2I_{i-1,j-1} + I_{i,j-1} - I_{i,j+1} - 2I_{i+1,j+1} - I_{i+1,j}|,
 \end{aligned}$$

de modo de que si la relación siguiente no se verifica:

$$S_{i,j} = \text{máx} \{S_{x;i,j}, S_{y;i,j}, S_{d+;i,j}, S_{d-;i,j}\} > 0,5(M - m), \quad (4.12)$$

donde $M = \text{máx} \{I_{i,j}\}$ y $m = \text{mín} \{I_{i,j}\}$, $0 \leq i \leq n$, $0 \leq j \leq m$, consideraremos que la celda había sido marcada por causa de la textura y supondremos que por dicha celda no pasa ningún contorno. Además con los filtros anteriores también podemos determinar la orientación del contorno que pasa por la celda $C_{i,j}$. Así si $S_{i,j} = S_{x;i,j}$ diremos que el contorno posee orientación vertical en dicha celda, si $S_{i,j} = S_{y;i,j}$ diremos que la orientación es vertical, y si $S_{i,j} = S_{d+;i,j}$ o $S_{i,j} = S_{d-;i,j}$, diremos que la orientación es diagonal positiva o negativa respectivamente. Notar que la constante 0,5 que aparece en (4.12) ha sido determinada empíricamente, y su misión es controlar el número de celdas detectadas como pertenecientes a un contorno. Los artículos [41] y [47] muestran otras formas de clasificar los contornos según su orientación.

Para finalizar con la detección de contornos, eliminamos aquellas celdas que han sido detectadas como susceptibles de formar parte de un perfil de la imagen, pero sin embargo se encuentran “aisladas” de otras celdas marcadas como integrantes de contornos. Así, si la celda $C_{i,j}$ está marcada tras las dos fases anteriores, tomamos una ventana 5×5 centrada en dicha celda, de modo que si esta ventana posee menos de tres celdas marcadas como susceptibles de contener un perfil, consideraremos que por la celda $C_{i,j}$ no pasa ningún contorno, y dicha celda no estará marcada al finalizar la detección.

La figura 4.5 muestra el resultado de aplicar las diferentes fases a las imágenes *Peppers* y *Geobl*. La primera fila muestra los contornos detectados empleando las diferencias divididas, la segunda

fila muestra el resultado obtenido tras aplicar los filtros de máscara S_x , S_y , S_{d_+} y S_{d_-} , y en la última fila hemos eliminado aquellas celdas marcadas en las etapas anteriores que estaban aisladas.

El algoritmo de detección de contornos explicado en esta sección presenta las siguientes ventajas respecto a los citados anteriormente. Por una parte, al aplicar la proposición 4.1 obtenemos unos perfiles menos gruesos que los obtenidos al aplicar los filtros de máscara mencionados en la sección 4.2, además los filtros definidos en (4.11) nos permiten clasificar los píxeles pertenecientes a los contornos en cuatro grandes bloques según la orientación predominante en éstos (horizontal, vertical, diagonal positivo o negativo). Por otra parte, el hecho de no suavizar la imagen original, como con el detector de Canny, nos asegura una correcta localización de los perfiles de la imagen. No obstante, el principal inconveniente de este algoritmo se produce al detectar contornos en imágenes de baja resolución, pues el imponer condiciones para que entre los contorno exista una separación mínima puede provocar la eliminación de éstos. Este algoritmo de detección lo aplicaremos en el capítulo 6 con el objetivo de realizar una interpolación adaptada a los contornos de la imagen, en la cual nos será de gran ayuda el poder clasificar los perfiles según su orientación.

4.7

Detección de contornos continuos

En esta sección proponemos un algoritmo para la detección de contornos que presentan una longitud mínima. Este algoritmo de detección ha sido desarrollado en colaboración con J. Baccou y J. Liandrat de la Universidad de Provence (Aix - Marseille I). Los contornos obtenidos mediante este algoritmo serán empleados en el capítulo 5 para realizar una predicción multidireccional dependiente de los contornos, y su posterior aplicación a la compresión de imágenes (ver [5, 13]). En esta sección y en el capítulo 5 asumimos que los datos que tenemos son valores puntuales.

La detección de contornos está basada en la utilización de los filtros definidos en (4.11), y se divide en tres fases:

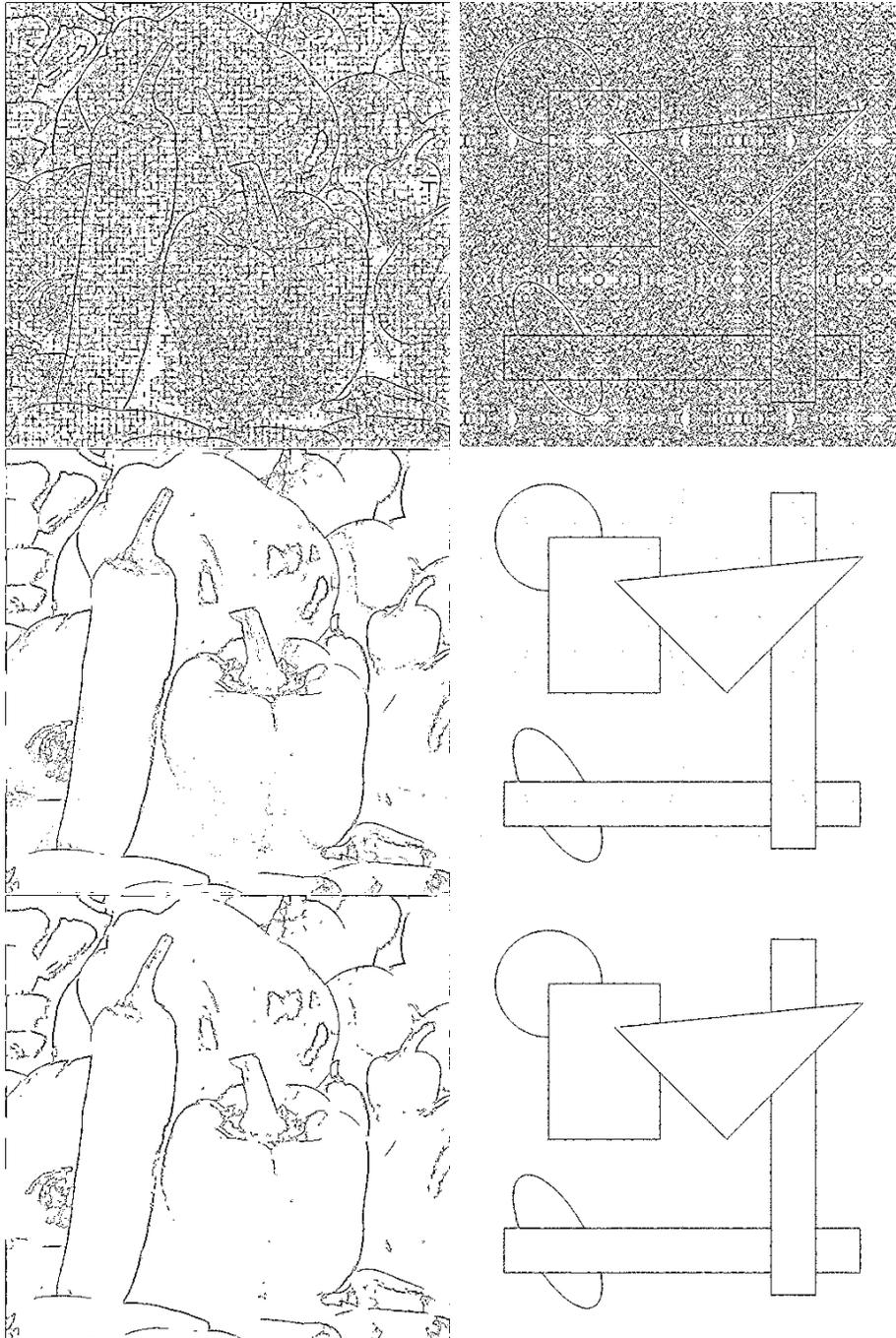


Figura 4.5: De arriba a abajo podemos observar el resultado de aplicar las diferentes fases a las imágenes "peppers" y "geobl". Arriba: diferencias divididas. Centro: filtros S_x , S_y , S_{d+} y S_{d-} . Abajo: resultado final.

Aplicación de los filtros de máscara de Sobel

En esta primera etapa aplicamos los filtros de máscara de Sobel S_x , S_y , S_{d_+} , S_{d_-} a cada píxel de la imagen. De este modo obtenemos máximos locales en aquellos píxeles que formen parte de contornos horizontales, verticales, diagonales positivos ó negativos respectivamente.

Los píxeles que verifiquen la relación (4.12) serán marcados, obteniendo así una matriz que nos indicará además de los posibles píxeles pertenecientes a los perfiles de la imagen, si éstos pertenecen a un contorno horizontal, vertical, diagonal positivo o negativo.

Reducción de las zonas detectadas

En la siguiente etapa, el objetivo es detectar entre los píxeles marcados anteriormente aquellos que estén comprendidos entre el “*principio*” y el “*final*” del contorno, reduciendo de este modo los píxeles marcados.

Así, si tenemos una secuencia vertical, $\{X_{n,m}\}_{m_1 \leq m \leq m_2}$, de longitud $m_2 - m_1$, donde todos los píxeles han sido marcados y poseen todos una orientación horizontal o diagonal, calcularemos las siguientes diferencias divididas para cada uno:

$$|I_{n,m-1} - 2 \cdot I_{n,m} + I_{n,m+1}|, \quad m_1 \leq m \leq m_2.$$

De modo que si $m'_1 \leq m'_2$ corresponden a los dos valores de diferencias máximas, reduciremos la secuencia de píxeles detectados a $\{X_{n,m}\}_{m'_1 \leq n \leq m'_2}$

Procediendo de modo análogo, reduciríamos las secuencias de píxeles detectados $\{X_{n,m}^J\}_{n_1 \leq n \leq n_2}$ con orientación vertical o diagonal.

Eliminación de los píxeles aislados y encadenado

Finalmente, encadenaríamos los píxeles detectados que son vecinos. De modo que si una cadena está constituida por un número de píxeles inferior a L_c , no es considerada como un contorno de la imagen, y dichos píxeles son eliminados de nuestro mapa de contornos.

Notar que tras esta última etapa podemos obtener una representación de los contornos de la imagen diferente a una matriz formada únicamente por 1 (píxeles pertenecientes a contornos) y 0 (píxeles restantes). Ya que para describir un contorno es suficiente con guardar las coordenadas del primer píxel, y posteriormente una serie de direcciones que nos indican donde se encuentra el píxel vecino que pertenece al contorno. Notar además, que el parámetro L_c puede ser modificado dependiendo de la naturaleza de la imagen.

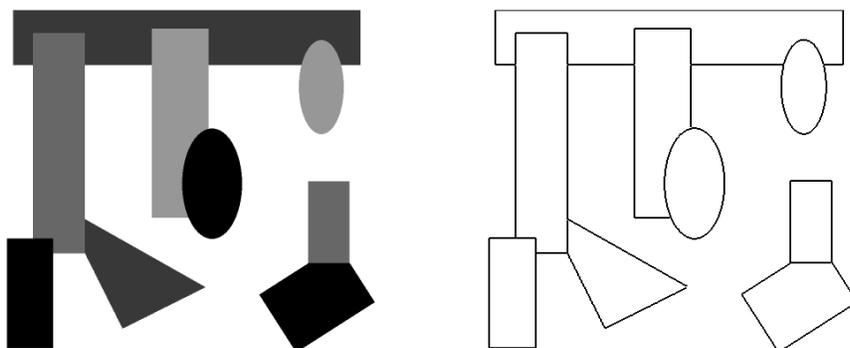


Figura 4.6: Contornos detectados para una imagen geométrica.

Las figuras 4.6 y 4.7 muestran los resultados obtenidos tras aplicar las tres etapas de detección. En la imagen geométrica, obtenemos prácticamente el mismo resultado que si aplicáramos únicamente los filtros de Sobel. Sin embargo en la imagen real, las

etapas 2 y 3 de la detección reducen considerablemente el número de píxeles detectados. Notar que el parámetro L_c empleados en el algoritmo de detección depende de la imagen a tratar.



Figura 4.7: De arriba izquierda a abajo derecha: Imagen original. Primera etapa (píxeles detectados = 25263). Segunda etapa (píxeles detectados = 18384). Tercera etapa (píxeles detectados = 6824)

5

Multirresolución 2D no separable para valores puntuales

5.1

Introducción

Los métodos basados en wavelets son unas herramientas eficaces para la compresión de imágenes ([36]). Por una parte, proporcionan algoritmos multiescala eficientes para la descomposición y reconstrucción de la imagen. Por otro lado, permiten una codificación eficiente gracias a la estructura de árbol de los coeficientes wavelet dominantes.

Sin embargo, estos métodos clásicos no consiguen resultados

completamente satisfactorios por dos principales motivos: en primer lugar hacen uso de una descomposición multiescala lineal y como consecuencia no se adaptan a las propiedades locales de la imagen (posición de los contornos) y en segundo lugar, la aplicación tensorial de la transformada wavelet, generalmente empleada, no resulta óptima para los contornos no horizontales ni verticales; como resultado, la tasa de compresión después de un truncamiento no lineal de los coeficientes wavelet no resulta óptima (ver [36]).

Con el objetivo de salvar estas desventajas, se han realizado trabajos recientes en los que las técnicas multiescala incorporan la geometría de la imagen como los Bandelets [40], Contourlets [25], etc. En este capítulo presentamos un algoritmo multiescala adaptado a los contornos, pero que permanece lineal, al menos parcialmente. Este trabajo [5] ha sido realizado en colaboración con J. Baccou y J. Liandrat de la Universidad de Provence (Marseille - I). El algoritmo consta de dos etapas,

1. Una detección de contornos, con la que obtenemos un mapa de puntos de segmentación.
2. Un algoritmo multiescala lineal, multidireccional, mapa-dependiente, basado en la multirresolución de Harten.

Una parte de este capítulo, como el diseño del operador predicción mapa-dependiente, ha sido desarrollada principalmente por Jean Baccou y también puede ser encontrado en su tesis [13]. Para que el capítulo sea más completo están detallados todos los pasos necesarios para construir el algoritmo de multirresolución 2D no separable para valores puntuales.

La estructura del capítulo es la siguiente: en la sección 5.2 describimos brevemente la multirresolución de Harten uno dimensional dependiente de puntos de segmentación y presentamos una nueva transformada multidireccional, mapa-dependiente. En la sección 5.3, describimos la técnica empleada para construir el mapa, basado en un simple algoritmo de detección de contornos. La técnica de codificación se presenta en la sección 5.4. La sección 5.5 está dedicada a las comparaciones numéricas entre nuestro algoritmo, y los resultados empleando interpolación ENO y lineal (i.e. no adaptada a los contornos de la imagen) en términos de

5. Multirresolución 2D no separable para valores puntuales 39

tasas de compresión y calidad de la imagen reconstruida. Para finalizar, la sección 5.6 muestra las conclusiones y perspectivas futuras.

5.2

Multirresolución de Harten dependiente

De ahora en adelante, consideraremos una malla uniforme $X^k = \{x_j^k, j \in \mathbb{Z}\}$. De este modo, para cada valor k , asociado a la escala 2^{-k} , x_j^k se define del siguiente modo: $x_j^k = j2^{-k}$.

5.2.1

Caso uno dimensional

De igual modo que en [10], [30] la multirresolución de Harten puede ser definida como un triplete $(V^k, D_k^{k-1}, P_{k-1}^k)$, $k \in \mathbb{Z}$. V_k es un espacio separable, donde k hace referencia al nivel de resolución y en el contexto del procesamiento de señales puede interpretarse como un conjunto de valores discretos, $f^k = \{f_j^k\}_{j \in \mathbb{Z}}$, en el nivel de resolución 2^{-k} , asociados a la señal, f , definidos en toda la recta real. D_k^{k-1} y P_{k-1}^k son los operadores decimación y predicción, que conectan dos niveles de resolución consecutivos, $K_0 \leq \dots \leq k \leq k+1 \leq \dots \leq K_{max}$. El operador decimación, D_k^{k-1} , es un operador lineal que va del espacio de mayor resolución, V^k , al de menor, V^{k-1} . Mientras que el operador predicción, P_{k-1}^k , va del espacio de menor resolución al de mayor.

El operador P_{k-1}^k define un esquema de subdivisión ([26]) actuando entre el nivel $k-1$ y k . De este modo, si escribimos:

$$\forall \{f_j^{k-1}\}_{j \in \mathbb{Z}} \in V_{k-1}, (P_{k-1}^k f^{k-1})_j = \sum_{m \in \mathbb{Z}} a_{j-2m}^{k,j} f_m^{k-1}, \quad (5.1)$$

entonces, los coeficientes $\{a_m^{k,j}\}_{m \in \mathbb{Z}}$, $(k, j) \in \mathbb{Z}^2$, pueden interpretarse como los filtros de un esquema de subdivisión. Si $\{a_m^{k,j}\}_{m \in \mathbb{Z}}$

son independientes de f^{k-1} , se dice que el esquema de subdivisión es lineal y además estacionario (respectivamente uniforme) si los coeficientes no dependen de k (respectivamente de j).

En general, en la multirresolución de Harten, no es necesario que P_{k-1}^k sea un operador lineal y en ese caso, el correspondiente esquema de subdivisión es no lineal.

Para cualquier señal, la discretización en V^k , f^k , puede ser sustituida por la discretización en V^{k-1} , f^{k-1} , más los coeficientes correspondientes a los *detalles* $d^{k-1} = \{d_j^{k-1}\}_{j \in \mathbb{Z}}$ definidos del modo siguiente: $\forall j \in \mathbb{Z}$, $d_j^{k-1} = f_j^k - (P_{k-1}^k f^{k-1})_j$.

Iterando este proceso, podemos definir dos algoritmos multiescala, asociados a los niveles $K_0 \leq k$: el algoritmo descomposición,

$$f^k \mapsto \{f^{K_0}, d^{K_0}, \dots, d^{k-1}\}$$

y el algoritmo reconstrucción,

$$\{f^{K_0}, d^{K_0}, \dots, d^{k-1}\} \mapsto f^k.$$

En el contexto de los valores puntuales donde $f_j^k = f(j2^{-k})$ y el operador D_{k+1}^k consiste simplemente en tomar los valores que ocupan la posición par, el operador P_k^{k+1} se construye a partir de una técnica interpolatoria. En este capítulo, emplearemos la interpolación de Lagrange de grado fijo D . Las aproximaciones al nivel de resolución $k+1$, son calculada empleando interpolantes de Lagrange, definidos a partir de los *stencils* asociados a cada par (k, j) como sigue:

Algoritmo. 5.1. Cálculo f^{k+1}

for $j \in \mathbb{Z}$

$$\hat{f}_{2j}^{k+1} = f_j^k$$

$$\hat{f}_{2j-1}^{k+1} = \sum_{m=-l_{k+1,2j-1}}^{r_{k+1,2j-1}-1} L_m^{l_{k+1,2j-1}, r_{k+1,2j-1}}(-1/2) f_{j+m}^k$$

end

con

$$L_m^{l_{k+1,2j-1}, r_{k+1,2j-1}}(x) = \prod_{n=-l_{k+1,2j-1}, n \neq m}^{r_{k+1,2j-1}-1} \frac{x-n}{m-n}. \quad (5.2)$$

5. Multirresolución 2D no separable para valores puntuales 41

Los enteros $l_{k+1,2j-1}$ (resp. $r_{k+1,2j-1}$) denotan el número de puntos a la izquierda (resp. derecha) del *stencil* empleado para la interpolación asociado al par (k, j) y a \hat{f}_{2j-1}^{k+1} . Dichos valores satisfacen $l_{k+1,2j-1} + r_{k+1,2j-1} = D + 1$.

A partir del algoritmo 5.1 y empleando la notación de la expresión (5.1), definimos la máscara asociada al esquema de subdivisión:

$$a^{k,2j} := \begin{cases} a_0^{k,2j} = 1, \\ a_m^{k,2j} = 0, \text{ si } m \neq 0, \end{cases} \quad (5.3)$$

y

$$a^{k,2j-1} := \begin{cases} a_{-2m-1}^{k,2j-1} = L_m^{l_{k,2j-1}, r_{k,2j-1}}(-1/2) \text{ si } m = -l_{j,2j-1}, \dots, r_{j,2j-1} - 1, \\ a_{2m}^{k,2j-1} = 0 \text{ en otro caso.} \end{cases} \quad (5.4)$$

De igual modo que en [14], consideramos tres estrategias naturales para definir, para cada par (k, j) , los valores $(l_{2j-1,k+1}, r_{2j-1,k+1})$: la primera consiste en tomar $l_{2j-1,k+1} = l$ y $r_{2j-1,k+1} = r = D - l + 1$, donde r y l son unos valores fijados. Esta estrategia nos conduce a la clásica predicción invariante por traslación. El correspondiente esquema de subdivisión es estacionario y uniforme (y por tanto lineal). En el caso de que $l = r$, se puede ver una correspondencia directa entre la multirresolución de Harten en el contexto de valores puntuales (especialmente el operador predicción) y el esquema interpolatorio de subdivisión de Deslauriers y Dubuc (ver [24]).

La segunda estrategia consiste en tomar $(l_{2j-1,k+1}, r_{2j-1,k+1})$ a partir de una familia de puntos de segmentación (llamada mapa) definida previamente sobre la recta real. Esto nos conduce a lo que llamaremos una predicción mapa-dependiente, que define un esquema de subdivisión lineal pero no estacionario y no uniforme.

La tercera estrategia consiste en definir $(l_{2j-1,k+1}, r_{2j-1,k+1})$ dependientes de f^k a partir de criterios locales. Esto nos lleva a una predicción dependiente de los datos y a un esquema de subdivisión no lineal. Un esquema interpolatorio clásico, que emplea esta estrategia, es la interpolación ENO (Sección 1.2 y [10, 29]). En este caso, la selección del *stencil* se hace a partir la regularidad local de los datos estimada a partir de las diferencias divididas.

A partir de aquí nos concentraremos en las estrategias lineales (estrategias 1 y 2) y principalmente en la dependiente del mapa.

A continuación proponemos la siguiente definición de *stencil*, asociado a la familia de puntos de segmentación separados¹ \mathcal{S}_S :

Definición 5.1. (*Dependencia del mapa*)

Dada una familia de puntos de segmentación \mathcal{S}_S , un *stencil* mapa-dependiente está controlado por:

- Un triplete (D, l, r) , $l \geq 0$, $r \geq 1$ y $l + r = D + 1$ tal que $(l_{k,2j-1}, r_{k,2j-1}) = (l, r)$ si $[x_{j-l}^{k-1}, x_{j+r-1}^{k-1}] \cap \mathcal{S}_S = \emptyset$.
- Una regla para seleccionar $(l_{k,2j-1}, r_{k,2j-1})$ si $[x_{j-l}^{k-1}, x_{j+r-1}^{k-1}] \cap \mathcal{S}_S \neq \emptyset$.

Con esta definición, el *stencil* empleado para la interpolación queda fijado por (l, r) para cualquier punto excepto en las proximidades de los puntos de segmentación, donde se aplicará la regla de selección. A continuación, especificamos la regla de selección.

Bajo el supuesto de que los puntos de segmentación están separados, entonces existe una escala en la cual cada punto de segmentación puede considerarse aislado respecta a la longitud del *stencil*. En este caso, consideramos la definición siguiente para un único punto de segmentación y_0 correspondiente a $\mathcal{S}_S = \{y_0\}$.

La regla de selección de *stencil* está diseñada con el objetivo de elegir *stencils* que eviten el punto de segmentación $\{y_0\}$. Así, dado (D, l, r) , definimos la regla de selección del siguiente modo:

¹Se dice que la familia de puntos \mathcal{S}_S es separada si y solo si $\exists \epsilon_{\mathcal{S}_S} > 0$ tal que $\forall x, y \in \mathcal{S}_S, x \neq y \Rightarrow |x - y| \geq \epsilon_{\mathcal{S}_S}$

5. Multirresolución 2D no separable para valores puntuales 43

$S_{0,4}$	$\frac{35}{16}$	$-\frac{35}{16}$	$\frac{21}{16}$	$-\frac{5}{16}$
$S_{1,3}$	$\frac{5}{16}$	$\frac{15}{16}$	$-\frac{5}{16}$	$\frac{1}{16}$
$S_{2,2}$	$-\frac{1}{16}$	$\frac{9}{16}$	$\frac{9}{16}$	$-\frac{1}{16}$

Tabla 5.1: Coeficientes del polinomio de Lagrange asociados al stencil $S_{l,r}$. Los stencils $S_{3,1}$ y $S_{4,0}$ se deducen por simetría.

Definición 5.2. (Selección del stencil)

Para todo $k \in \mathbb{Z}$, el índice j_{k-1} se define como el único entero verificando $y_0 \in [x_{j_{k-1}-1}^{k-1}, x_{j_{k-1}}^{k-1}[$. Para todo k y j tal que $y_0 \in [x_{j-l}^{k-1}, x_{j+r-1}^{k-1}[$:

■ Si $y_0 \in [x_{2j_{k-1}-2}^k, x_{2j_{k-1}-1}^k[$, entonces

$$\begin{cases} \text{Si } j < j_{k-1} \Rightarrow r_{k,2j-1} = \inf\{j_{k-1} - j, r\} \text{ y } l_{k,2j-1} = D + 1 - j_{k-1} + j, \\ \text{Si } j \geq j_{k-1} \Rightarrow r_{k,2j-1} = D + 1 + j_{k-1} - j \text{ y } l_{k,2j-1} = \inf\{j - j_{k-1}, l\}. \end{cases}$$

■ Si $y_0 \in [x_{2j_{k-1}-1}^k, x_{2j_{k-1}}^k[$, entonces

$$\begin{cases} \text{Si } j \leq j_{k-1} \Rightarrow r_{k,2j-1} = \inf\{j_{k-1} - j, r\} \text{ y } l_{k,2j-1} = D + 1 - j_{k-1} + j, \\ \text{Si } j > j_{k-1} \Rightarrow r_{k,2j-1} = D + 1 + j_{k-1} - j \text{ y } l_{k,2j-1} = \inf\{j - j_{k-1}, l\}. \end{cases}$$

Las figuras 5.1 y 5.2, y la tabla 5.1 muestran esta regla de selección cuando $D = 3$, $l = 2$ y $r = 2$.

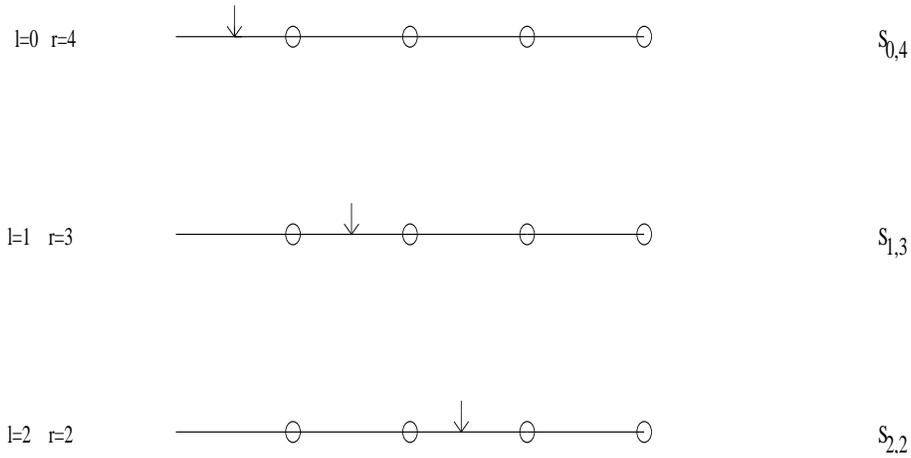


Figura 5.1: Posibles stencils para una interpolación con 4 nodos. $S_{l,r}$ hace referencia al stencil con l nodos a la izquierda y r nodos a la derecha.

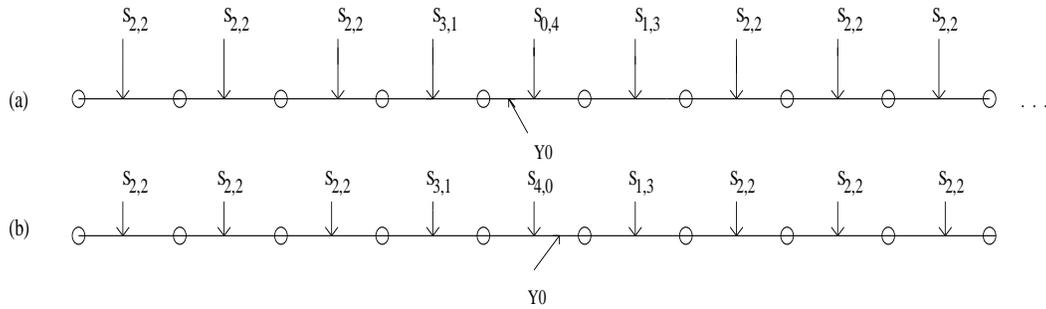


Figura 5.2: Ejemplo de selección de stencil a partir de la definición 5.2.
 (a) $y_0 \in [x_{2^{j_k-1}-2}^k, x_{2^{j_k-1}-1}^k[$, (b) $y_0 \in [x_{2^{j_k-1}-1}^k, x_{2^{j_k-1}}^k[$

Los algoritmos mapa-dependientes de multirresolución asociados al operador predicción definido previamente no encajan en aquellas familias clásicas de las que disponemos resultados generales sobre el análisis numérico de los esquemas de subdivisión ([26]). Sin embargo, una vez fijados los puntos de segmentación, el operador definido resulta ser lineal y no invariante por traslación. De este modo, los problemas de convergencia del esquema de subdivisión y estabilidad de la multirresolución permanecen relacionados. En [14] se prueba que el esquema de subdivisión definido previamente es uniformemente convergente en el espacio funcional $\mathcal{C}(\mathbb{R})$, donde f son funciones definidas y continuas en

5. Multirresolución 2D no separable para valores puntuales 45

$\mathbb{R} \setminus \mathcal{S}_f$, donde \mathcal{S}_f es un conjunto no vacío de números reales de modo que $\forall x \in \mathcal{S}_f, \lim_{y \rightarrow x, y > x} f(y) = f^+(x)$ y $\lim_{y \rightarrow x, y < x} f(y) = f^-(x)$. De un modo más preciso, la convergencia del esquema de subdivisión viene dada por la siguiente definición.

Definición 5.3. (Convergencia del esquema de subdivisión)

El esquema de subdivisión asociado a P_{k-1}^k se dice que es L^∞ -convergente si para cualquier sucesión de números reales $\{f_j^0\}_{j \in \mathbb{Z}} \in V^0$, existe una función $f \in \mathcal{C}(\mathbb{R})$ (llamada función límite asociada a f^0) de modo que $\forall \epsilon, \exists K$ tal que $\forall k \geq K$ verifica

$$\| P_{k-1}^k \dots P_1^2 P_0^1 f^0 - f^+ \left(\frac{\cdot}{2^k} \right) \|_\infty \leq \epsilon,$$

o

$$\| P_{k-1}^k \dots P_1^2 P_0^1 f^0 - f^- \left(\frac{\cdot}{2^k} \right) \|_\infty \leq \epsilon,$$

donde $\|\cdot\|_\infty$ representa la norma infinito.

La convergencia implica la existencia de funciones límite, siendo definidas como la función a la que converge el proceso de subdivisión empezando a partir de un único valor no nulo (igual a 1) en el nivel 0. Estas funciones se construyen por traslación de un conjunto finito de funciones de soporte compacto. De este modo, podemos obtener, del mismo modo que en [26] por ejemplo, que el esquema de subdivisión es estable y, gracias a la linealidad, que el esquema de multirresolución es estable. De un modo más preciso esto significa:

Definición 5.4. (Estabilidad de la multirresolución)

El **algoritmo de descomposición es estable** respecto a la norma $\| \cdot \|$ si:

$\exists C$ de modo que $\forall k, \forall (f^k, f_\epsilon^k)$, si $f^k \mapsto \{f^{K_0}, d^{K_0}, \dots, d^{k-1}\}$ y $f_\epsilon^k \mapsto \{f_\epsilon^{K_0}, d_\epsilon^{K_0}, \dots, d_\epsilon^{k-1}\}$, entonces,

$$\|f_\epsilon^{K_0} - f^{K_0}\| \leq C \|f_\epsilon^k - f^k\|, \text{ y } \forall m < k, \|d_\epsilon^m - d^m\| \leq C \|f_\epsilon^k - f^k\|.$$

El **algoritmo de reconstrucción es estable** respecto a la norma $\| \cdot \|$ si:

$\exists C$ de modo que $\forall k > K_0, \forall \{f^{K_0}, d^{K_0}, \dots, d^{k-1}\}$ y $\{f_\epsilon^{K_0}, d_\epsilon^{K_0}, \dots, d_\epsilon^{k-1}\}$, si $\{f^{K_0}, d^{K_0}, \dots, d^{k-1}\} \mapsto f^k$ y $\{f_\epsilon^{K_0}, d_\epsilon^{K_0}, \dots, d_\epsilon^{k-1}\} \mapsto f_\epsilon^k$, entonces,

$$\|f_\epsilon^k - f^k\| \leq C \sup (\|f_\epsilon^{k-1} - f^{k-1}\|, \|d_\epsilon^{k-1} - d^{k-1}\|).$$

La cuestión de la estabilidad es crucial en caso de emplear el esquema de multirresolución en el contexto de compresión de señales. Así, la diferencia entre la imagen inicial y la reconstruida después de la compresión viene controlada por las constantes de estabilidad.

Nota 5.1. La linealidad del esquema de subdivisión mapa-dependiente juega un papel clave para asegurar su estabilidad. No resulta de igual modo si consideramos esquema de tipo ENO [10, 23, 29]. En dichos esquemas, el "mejor" stencil para realizar la interpolación de Lagrange se obtiene a partir del cálculo de las diferencias divididas en cada nivel de resolución de los datos reconstruidos. Una perturbación de los datos (como resultado del truncamiento de los detalles) puede modificar la predicción seleccionando distintos stencils que los iniciales y provocar inestabilidades (ver figura 5.12)).

5. Multirresolución 2D no separable para valores puntuales 47

Si no se puede garantizar la estabilidad de la multirresolución se debe utilizar el algoritmo de control del error [1].

5.2.2

Predicción 2D mapa-dependiente

Consideremos los datos vienen dados en forma de matriz, i.e. $\{f_{i,j}^k\}_{(i,j) \in \mathbb{Z}^2}$. Así podemos construir una predicción 2D bidireccional como una generalización del algoritmo 5.1. En este caso, los puntos de segmentación son reemplazados por curvas de segmentación las cuales a partir de intersecciones de rectas verticales y horizontales dan lugar a puntos de segmentación. Como consecuencia de la curvatura de las curvas de segmentación podemos obtener puntos no aislados. En nuestro contexto, es posible la no adaptación en las cercanías de dichos puntos (ver Sección 5.2.2, predicción bidireccional). Sin embargo, si suponemos que las curvas de segmentación son Lipstchitz a trozos, el número de puntos no separados permanece acotado.

Para empezar, consideraremos que las curvas de segmentación vienen definidas por una familia de curvas, $\{C_i\}_{i \in I}$.

Predicción bidireccional mapa dependiente

En este caso, el algoritmo 5.1 es aplicado recursivamente a las filas y columnas de la matriz. De este modo, generalizando la notación del Sección 5.2.1 al caso 2D, introducimos el siguiente algoritmo de predicción en 2D:

Algoritmo. 5.2. Cálculo f^{k+1}

$$\begin{aligned} \text{for } (i, j) \in \mathbb{Z}^2 \\ \hat{f}_{2i,2j}^{k+1} &= \hat{f}_{i,j}^k, \\ \hat{f}_{2i,2j-1}^{k+1} &= \sum_{n=-l_{k+1,2i,2j-1}^H}^{r_{k+1,2i,2j-1}^H-1} L_n^{l_{k+1,2i,2j-1}^H, r_{k+1,2i,2j-1}^H} (-1/2) f_{i,j+n}^k, \\ \hat{f}_{2i-1,2j}^{k+1} &= \sum_{m=-l_{k+1,2i-1,2j}^V}^{r_{k+1,2i-1,2j}^V-1} L_m^{l_{k+1,2i-1,2j}^V, r_{k+1,2i-1,2j}^V} (-1/2) f_{i+m,j}^k, \end{aligned}$$

$$\hat{f}_{2i-1,2j-1}^{k+1} = \sum_{m=-l_{k+1,2i-1,2j-1}^V}^{r_{k+1,2i-1,2j-1}^V-1} L_m^{l_{k+1,2i-1,2j-1}^V, r_{k+1,2i-1,2j-1}^V} (-1/2) \hat{f}_{2i+2m,2j-1}^{k+1}.$$

end

El superíndice H (resp. V) hace referencia a *stencils* horizontales (resp. verticales). El número de nodos a la izquierda y derecha del *stencil* está en función de la familia de curvas de segmentación siguiendo la regla uno-dimensional cuando los nodos están separados. En caso contrario, el número de nodos a la izquierda (resp. derecha) es tomado a partir de un valor de referencia l (resp. r). Notar que en caso de que el conjunto de curvas sea vacío, la predicción bidireccional mapa-dependiente coincide con el producto tensorial clásico [1]. En caso de asociar un mapa a una familia de curvas de segmentación (ver sección 5.3) diremos que el algoritmo de multirresolución es mapa-dependiente.

Notar que la predicción anterior no tiene en cuenta la orientación de las curvas. Por este hecho, introducimos una generalización del caso 1D que incorpora una dependencia del mapa multidireccional.

Predicción mapa-dependiente multidireccional

El operador predicción mapa-dependiente multidireccional se construye como una modificación del bidireccional definido previamente, incorporando la orientación local del mapa [13]. De este modo, nos centramos en cuatro posibles direcciones para la predicción (una horizontal, una vertical y dos diagonales). Dado que los valores asociados a $\hat{f}_{2i-1,2j}^{k+1}$ y $\hat{f}_{2i,2j-1}^{k+1}$ no pueden ser fácilmente calculados a partir de interpolación diagonal empleando $\{f_{i,j}^k\}_{(i,j) \in \mathbb{Z}^2}$, la diferencia con el algoritmo 5.2 está en la predicción de los valores $\hat{f}_{2i-1,2j-1}^{k+1}$ la cual es realizada empleando, en caso necesario, interpolaciones diagonales de acuerdo con la orientación del mapa (ver figura 5.3).

En la sección 5.3 explicamos como obtenemos un mapa de puntos de segmentación para un nivel de resolución dado, y como a partir de este mapa obtenemos los mapas asociados a niveles de resolución inferiores. Supongamos ahora que disponemos de los

5. Multirresolución 2D no separable para valores puntuales 49

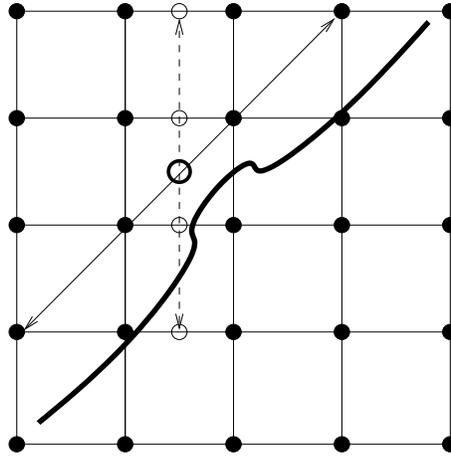


Figura 5.3: Dirección de la predicción para un punto del tipo $\hat{f}_{2i-1,2j-1}^{k+1}$. La línea discontinua indica la dirección empleada con la estrategia bidireccional, mientras que la línea continua indica la empleada con la predicción multidireccional.

mapas $\{map^k\}_{k=k_0}^{k_{max}}$, donde

$$map^k(i, j) = \begin{cases} 1, & \text{si } (x_i^k, y_j^k) \text{ es un punto de segmentación,} \\ 0, & \text{en caso contrario.} \end{cases}$$

Nuestro objetivo es aplicar una predicción mapa-dependiente para comprimir imágenes. En este contexto, el mapa que contiene los puntos de segmentación representa los contornos de la imagen. De ahora en adelante, supongamos que nuestro mapa es ideal (es decir que los contornos están contenidos entre dos puntos de segmentación consecutivos), y que conocemos los mapas de los puntos de segmentación en cada nivel de resolución. Bajo estos supuestos y haciendo uso de *stencils* formados por cuatro nodos, veamos como obtener la predicción para aquellos valores cuyo *stencil* contiene un nodo que pertenece a un punto de segmentación.

Veamos en primer lugar como obtener los valores $\hat{f}_{2i,2j-1}^k$ (los valores $\hat{f}_{2i-1,2j}^k$ se obtienen por el mismo procedimiento intercambiando j y i).

Dado que el mapa se supone ideal, existen un número finito de

configuraciones posibles para los nodos marcados alrededor del punto a predecir. En este caso las posibilidades son las siguientes:

Primer caso: $map^k(i, j - 1) + map^k(i, j) \neq 2$.

El segmento $[x_{i,j-1}^k, x_{i,j}^k]$ no contiene ningún contorno:

- Si $map^k(i, j - 1) = 0$ y $map^k(i, j) = 1$, obtenemos $\hat{f}_{2i,2j-1}^{k+1}$ empleando un *stencil* decentrado a la izquierda,

$$\hat{f}_{2i,2j-1}^{k+1} = \sum_{i=-3}^0 L_i^{3,1}(-1/2) f_{i,j+i}^k \quad (5.5)$$

- Si $map^k(i, j - 1) = 1$ y $map^k(i, j) = 0$, obtenemos $\hat{f}_{2i,2j-1}^{k+1}$ empleando un *stencil* decentrado a la derecha,

$$\hat{f}_{2i,2j-1}^{k+1} = \sum_{i=-1}^2 L_i^{1,3}(-1/2) f_{i,j+i}^k \quad (5.6)$$

La figura 5.4 muestra las dos configuraciones posibles para el primer caso.

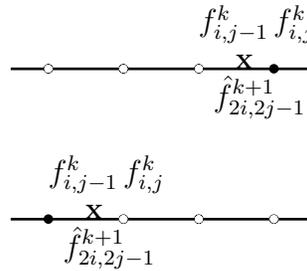


Figura 5.4: Configuraciones posibles para el cálculo de $\hat{f}_{2i,2j-1}^{k+1}$. Los puntos negros representan los puntos de segmentación.

Segundo caso: $map^k(i, j - 1) + map^k(i, j) = 2$.

En este caso necesitamos información del nivel $k + 1$.

5. Multirresolución 2D no separable para valores puntuales 51

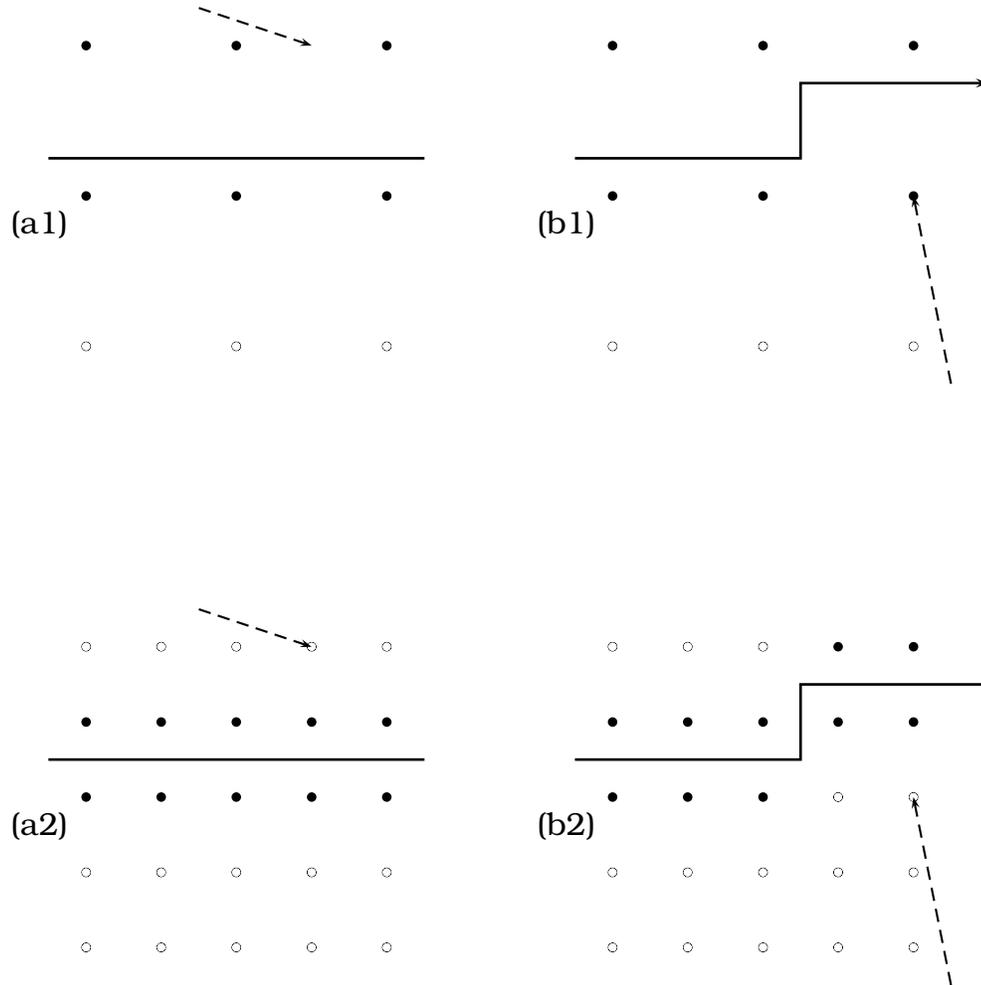


Figura 5.5: Configuraciones correspondientes a $map^{k+1}(2i, 2j - 2) = 0$ y $map^{k+1}(2i, 2j) = 0$. (a1)-(a2) corresponde un contorno horizontal, (b1)-(b2) a una esquina. La línea continua representa el contorno, mientras que la flecha indica el segmento que contiene el punto a predecir.

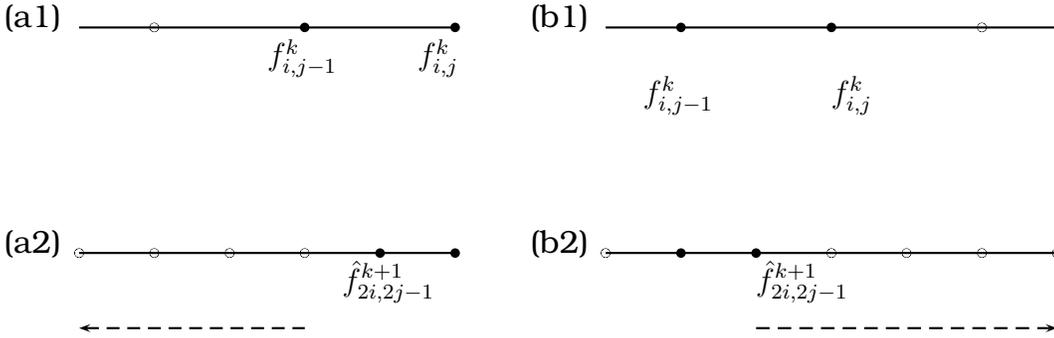


Figura 5.6: Configuraciones posibles para el caso en que $map^{k+1}(2i, 2j - 2) + map^{k+1}(2i, 2j) = 1$. (a1)-(b1) map^k , (a2)-(b2) map^{k+1} . La flecha indica el sentido de predicción.

- Si $map^{k+1}(2i, 2j - 2) + map^{k+1}(2i, 2j) = 0$ (ver figura 5.5). Empleando una interpolación centrada con dos nodos nos permite evitar el contorno, i.e.

$$\hat{f}_{2i,2j-1}^{k+1} = \frac{f_{i,j-1}^k + f_{i,j}^k}{2}$$

- Si $map^{k+1}(2i, 2j - 2) + map^{k+1}(2i, 2j) = 1$:
 - Si $map^{k+1}(2i, 2j - 2) = 1$ y $map^{k+1}(2i, 2j) = 0$ (figura 5.6 (b1)-(b2)), entonces $map^{k+1}(2i, 2j - 1) = 1$ y por tanto empleamos una extrapolación decentrada a derecha,

$$\hat{f}_{2i,2j-1}^{k+1} = \sum_{i=0}^3 L_i^{0,4}(-1/2) f_{i,j+i}^k. \quad (5.7)$$

- Si $map^{k+1}(2i, 2j - 2) = 0$ y $map^{k+1}(2i, 2j) = 1$ (figura 5.6 (a1)-(a2)), entonces $map^{k+1}(2i, 2j - 1) = 1$ y por tanto empleamos una extrapolación decentrada a izquierda,

$$\hat{f}_{2i,2j-1}^{k+1} = \sum_{i=-4}^{-1} L_i^{4,0}(-1/2) f_{i,j+i}^k. \quad (5.8)$$

- Si $map^{k+1}(2i, 2j - 2) + map^{k+1}(2i, 2j) = 2$: entonces el punto a predecir se trata de una esquina.

5. Multirresolución 2D no separable para valores puntuales 53

Dado que el mapa de puntos de segmentación se supone ideal, una esquina sólo puede darse en dos configuraciones. La figura 5.7 muestra una configuración, la otra se obtiene por medio de una rotación de ángulo π . La información dada por el mapa en el entorno del punto nos permite seleccionar una dirección para la predicción. Además, en caso de que $map^{k+1}(2i-1, 2j) = 0$ y $map^{k+1}(2i+1, 2j-2) = 0$, los valores siguientes son evaluados:

$$\begin{aligned} S_R &= map^{k+1}(2i, 2j) + map^{k+1}(2i-1, 2j) + \\ &\quad + map^{k+1}(2i-1, 2j-1), \\ S_L &= map^{k+1}(2i, 2j-2) + map^{k+1}(2i+1, 2j-2) + \\ &\quad + map^{k+1}(2i+1, 2j-1). \end{aligned}$$

Nuestro objetivo es seleccionar el entorno más regular, así si $S_R > S_L$ (5.7) (resp. $S_L > S_R$), emplearemos extrapolación a izquierda (resp. extrapolación a derecha 5.6).

Del igual modo, si $map^{k+1}(2i-1, 2j-2) = 0$ y $map^{k+1}(2i+1, 2j) = 0$, evaluamos:

$$\begin{aligned} S_R &= map^{k+1}(2i, 2j) + map^{k+1}(2i+1, 2j) + \\ &\quad + map^{k+1}(2i+1, 2j-1), \\ S_L &= map^{k+1}(2i, 2j-2) + map^{k+1}(2i-1, 2j-2) + \\ &\quad + map^{k+1}(2i-1, 2j-1), \end{aligned}$$

y seleccionamos el entorno más regular: si $S_R > S_L$ (resp. $S_L > S_R$), emplearemos extrapolación a izquierda (5.7) (resp. extrapolación a derecha (5.6)).

- Si el punto a predecir no pertenece a una esquina (figura 5.7 (b)), $\hat{f}_{2i,2j-1}^{k+1}$ es calculado del siguiente modo:

$$\hat{f}_{2i,2j-1}^{k+1} = \frac{f_{i,j-1}^k + f_{i,j}^k}{2} \quad (5.9)$$

Veamos ahora como obtener los valores $\hat{f}_{2i-1,2j-1}^{k+1}$. En este caso, la predicción tiene en cuenta la orientación de los contornos. De nuevo se presentan dos posibilidades:

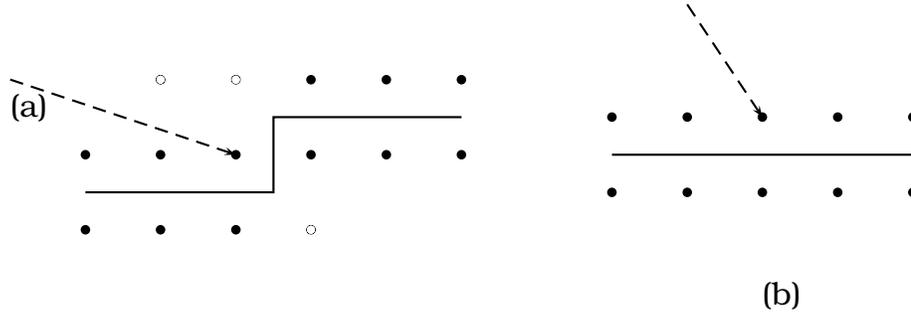


Figura 5.7: Configuraciones cuando el punto a predecir está en una esquina (a) o, en un contorno horizontal (b).

Primer caso: $map^k(2i-2, 2j-2) + map^k(2i-2, 2j) + map^k(2i-2, 2j) + map^k(2i, 2j) \neq 4$:

La distribución de los puntos marcados alrededor del punto a predecir nos proporciona información sobre la orientación del contorno en las cercanías del punto. En este caso hay ocho posibles configuraciones: dos corresponden a un perfil horizontal (figura 5.8 (a1)-(a2)), dos a perfiles verticales (figura 5.8 (b1)-(b2)) y cuatro a contornos diagonales (figura 5.8 (c1)-(c2)-(c3)-(c4)).

Así, $\hat{f}_{2i-1, 2j-1}^{k+1}$ se calcula del siguiente modo:

$$\text{Casos (a1),(a2): } \hat{f}_{2i-1, 2j-1}^{k+1} = \frac{\hat{f}_{2i-1, 2j-2}^{k+1} + \hat{f}_{2i-1, 2j}^{k+1}}{2}$$

$$\text{Casos (b1),(b2): } \hat{f}_{2i-1, 2j-1}^{k+1} = \frac{\hat{f}_{2i-2, 2j-1}^{k+1} + \hat{f}_{2i, 2j-1}^{k+1}}{2}$$

$$\text{Casos (c1),(c2): } \hat{f}_{2i-1, 2j-1}^{k+1} = \frac{\hat{f}_{2i-2, 2j-2}^{k+1} + \hat{f}_{2i, 2j}^{k+1}}{2}$$

$$\text{Casos (c3),(c4): } \hat{f}_{2i-1, 2j-1}^{k+1} = \frac{\hat{f}_{2i-2, 2j}^{k+1} + \hat{f}_{2i, 2j-2}^{k+1}}{2}$$

Segundo caso: $map^k(2i-2, 2j-2) + map^k(2i-2, 2j) + map^k(2i-2, 2j) + map^k(2i, 2j) = 4$:

Dado que esta configuración no proporciona información sobre la orientación del contorno, aplicamos una predicción bidireccional en dirección vertical (de igual modo que en los valores $\hat{f}_{2i-1, 2j}^{k+1}$).

5. Multirresolución 2D no separable para valores puntuales 55

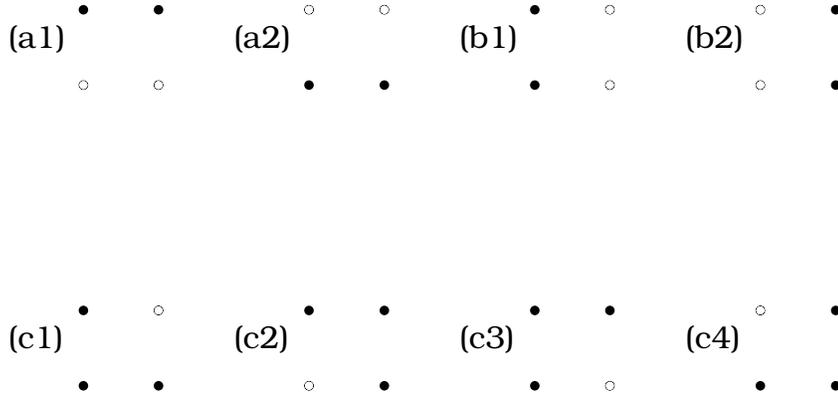


Figura 5.8: Configuraciones posibles para la predicción de $\hat{f}_{2^{i-1}, 2^{j-1}}^{k+1}$.

5.3

Construcción del mapa

De ahora en adelante, consideremos imágenes de tamaño $2^{k_{max}} \times 2^{k_{max}}$, y por tanto $(i, j) \in \{0, \dots, 2^{k_{max}-1}\}$.

Supongamos también que los contornos de la imagen coinciden con curvas de segmentación, $\{C_i\}_{i=1, \dots, K}$, y el mapa de los contornos en el nivel de resolución k (ver figura 5.9(a)) es una matriz binaria, $\{map^k(i, j)\}_{0 \leq i, j < 2^k-1}$, de modo que $map^k(i, j) = 1$ si la intersección de longitud 2^{-k+1} centrada en el punto $(i2^{-k}, j2^{-k})$ intersecciona las curvas de segmentación. Es decir, $map^k(i, j) = 1$ si y solo si $\exists 1 \leq i \leq K$ tal que $C_i \cap (((i-1)2^{-k}, j2^{-k}); ((i+1)2^{-k}, j2^{-k})) \cup ((i2^{-k}, (j-1)2^{-k}); (i2^{-k}, (j+1)2^{-k})) \neq \emptyset$.

Es evidente que un simple análisis de la matriz map^k proporciona los intervalos diádicos que interseccionan las curvas de segmentación y por tanto determina la selección del *stencil*.

En la práctica, el mapa de contornos es construido para el nivel de resolución máxima k_{max} . Posteriormente, a partir de $map^{k_{max}}$ se deduce el mapa para cualquier nivel de resolución $k_0 \leq k \leq k_{max}$. El procedimiento para obtener los mapas de contornos de niveles de resolución inferior está descrito en la Sección 5.3.2

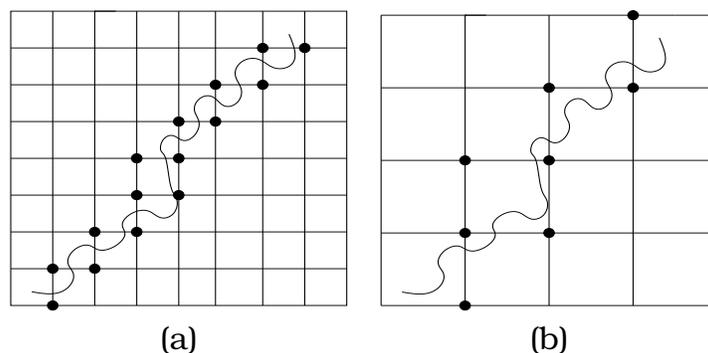


Figura 5.9: Ejemplo de mapa, la curva representa un contorno mientras que los puntos negros pertenecen al mapa. (a) mapa en el nivel k , (b) mapa en el nivel $k - 1$.

5.3.1

Construcción de $map^{k_{max}}$

El proceso para obtener los contornos de la imagen en el nivel de resolución máxima está descrito en la sección 4.7. Este proceso se divide en tres etapas:

- Detección de contornos: básicamente consiste en aplicar los filtros de Sobel y eliminar como píxeles pertenecientes susceptibles de pertenecer a los contornos aquellos que no superan un cierto umbral. Como resultado se obtienen dos matrices: una es una primera aproximación al mapa de los contornos y la segunda nos proporciona información acerca de la orientación de los perfiles.
- Reducción de la amplitud de las zonas detectadas: en esta etapa, empleando la matriz que contiene las orientaciones de los contornos, se reduce la amplitud de éstos.
- Eliminación de píxeles aislados y encadenado: en esta etapa rechazamos aquellos píxeles marcados que no están incluidos en un contorno de longitud mínima. Para encadenar los píxeles marcados empleamos los códigos de encadenado NESW (North-East-South-West) [33].

5.3.2

Mapas de contornos de distintos niveles

El siguiente procedimiento nos permite obtener el mapa del nivel de resolución k a partir del mapa del nivel de resolución inmediatamente superior, $k + 1$ (ver figura 5.9): para cada valor par de i , si $map^{k+1}(i, j) = 1$ y $map^{k+1}(i, j + 1) = 1$ entonces:

- Si j es par, entonces $map^k(\frac{i}{2}, \frac{j}{2}) = 1$ y $map^k(\frac{i}{2}, \frac{j}{2} + 1) = 1$.
- Si j es impar, entonces $map^k(\frac{i}{2}, \frac{j-1}{2}) = 1$ y $map^k(\frac{i}{2}, \frac{j+1}{2}) = 1$.

Intercambiando i y j , aplicamos posteriormente el mismo procedimiento para los valores pares de j si $map^{k+1}(i, j) = 1$ y $map^{k+1}(i + 1, j) = 1$.

Nota 5.2. *Notar que el procedimiento completo (construcción del mapa y la predicción mapa-dependiente introducida en la sección 5.2.2) define una multirresolución dependiente de los datos (i.e. no lineal) comparable con el planteamiento clásico ENO (Sección 1.2 o [10]) o el algoritmo más reciente ENO-EA (Capítulo 6 o [22]). No obstante, las principales diferencias entre este planteamiento y las técnicas citadas anteriormente son las siguientes:*

- *La predicción mapa-dependiente tiene en cuenta la orientación de los contornos detectados. Por tanto, la aproximación obtenida en los contornos no verticales y no horizontales es mejor que la que proporcionan los planteamientos bidireccionales (donde la predicción únicamente tiene en cuenta las direcciones vertical y horizontal). Ver, por ejemplo, el zoom de la figura 5.11.*
- *Si dejamos a un lado el problema de la definición del mapa, el procedimiento de multirresolución mapa-dependiente es lineal y por tanto estable respecto a las perturbaciones debidas al truncamiento después de la descomposición. Este no es el caso de los planteamientos basados en ENO, pues en éstos la detección y la predicción se realizan en cada nivel de resolución (ver nota 5.1)*

Sin embargo, el principal inconveniente de este procedimiento radica en el hecho de tener que guardar la información contenida en el mapa. Este punto, así como la codificación de la información comprimida es tratado en la sección siguiente.

5.4

Codificación de la información comprimida

La eficiencia de la codificación está relacionado con la naturaleza y estructura de los datos. Los datos que debemos comprimir son: el mapa, la imagen en baja resolución y los coeficientes correspondientes a los detalles tras ser truncados por un parámetro ϵ y posteriormente redondeados.

Haciendo uso de la tercera etapa de la construcción del mapa, este puede guardarse mediante dos archivos. El primero, contiene el principio de las cadenas y sus longitudes, mientras que el segundo contiene las cadenas NESW que representan la dirección que debemos seguir para ir de un píxel al siguiente.

Después, la imagen en baja resolución, los coeficientes de los detalles truncados y redondeados y el mapa son codificados haciendo uso el algoritmo *lossless* PPMZ [17] (prediction by partial matching PPM).

5.5

Aplicaciones y comparaciones numéricas

A continuación mostramos varios tests numéricos para comparar los siguientes algoritmos de compresión:

MDC : algoritmo de compresión mapa-dependiente, construido con un *stencil* uno dimensional de cuatro nodos y la correspon-

5. Multirresolución 2D no separable para valores puntuales 59

diente predicción multidireccional mapa-dependiente (ver Sección 5.2.2) asociada a la construcción del mapa basado en los filtros de Sobel (sección 5.3).

ENO4 : esquema ENO4 aplicado tensorialmente, con stenciles formados por cuatro nodos.

LIN4 : la compresión clásica aplicada tensorialmente, aplicada con stenciles centrados formados por cuatro nodos.

En todos los algoritmos anteriores se ha aplicado el algoritmo de codificación PPMZ (sección 5.4).

La comparación se realizó en términos de :

- Tasa de compresión: definida como $\text{bpp} = \frac{N}{512^2}$, donde N representa el número de bits necesarios para guardar la imagen comprimida y el denominador representa el número de píxeles de la imagen original.
- Calidad de la imagen reconstruida: medida en términos del PSNR (peak signal-to-noise ratio), definido como en la Sección 3.4. Notar que la calidad visual de la imagen reconstruida representa también un término importante a tener en cuenta.

5.5.1

Imagen geométrica

La tabla 5.2 y la figura 5.10, izquierda, muestran las tasas de compresión, bpp , y los correspondientes PSNR obtenidos empleando los tres algoritmos de compresión en el caso de una imagen geométrica. Se puede observar que el algoritmo MDC resulta muy eficiente para imágenes de este tipo ya que, para igual PSNR, la tasa de compresión resulta diez veces menor que la tasa alcanzada con el algoritmo LIN4. Si comparamos con la compresión ENO4, los efectos de inestabilidad asociados a la no linealidad de ENO4 hace que los resultados obtenidos con MDC sean mucho mejores.

La calidad del método MDC para una imagen geométrica, puede evaluarse también en términos de calidad visual de la imagen reconstruida (figura 5.11); podemos observar que los únicos lugares donde la reconstrucción no es óptima es en las esquinas.

	nnz		bytes		bpp	PSNR
	mapa	detalles	mapa	detalles		
LIN4	0	7191	0	7262	0,22	36,26
	0	5800	0	6162	0,19	34,41
	0	4971	0	5293	0,16	32,85
	0	3321	0	3542	0,11	29,22
ENO4	0	6968	0	7163	0,22	32,22
	0	6201	0	6593	0,20	31,64
	0	4668	0	5251	0,16	30,03
	0	2286	0	2350	0,07	25,62
MDC	6871	198	436	539	0,030	53,72
	6871	143	436	429	0,026	49,89
	6871	139	436	390	0,025	49,84
	6871	93	436	300	0,022	43,38

Tabla 5.2: Imagen geométrica: coeficientes no nulos y bytes del mapa y detalles, bits por pixel (*bpp*) y PSNR.

En caso de que la imagen geométrica sea perturbada con ruido, el método MDC vuelve a obtener mejores resultados que LIN4 y ENO4 (ver figura 5.10 centro). Además, el análisis de la calidad de la imagen reconstruida después de truncar puede servir para evaluar la capacidad del algoritmo para ampliar zonas cercanas a los contornos de la imagen. Los resultados de la figura 5.12 y la tabla 5.3 muestran que, en este contexto, MDC resulta más eficiente que los otros algoritmos.

5.5.2

Imagen real

La tabla 5.4 y figura 5.10 (derecha), muestran la tasa de compresión, *bpp*, y PSNR en el caso de aplicar los algoritmos a una imagen real.

En este caso el algoritmo mapa-dependiente no resulta ser tan

5. Multirresolución 2D no separable para valores puntuales 61

	nnz		bytes		bpp	PSNR
	mapa	detalles	mapa	detalles		
LIN4	0	11060	0	15501	0,47	35,86
	0	6116	0	10008	0,30	32,43
	0	3873	0	6203	0,19	29,54
	0	3234	0	5015	0,15	27,96
ENO4	0	7611	0	13963	0,43	33,27
	0	5687	0	10511	0,32	31,55
	0	3477	0	6795	0,21	28,15
	0	2104	0	4206	0,13	25,92
MDC	6892	951	598	2209	0,085	36,21
	6892	268	598	866	0,045	35,16
	6892	189	598	635	0,038	34,65
	6892	161	598	549	0,035	34,14

Tabla 5.3: *Image geométrica con ruido: coeficientes no nulos y bytes del mapa y detalles, bits por pixel (bpp) y PSNR.*

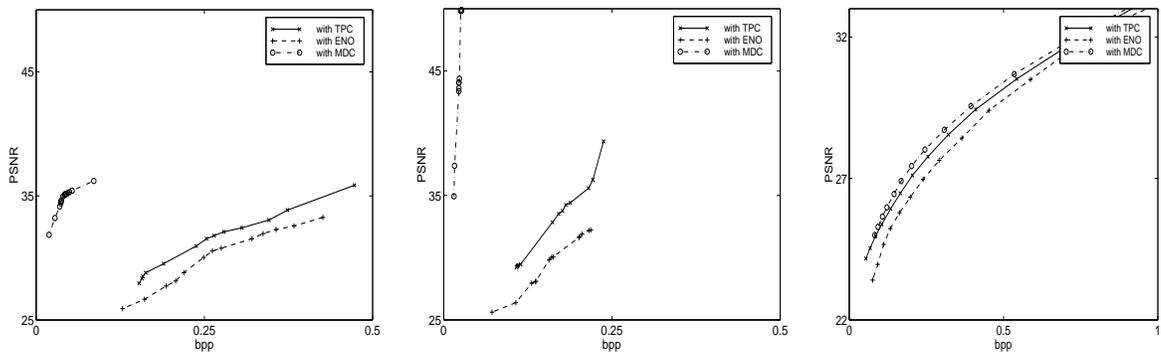


Figura 5.10: *PSNR vs. bpp. Izquierda: imagen geométrica. Centro: geométrica con ruido. Derecha: imagen real.*

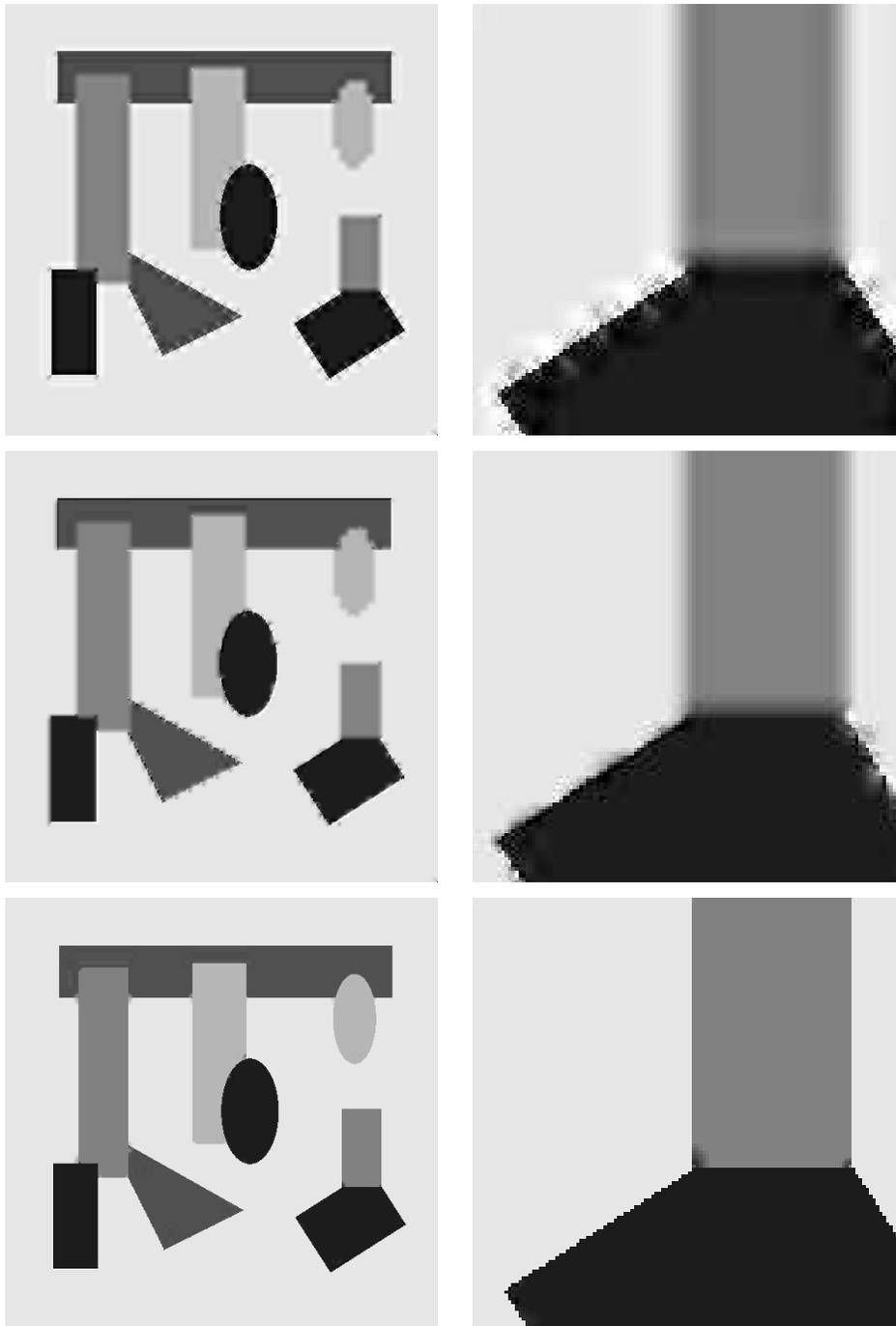


Figura 5.11: Reconstrucción de una imagen geométrica, $k_0 = 6$, $k_{max} = 9$. Izquierda: de arriba a abajo, LIN4, $bpp=0.11$, $PSNR=29.22$; ENO4 $bpp=.07$, $PSNR=25.62$; MDC, $bpp=0.02$ and $PSNR=43.38$. Derecha: detalle de las reconstrucciones.

5. Multirresolución 2D no separable para valores puntuales 63

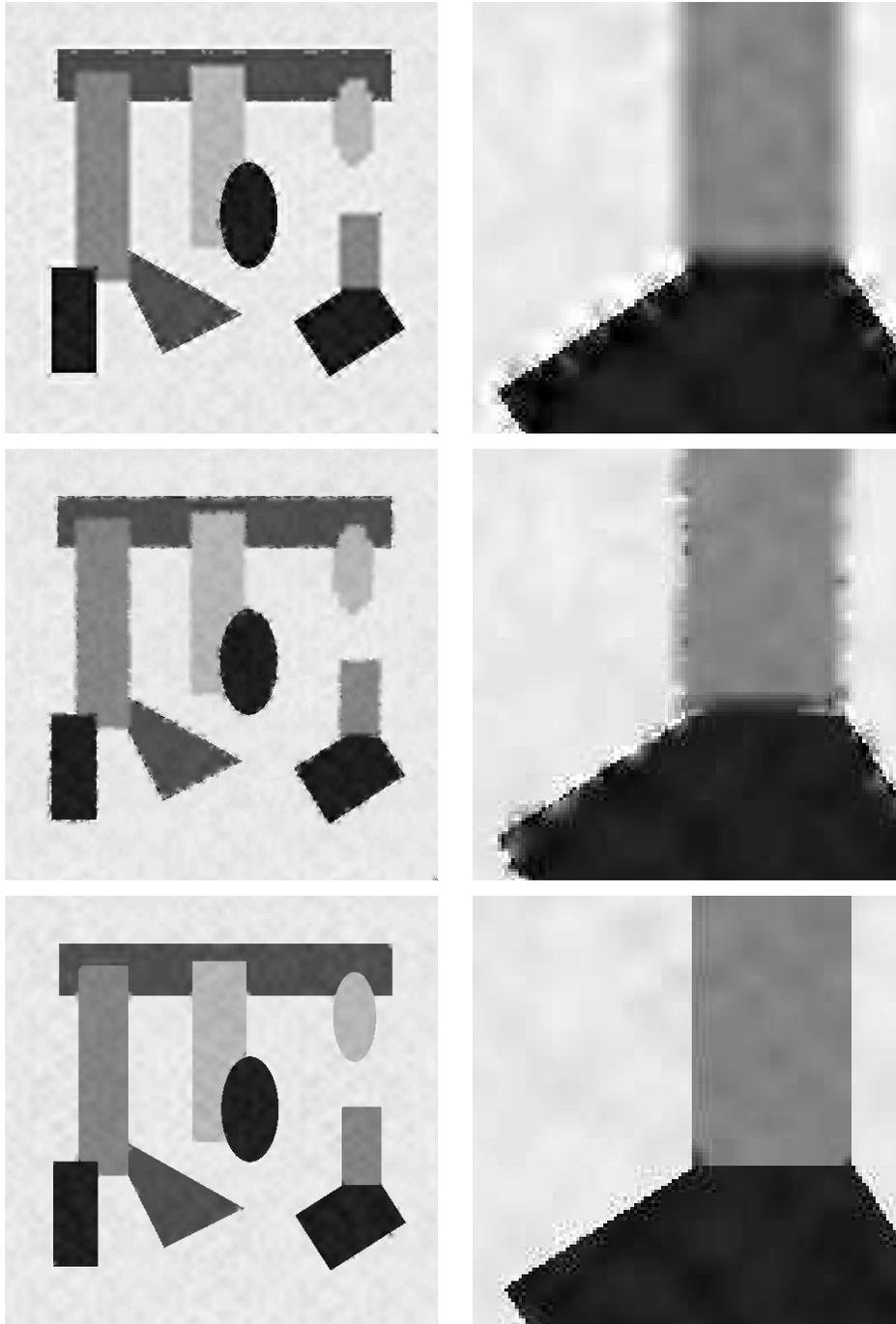


Figura 5.12: Reconstrucción de una imagen geométrica con ruido, $k_0 = 6$, $k_{max} = 9$. Izquierda: de arriba a abajo, LIN4, $bpp=0.15$, $PSNR=27.96$; ENO4, $bpp=.13$, $PSNR=25.92$; MDC, $bpp=0.03$, $PSNR=34.14$. Derecha: detalle de las reconstrucciones.

eficiente, para igual PSNR, la tasa de compresión, bpp resulta ser ligeramente menor que la obtenida con el algoritmo LIN4.

No obstante, se puede observar en la figura 5.13, que el esquema MDC proporciona una mejor reconstrucción de los contornos (y por tanto una mayor calidad visual de la imagen reconstruida) incluso si el PSNR resulta ser menor que el obtenido con LIN4. De igual modo, los resultados alcanzados con MDC resultan ser mejores que los obtenidos con ENO4.

	nnz		bytes		bpp	PSNR
	mapa	detalles	mapa	detalles		
LIN4	0	5133	0	10544	0,32	28,55
	0	3972	0	8384	0,25	27,77
	0	3099	0	6731	0,20	27,11
	0	2450	0	5456	0,17	26,48
	0	1948	0	4418	0,13	25,93
ENO4	0	5742	0	11970	0,36	28,43
	0	4463	0	9555	0,29	27,64
	0	3531	0	7840	0,24	26,97
	0	2450	0	6481	0,20	26,34
	0	2293	0	5337	0,16	25,79
MDC	6824	4149	1181	8941	0,31	28,72
	6824	3115	1181	6880	0,25	28,02
	6824	2386	1181	5467	0,20	27,44
	6824	1838	1181	4356	0,17	26,92
	6824	1461	1181	3603	0,15	26,45

Tabla 5.4: Imagen real: coeficientes no nulos y bytes del mapa y detalles, bits por pixel (bpp) y PSNR.

Nota 5.3. El codificador PPMZ no es el codificador óptimo para los datos transformados ya que no tiene en cuenta la estructura de árbol de los detalles. Dicha estructura si es aprovechada por los algoritmos EZW ([45]) o EBCOT ([46]). Con estos codificadores, tendríamos unas tasas de compresión superiores, no obstante las

5. Multirresolución 2D no separable para valores puntuales 65

conclusiones serían las mismas respecto a las comparaciones entre los algoritmos MDC, LIN4 y ENO4.

5.6

Conclusiones

En este capítulo, se ha presentado un algoritmo multiescala empleando la multirresolución de Harten. Su construcción implica una detección de contornos que nos proporciona un mapa de éstos y una predicción mapa-dependiente multidireccional. Las ventajas de este algoritmo son las siguientes:

- La transformada multiescala incorpora información sobre la geometría de la imagen.
- Dejando a un lado la detección de contornos, el proceso multiescala mapa-dependiente es completamente lineal. Por tanto, la convergencia del esquema de subdivisión asociado implica la estabilidad respecto a perturbaciones.
- El mapa y la descomposición multiescala pueden ser guardados de un modo eficiente.

Los resultados obtenidos son de dos tipos: por una parte, teóricamente, el algoritmo de compresión mapa-dependiente resulta ser óptimo para comprimir imágenes regulares a trozos. Por otra parte, las aplicaciones muestran que el algoritmo mapa-dependiente mejora los algoritmos standards como ENO y el clásico producto tensorial lineal, incluso en el caso de una imagen real.

Además de la mejora relacionada con el uso de codificadores que aprovechan la estructura de árbol de los coeficientes, se espera mejorar los resultados optimizando el proceso de detección en la primera etapa de construcción del mapa e incorporando una mayor dependencia del mapa en el proceso de predicción.



Figura 5.13: Reconstrucción de una imagen real, $k_0 = 6$, $k_{max} = 9$. Izquierda: de arriba a abajo, LIN4, $bpp=0.20$, $PSNR=27.11$; ENO4, bpp $PSNR=25.79$; con MDC, $bpp=0.15$, $PSNR=26.45$. Derecha: detalle de las reconstrucciones.

6

Multirresolución 2D no separable para medias en celda

6.1

Introducción

El uso de técnicas lineales en ampliación de imágenes (*zooming*) proporciona algoritmos rápidos y eficientes. No obstante los resultados obtenidos con estas técnicas presentan resultados insatisfactorios en los contornos de la imagen, produciendo perfiles difusos al no tener en cuenta la geometría de la imagen. Un modo de evitar la difusión de contornos es emplear técnicas interpolatorias no lineales que eviten emplear información de aquellas zonas que

contienen discontinuidades. Ésta es la idea de las técnicas interpolatorias no lineales ENO y ENO-Subcell Resolution, con las que se consigue reducir la difusión en los contornos ([12]). El principal inconveniente que presentan las técnicas interpolatorias ENO y ENO-SR es que son técnicas interpolatorias uno-dimensionales, de este modo su aplicación en la ampliación y compresión de imágenes debe realizarse de modo tensorial, produciendo un efecto escalera en aquellos perfiles que no son de naturaleza horizontal o vertical.

Con el objetivo de reducir el efecto escalera producido al aplicar tensorialmente técnicas interpolatorias no lineales introducimos un operador predicción dos-dimensional que utiliza la geometría de la imagen. Para ello detecta los contornos de la imagen a partir de la información proporcionada por el uso conjunto de las diferencias divididas de los datos y los filtros de Sobel (ver apartado 4.6), y posteriormente en aquellas celdas marcadas por la detección de contornos se aplica un operador predicción dos-dimensional en el que calcularemos las medias en las celdas que contienen los perfiles bajo dos supuestos: un primer caso, suponiendo que la imagen es constante a ambos lados del contorno, y un segundo caso, bajo el supuesto de que a ambos lados del perfil la imagen toma la forma de dos polinomios $p_1(x, y)$ y $p_2(x, y)$. En [22, 39, 6]) podemos encontrar algunos resultados en la misma línea que los presentados en este capítulo.

La estructura del capítulo es la siguiente: la sección 6.2 describe como calculamos las aproximaciones en aquellas celdas afectadas por los contornos en los dos supuestos mencionados anteriormente. En la sección 6.3 explicamos brevemente la detección de contornos empleada (ver apartado 4.6). Finalmente, en la sección 6.4 se explica el algoritmo de interpolación adaptada. Los resultados numéricos se muestran en la sección 6.5, en ellos comparamos los resultados obtenidos aplicando diferentes técnicas interpolatorias, tanto en la interpolación como compresión de imágenes.

6.2

Predicción en contornos

De ahora en adelante consideraremos que nuestros datos son el resultado de aplicar un operador discretización a una función $f(x, y)$ definida sobre $[0, 1] \times [0, 1]$.

Supongamos que los datos que tenemos, $\{\bar{f}_{i,j}^k\}_{i,j=1}^{2^k}$, representan las medias de $f(x, y)$ en la discretización $\{C_{i,j}^k\}_{i,j=1}^{N_k}$, donde k indica el nivel de resolución, $C_{i,j}^k = [x_{i-1}^k, x_i^k] \times [y_{j-1}^k, y_j^k]$, y $x_n^k = y_n^k = n \cdot \frac{1}{2^k}$, $n = 0, \dots, 2^k$, i.e.,

$$\bar{f}_{i,j}^k = \frac{1}{|C_{i,j}^k|} \int_{C_{i,j}^k} f(x, y) \, dx dy, \quad i, j = 1, \dots, 2^k. \quad (6.1)$$

El algoritmo de detección de contornos que emplearemos (sección 6.3), clasifica los contornos en cuatro tipos: horizontales, verticales, diagonales positivos y diagonales negativos. Esto nos permite actuar de un modo más adecuado en cada caso. Así supongamos que $C_{i,j}^k$ es una celda que ha sido seleccionada por el algoritmo de detección de contornos como perteneciente a un contorno horizontal. Para aplicar nuestra predicción, necesitamos obtener una aproximación local al contorno. Para ello supondremos en primer lugar que el contorno es una línea recta y, en segundo lugar que a ambos lados del contorno la imagen es constante, tomando ésta los valores a y b a ambos lados del perfil. Bajo estos supuestos podemos calcular la recta que aproxima al contorno en la celda $C_{i,j}^k$. Para ello calculamos los puntos y'_{j-1} , y'_j , y'_{j+1} , siendo éstos las intersecciones del contorno con las rectas $x = x_{i-3/2}^k$, $x = x_{i-1/2}^k$, $x = x_{i+1/2}^k$ respectivamente (figura 6.1). Notar que y'_{j-1} , y'_j , y'_{j+1} verifican la siguiente relación,

$$a(y_{j+1}^k - y'_{j+h}) + b(y'_{j+h} - y_{j-2}^k) = \bar{f}_{i+h,j+1}^k + \bar{f}_{i+h,j}^k + \bar{f}_{i+h,j-1}^k, \quad h = -1, 0, 1, \quad (6.2)$$

y por tanto,

$$y'_{j+h} = \frac{(\bar{f}_{i+h,j+1}^k + \bar{f}_{i+h,j}^k + \bar{f}_{i+h,j-1}^k) - ay_{j+1}^k + by_{j-2}^k}{b - a}, \quad h = -1, 0, 1.$$

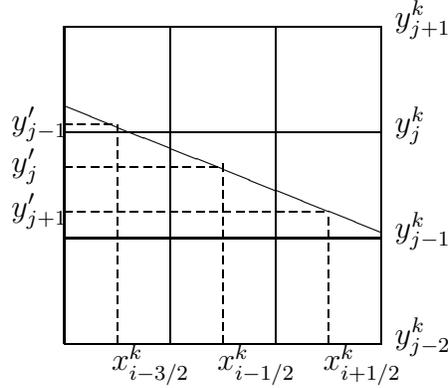


Figura 6.1: Contorno horizontal.

Calculando la recta de regresión $\{px + qy = c\}$ asociada a los puntos $(x_{i-3/2}^k, y_{j-1}^k)$, $(x_{i-1/2}^k, y_j^k)$ y $(x_{i+1/2}^k, y_{j+1}^k)$, tendremos una aproximación al perfil en la celda $C_{i,j}^k$. A efectos prácticos, consideraremos:

$$a = \bar{f}_{i+h, j+2}^k \text{ y } b = \bar{f}_{i+h, j-2}^k \quad h = -1, 0, 1.$$

Para hallar la aproximación al perfil de la imagen en celdas marcadas como integrantes de un contorno vertical, procederemos análogamente con la salvedad de que en este caso nuestra aproximación estará constituida ahora por la recta de regresión asociada a los puntos $(x'_{i-1}, y_{j+3/2}^k)$, $(x'_i, y_{j+1/2}^k)$ y $(x'_{i+1}, y_{j-1/2}^k)$, donde los puntos x'_{i-1} , x'_i y x'_{i+1} no son más que las intersecciones del perfil con las rectas $y = y_{j-3/2}^k, y_{j-1/2}^k, y_{j+1/2}^k$ respectivamente, cuyas expresiones, suponiendo que el perfil es una línea recta y tomando los valores de a y b a la izquierda y derecha de éste respectivamente, vendrán dadas por:

$$x'_{i+h} = \frac{\bar{f}_{i-1, j-h}^k + \bar{f}_{i, j-h}^k + \bar{f}_{i+1, j-h}^k + ax_{i-2}^k - bx_{i+1}^k}{a - b}, \quad h = -1, 0, 1. \quad (6.3)$$

Para calcular los puntos anteriores, tomaremos como a y b los valores siguientes:

$$a = \bar{f}_{i-2, j-k}^k \text{ y } b = \bar{f}_{i+2, j-k}^k \quad k = -1, 0, 1.$$

6. Multirresolución 2D no separable para medias en celda 171

La aproximación para las celdas marcadas como pertenecientes a contornos diagonales la calcularemos del mismo modo que para las celdas de contornos horizontales.

Una vez hemos calculado una aproximación al perfil de la imagen en $C_{i,j}^k$, estamos en condiciones de obtener una aproximación a las medias de $f(x, y)$ en las celdas $C_{2i-1,2j-1}^{k+1}$, $C_{2i-1,2j}^{k+1}$, $C_{2i,2j-1}^{k+1}$ y $C_{2i,2j}^{k+1}$. Para ello proponemos dos posibilidades:

Caso 1 : En este caso suponemos que la imagen es constante a ambos lados del contorno.

Caso 2 : En este caso se supone que la imagen a ambos del contorno coincide con un polinomio del tipo siguiente:

$$\sum_{0 \leq l, m \leq 2} a_{l,m} x^l y^m. \quad (6.4)$$

6.2.1

Caso 1: Imagen constante a trozos

Supongamos que la celda $C_{i,j}^k$ está marcada por contener un perfil, si además suponemos que $f(x, y)$ toma los valores a y b a ambos lados del contorno, con $a = \bar{f}_{i,j+2}^k$, $b = \bar{f}_{i,j-2}^k$ si la celda ha sido identificada como integrante de un perfil horizontal o diagonal ($a = \bar{f}_{i-2,j}^k$, $b = \bar{f}_{i+2,j}^k$ si la celda pertenece a un perfil vertical). Entonces la expresión de $f(x, y)$ en la celda $C_{i,j}^k$ es la siguiente:

$$f_{C_{i,j}^k}(x, y) = a\chi_{\{px+qy \geq c\}} + b\chi_{\{px+qy < c\}}.$$

De este modo la aproximación a $\bar{f}_{2i-1,2j-1}^{k+1}$, $\bar{f}_{2i-1,2j}^{k+1}$, $\bar{f}_{2i,2j-1}^{k+1}$ y $\bar{f}_{2i,2j}^{k+1}$ puede ser calculada evaluando las siguientes integrales:

$$\hat{f}_{m,n}^{k+1} = \frac{1}{h_{k+1}^2} \int_{C_{m,n}^{k+1}} f_{C_{i,j}^k}(x, y) dx dy, \quad m = 2i - 1, 2i, n = 2j - 1, 2j. \quad (6.5)$$

Si la aproximación al contorno en la celda $C_{i,j}^k$, $\{px + qy = c\}$, verifica $p, q \neq 0$ podemos escribir la recta como $y = mx + n$, siendo $m = -\frac{p}{q}$ y $n = \frac{c}{q}$, de este modo podemos calcular la media de $f(x, y)$ en la celda $C = [x_0, x_1] \times [y_0, y_1]$, con $h^2 = (x_1 - x_0)(y_1 - y_0)$, a partir de las siguientes relaciones:

a) Si $mx_0 + n < y_0$ y $mx_1 + n < y_0$:

$$\frac{1}{h^2} \int_C f(x, y) dx dy = a.$$

b) Si $mx_0 + n < y_0$ y $y_0 \leq mx_1 + n \leq y_1$: llamamos $x_l = \frac{y_0 - n}{m}$ y $y_r = mx_1 + n$, obteniendo:

$$\frac{1}{h^2} \int_C f(x, y) dx dy = \frac{1}{h^2} \left(\frac{1}{2} (a-b) (y_r - y_0) (x_1 - x_r) + b (x_1 - x_0) (y_1 - y_0) \right)$$

c) Si $mx_0 + n < y_0$ y $mx_1 + n > y_1$: llamamos $x_l = \min\left\{\frac{y_0 - n}{m}, \frac{y_1 - n}{m}\right\}$ y $x_r = \max\left\{\frac{y_0 - n}{m}, \frac{y_1 - n}{m}\right\}$,

$$\frac{1}{h^2} \int_C f(x, y) dx dy = \frac{1}{h^2} \left(\frac{1}{2} (y_1 - y_0) (a(x_l - x_0)(x_r - x_0) + b(x_1 - x_r)(x_1 - x_l)) \right).$$

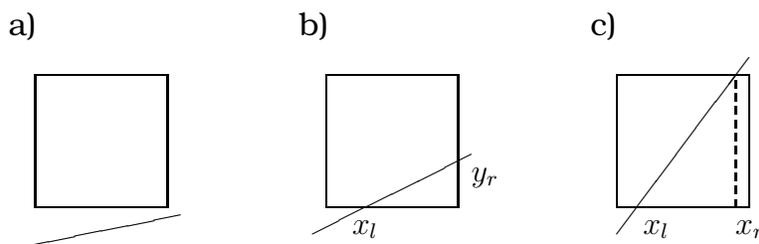


Figura 6.2: $mx_0 + n < y_0$

d) Si $y_0 \leq mx_0 + n \leq y_1$ y $mx_1 + n < y_0$: llamamos $y_l = mx_0 + n$ y $x_l = \frac{y_0 - n}{m}$, donde:

$$\frac{1}{h^2} \int_C f(x, y) dx dy = \frac{1}{h^2} \left(\frac{1}{2} (x_l - x_0) (y_l - y_0) (b - a) + a (x_1 - x_0) (y_1 - y_0) \right).$$

e) Si $y_0 \leq mx_0 + n \leq y_1$ y $y_0 \leq mx_1 + n \leq y_1$: llamamos $y_l = mx_0 + n$ y $y_r = mx_1 + n$, así:

$$\frac{1}{h^2} \int_C f(x, y) dx dy = \frac{1}{h^2} \left(\frac{1}{2} (a(x_1 - x_0) ((y_1 - y_l)(y_1 - y_r) + b(y_l - y_0)(y_r - y_0)) \right).$$

6. Multirresolución 2D no separable para medias en celda 173

f) Si $y_0 \leq mx_0 + n \leq y_1$ y $mx_1 + n > y_1$., llamamos $y_l = mx_0 + n$ y $x_r = \frac{y_1 - n}{m}$, por tanto:

$$\frac{1}{h^2} \int_C f(x, y) dx dy = \frac{1}{h^2} \left(\frac{1}{2} (x_r - x_0) ((y_1 - y_l)(a - b) + b(y_1 - y_0)(x_1 - x_0)) \right).$$

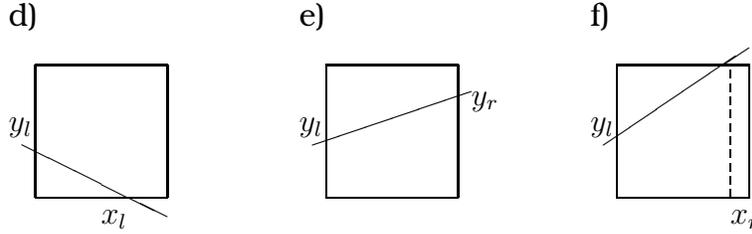


Figura 6.3: $y_0 \leq mx_0 + n \leq y_1$

g) Si $mx_0 + n > y_1$ y $mx_1 + n < y_0$., sea $x_l = \frac{y_1 - n}{m}$ y $x_r = \frac{y_0 - n}{m}$, así la media buscada es:

$$\frac{1}{h^2} \int_C f(x, y) dx dy = \frac{1}{h^2} \left(\frac{1}{2} (y_1 - y_0) (b(x_r - x_0)(x_l - x_0) + a(x_1 - x_l)(x_1 - x_r)) \right).$$

h) Si $mx_0 + n > y_1$ y $y_0 \leq mx_1 + n \leq y_1$., sea $x_l = \frac{y_1 - n}{m}$ y $y_r = mx_1 + n$, obteniendo:

$$\frac{1}{h^2} \int_C f(x, y) dx dy = \frac{1}{h^2} \left(\frac{1}{2} (y_1 - y_r) (x_1 - x_l) (b - a) + a(x_1 - x_0) (y_1 - y_0) \right).$$

i) Si $mx_0 + n > y_1$ y $mx_1 + n > y_1$., tenemos:

$$\frac{1}{h^2} \int_C f(x, y) dx dy = b.$$

Notar que en el caso $p = 0$, nuestra aproximación al perfil es $y = c/q = k_1$, de modo que si $k_1 \geq y_1$, nuestra aproximación para la media en la celda C será b , si $k_1 \leq y_0$ la media en esta celda será a y, si $y_0 \leq k_1 \leq y_1$ la media será $\frac{1}{h^2} (a(y_1 - k_1) + b(k_1 - y_0))(x_1 - x_0)$. Del mismo modo, si $q = 0$, nuestra aproximación al contorno de la imagen es $x = c/p = k_2$, donde si $k_2 < x_0$ nuestra aproximación vendrá dada por b , si $k_2 > x_1$ nuestra aproximación tomará el valor de a y, si $x_0 \leq k_2 \leq x_1$ la aproximación vendrá dada por $\frac{1}{h_2} (y_1 - y_0) (a(k - x_0) + b(x_1 - k))$.

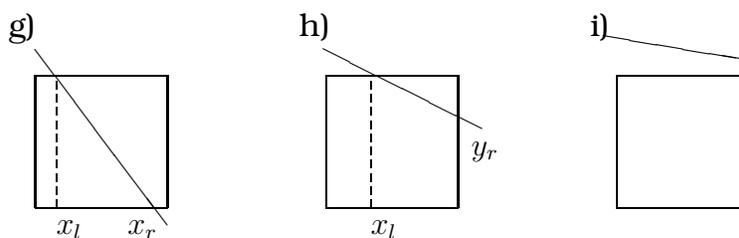


Figura 6.4: $mx_0 + n > y_1$

6.2.2

Caso 2: Imagen polinómica a trozos

En este caso, suponemos que $f(x, y)$ posee la siguiente expresión:

$$f|_{C_{i,j}^0}(x, y) = p_1(x, y)\chi_{\{px+qy \geq c\}} + p_2(x, y)\chi_{\{px+qy < c\}},$$

siendo p_1 y p_2 las siguientes funciones

$$p_1(x, y) = \sum_{0 \leq l, m \leq 2} a_{l,m} x^l y^m, \quad p_2(x, y) = \sum_{0 \leq l, m \leq 2} b_{l,m} x^l y^m,$$

donde $p_1(x)$ y $p_2(x)$ son polinomios cuya forma viene dada por la expresión (6.4). La obtención de éstos polinomios la explicaremos posteriormente.

De este modo, procediendo como en el supuesto anterior calcularemos la aproximación a la media de $f(x, y)$ en las celdas $C_{2i-1, 2j-1}^{k+1}$, $C_{2i-1, 2j}^{k+1}$, $C_{2i, 2j-1}^{k+1}$ y $C_{2i, 2j}^{k+1}$ como

$$\hat{f}_{m,n}^{k+1} = \frac{1}{h_{k+1}^2} \int_{C_{m,n}^{k+1}} f|_{C_{i,j}^k}(x, y) dx dy, \quad m = 2i - 1, 2i, \quad n = 2j - 1, 2j.$$

Estas integrales son calculadas a partir de las relaciones (6.6), (6.7) y (6.8):

6. Multirresolución 2D no separable para medias en celda 175

$$\begin{aligned}
\int_{x_0}^{x_1} \int_{y_0}^{y_1} p(x, y) dy dx &= (-x_0 y_1 + x_0 y_0 - x_1 y_0 + x_1 y_1) c_{0,0} \\
&+ (-1/2 x_1^2 y_0 - 1/2 x_0^2 y_1 + 1/2 x_1^2 y_1 + 1/2 x_0^2 y_0) c_{1,0} \\
&+ (-1/2 y_0^2 x_1 + 1/2 y_1^2 x_1 + 1/2 y_0^2 x_0 - 1/2 y_1^2 x_0) c_{0,1} \\
&+ (1/4 x_1^2 y_1^2 - 1/4 x_1^2 y_0^2 + 1/4 x_0^2 y_0^2 - 1/4 x_0^2 y_1^2) c_{1,1} \\
&+ (-1/6 x_1^3 y_0^2 + 1/6 x_1^3 y_1^2 + 1/6 x_0^3 y_0^2 - 1/6 x_0^3 y_1^2) c_{2,1} \\
&+ (-1/6 x_1^2 y_0^3 + 1/6 x_0^2 y_0^3 + 1/6 x_1^2 y_1^3 - 1/6 x_0^2 y_1^3) c_{1,2} \\
&+ (-1/3 y_1^3 x_0 - 1/3 y_0^3 x_1 + 1/3 y_0^3 x_0 + 1/3 y_1^3 x_1) c_{0,2} \\
&+ (-1/3 x_1^3 y_0 + 1/3 x_1^3 y_1 - 1/3 x_0^3 y_1 + 1/3 x_0^3 y_0) c_{2,0} \\
&+ (1/9 x_0^3 y_0^3 + 1/9 x_1^3 y_1^3 - 1/9 x_1^3 y_0^3 - 1/9 x_0^3 y_1^3) c_{2,2},
\end{aligned} \tag{6.6}$$

$$\begin{aligned}
\int_{x_0}^{x_1} \int_{y_0}^{mx+n} p(x, y) dy dx &= (-y_0 x_1 + 1/2 m x_1^2 - n x_0 + y_0 x_0 + n x_1 - 1/2 m x_0^2) c_{0,0} \\
&+ (1/3 x_1^3 m - 1/2 x_0^2 n + 1/2 x_1^2 n \\
&+ 1/2 x_0^2 y_0 - 1/2 x_1^2 y_0 - 1/3 x_0^3 m) c_{1,0} \\
&+ (-1/6 m^2 x_0^3 + 1/6 m^2 x_1^3 - 1/2 n^2 x_0 + 1/2 n^2 x_1 \\
&+ 1/2 y_0^2 x_0 - 1/2 m x_0^2 n + 1/2 m x_1^2 n - 1/2 y_0^2 x_1) c_{0,1} \\
&+ (-1/3 x_0^3 m n - 1/4 x_1^2 y_0^2 + 1/4 x_1^2 n^2 + 1/3 x_1^3 m n \\
&- 1/4 x_0^2 n^2 + 1/4 x_0^2 y_0^2 - 1/8 x_0^4 m^2 + 1/8 x_1^4 m^2) c_{1,1} \\
&+ (-1/6 x_1^3 y_0^2 + 1/10 x_1^5 m^2 - 1/6 x_0^3 n^2 + 1/6 x_1^3 n^2 \\
&- 1/4 x_0^4 m n + 1/4 x_1^4 m n - 1/10 x_0^5 m^2 + 1/6 x_0^3 y_0^2) c_{2,1} \\
&+ (1/6 x_0^2 y_0^3 + 1/15 x_1^5 m^3 - 1/6 x_0^2 n^3 + 1/6 x_1^2 n^3 \\
&+ 1/3 x_1^3 m n^2 - 1/15 x_0^5 m^3 - 1/3 x_0^3 m n^2 \\
&- 1/4 x_0^4 m^2 n - 1/6 x_1^2 y_0^3 + 1/4 x_1^4 m^2 n) c_{1,2} \\
&+ (1/3 x_0^3 y_0 + 1/3 x_1^3 n - 1/3 x_1^3 y_0 \\
&- 1/4 x_0^4 m + 1/4 x_1^4 m - 1/3 x_0^3 n) c_{2,0} \\
&+ (1/2 m x_1^2 n^2 + 1/3 n^3 x_1 - 1/12 m^3 x_0^4 - 1/2 m x_0^2 n^2 \\
&+ 1/12 m^3 x_1^4 - 1/3 m^2 x_0^3 n - 1/3 y_0^3 x_1 \\
&+ 1/3 y_0^3 x_0 - 1/3 n^3 x_0 + 1/3 m^2 x_1^3 n) c_{0,2} \\
&+ (-1/9 x_1^3 y_0^3 + 1/9 x_1^3 n^3 - 1/9 x_0^3 n^3 + 1/5 x_1^5 m^2 n \\
&- 1/5 x_0^5 m^2 n + 1/18 x_1^6 m^3 + 1/9 x_0^3 y_0^3 \\
&- 1/4 x_0^4 m n^2 - 1/18 x_0^6 m^3 + 1/4 x_1^4 m n^2) c_{2,2},
\end{aligned} \tag{6.7}$$

$$\begin{aligned}
\int_{x_0}^{x_1} \int_{mx+n}^{y_1} p(x, y) dy dx &= (1/2 mx_0^2 + y_1 x_1 - 1/2 mx_1^2 - nx_1 - y_1 x_0 + nx_0) c_{0,0} \\
&+ (1/2 x_1^2 y_1 - 1/3 x_1^3 m + 1/3 x_0^3 m \\
&+ 1/2 x_0^2 n - 1/2 x_0^2 y_1 - 1/2 x_1^2 n) c_{1,0} \\
&+ (-1/2 n^2 x_1 + 1/2 n^2 x_0 + 1/6 m^2 x_0^3 + 1/2 y_1^2 x_1 \\
&+ 1/2 mx_0^2 n - 1/6 m^2 x_1^3 - 1/2 mx_1^2 n - 1/2 y_1^2 x_0) c_{0,1} \\
&+ (-1/4 x_0^2 y_1^2 - 1/3 x_1^3 mn + 1/8 x_0^4 m^2 - 1/8 x_1^4 m^2 \\
&+ 1/4 x_1^2 y_1^2 + 1/4 x_0^2 n^2 - 1/4 x_1^2 n^2 + 1/3 x_0^3 mn) c_{1,1} \\
&+ (-1/10 x_1^5 m^2 - 1/6 x_1^3 n^2 + 1/10 x_0^5 m^2 + 1/6 x_0^3 n^2 \\
&+ 1/6 x_1^3 y_1^2 + 1/4 x_0^4 mn - 1/4 x_1^4 mn - 1/6 x_0^3 y_1^2) c_{2,1} \\
&+ (-1/3 x_1^3 mn^2 + 1/3 x_0^3 mn^2 - 1/6 x_1^2 n^3 + 1/15 x_0^5 m^3 \\
&- 1/6 x_0^2 y_1^3 + 1/4 x_0^4 m^2 n - 1/15 x_1^5 m^3 \\
&+ 1/6 x_0^2 n^3 - 1/4 x_1^4 m^2 n + 1/6 x_1^2 y_1^3) c_{1,2} \\
&+ (-1/3 x_1^3 n + 1/3 x_0^3 n + 1/4 x_0^4 m \\
&- 1/4 x_1^4 m + 1/3 x_1^3 y_1 - 1/3 x_0^3 y_1) c_{2,0} \\
&+ (1/12 m^3 x_0^4 - 1/12 m^3 x_1^4 - 1/3 m^2 x_1^3 n + 1/3 n^3 x_0 \\
&- 1/3 y_1^3 x_0 + 1/3 y_1^3 x_1 - 1/2 mx_1^2 n^2 \\
&+ 1/3 m^2 x_0^3 n + 1/2 mx_0^2 n^2 - 1/3 n^3 x_1) c_{0,2} \\
&+ (-1/5 x_1^5 m^2 n - 1/9 x_0^3 y_1^3 + 1/18 x_0^6 m^3 - 1/18 x_1^6 m^3 \\
&- 1/9 x_1^3 n^3 + 1/4 x_0^4 mn^2 - 1/4 x_1^4 mn^2 \\
&+ 1/9 x_1^3 y_1^3 + 1/9 x_0^3 n^3 + 1/5 x_0^5 m^2 n) c_{2,2},
\end{aligned} \tag{6.8}$$

donde $p(x, y) = \sum_{0 \leq l, m \leq 2} c_{l,m} x^l y^m$. Así si procedemos de igual modo que en el caso anterior, sea $C = [x_0, x_1] \times [y_0, y_1]$, y sea la recta $y = mx + n$ nuestra aproximación al perfil de la imagen (supondremos $p, q \neq 0$), $(x_1 - x_0)(y_1 - y_0) = h^2$, además supongamos que encima de dicho contorno la función $f(x, y)$ toma el valor de $p_1(x, y)$ y debajo de éste el valor de $p_2(x, y)$, entonces la media de f en las celdas dadas por las figuras 6.2, 6.3 y 6.4 las podemos obtener empleando las siguientes relaciones:

a) Si $mx_0 + n < y_0$ y $mx_1 + n < y_0$:

$$\frac{1}{h^2} \int_C f(x, y) dx dy = \frac{1}{h^2} \int_{x_0}^{x_1} \int_{y_0}^{y_1} p_1(x, y) dy dx.$$

6. Multirresolución 2D no separable para medias en celda 177

b) Si $mx_0 + n < y_0$ y $y_0 \leq mx_1 + n \leq y_1$: llamamos $x_l = \frac{y_0 - n}{m}$ y $y_r = mx_1 + n$, obteniendo:

$$\begin{aligned} \frac{1}{h^2} \int_C f(x, y) dx dy &= \frac{1}{h^2} \int_{x_0}^{x_l} \int_{y_0}^{y_1} p_1(x, y) dx dy + \int_{x_l}^{x_1} \int_{mx+n}^{y_1} p_1(x, y) dx dy \\ &+ \int_{x_l}^{x_1} \int_{y_0}^{mx+n} p_2(x, y) dx dy. \end{aligned}$$

c) Si $mx_0 + n < y_0$ y $mx_1 + n > y_1$: llamamos $x_l = \min\{\frac{y_0 - n}{m}, \frac{y_1 - n}{m}\}$ y $x_r = \max\{\frac{y_0 - n}{m}, \frac{y_1 - n}{m}\}$,

c.i) $m \geq 0$:

$$\begin{aligned} \frac{1}{h^2} \int_C f(x, y) dx dy &= \frac{1}{h^2} \left(\int_{x_0}^{x_l} \int_{y_0}^{y_1} p_1(x, y) dy dx + \int_{x_l}^{x_r} \int_{mx+n}^{y_1} p_1(x, y) dy dx \right. \\ &+ \left. \int_{x_l}^{x_r} \int_{y_0}^{mx+n} p_2(x, y) dy dx + \int_{x_r}^{x_1} \int_{y_0}^{y_1} p_2(x, y) dy dx \right). \end{aligned}$$

c.ii) $m < 0$:

$$\begin{aligned} \frac{1}{h^2} \int_C f(x, y) dx dy &= \frac{1}{h^2} \left(\int_{x_0}^{x_l} \int_{y_0}^{y_1} p_2(x, y) dy dx + \int_{x_l}^{x_r} \int_{y_0}^{mx+n} p_2(x, y) dy dx \right. \\ &+ \left. \int_{x_l}^{x_r} \int_{mx+n}^{y_1} p_1(x, y) dy dx + \int_{x_r}^{x_1} \int_{y_0}^{y_1} p_1(x, y) dy dx \right). \end{aligned}$$

d) Si $y_0 \leq mx_0 + n \leq y_1$ y $mx_1 + n < y_0$: llamamos $y_l = mx_0 + n$ y $x_l = \frac{y_0 - n}{m}$, donde:

$$\begin{aligned} \frac{1}{h^2} \int_C f(x, y) dx dy &= \frac{1}{h^2} \left(\int_{x_0}^{x_l} \int_{y_0}^{mx+n} p_2(x, y) dy dx + \int_{x_0}^{x_l} \int_{mx+n}^{y_1} p_1(x, y) dy dx \right. \\ &+ \left. \int_{x_l}^{x_1} \int_{y_0}^{y_1} p_1(x, y) dy dx \right). \end{aligned}$$

e) Si $y_0 \leq mx_0 + n \leq y_1$ y $y_0 \leq mx_1 + n \leq y_1$: llamamos $y_l = mx_0 + n$ y $y_r = mx_1 + n$, así:

$$\frac{1}{h^2} \int_C f(x, y) dx dy = \frac{1}{h^2} \left(\int_{x_0}^{x_1} \int_{mx+n}^{y_1} p_1(x, y) dy dx + \int_{x_0}^{x_1} \int_{y_0}^{mx+n} p_2(x, y) dy dx \right).$$

f) Si $y_0 \leq mx_0 + n \leq y_1$ y $mx_1 + n > y_1$:, llamamos $y_l = mx_0 + n$ y $x_r = \frac{y_1 - n}{m}$, por tanto:

$$\begin{aligned} \frac{1}{h^2} \int_C f(x, y) dx dy &= \frac{1}{h^2} \left(\int_{x_0}^{x_r} \int_{mx+n}^{y_1} p_1(x, y) dy dx + \frac{1}{h^2} \left(\int_{x_0}^{x_r} \int_{y_0}^{mx+n} p_2(x, y) dy dx \right. \right. \\ &\quad \left. \left. + \int_{x_r}^{x_1} \int_{y_0}^{y_1} p_2(x, y) dy dx \right) \right). \end{aligned}$$

g) Si $mx_0 + n > y_1$ y $mx_1 + n < y_0$:, sea $x_l = \min \left\{ \frac{y_0 - n}{m}, \frac{y_1 - n}{m} \right\}$ y $x_r = \max \left\{ \frac{y_0 - n}{m}, \frac{y_1 - n}{m} \right\}$, así la media buscada es:

g.i) $m \geq 0$:

$$\begin{aligned} \frac{1}{h^2} \int_C f(x, y) dx dy &= \frac{1}{h^2} \left(\int_{x_0}^{x_l} \int_{y_0}^{y_1} p_1(x, y) dy dx + \int_{x_l}^{x_r} \int_{mx+n}^{y_1} p_1(x, y) dy dx \right. \\ &\quad \left. + \int_{x_l}^{x_r} \int_{y_0}^{mx+n} p_2(x, y) dy dx + \int_{x_r}^{x_1} \int_{y_0}^{y_1} p_2(x, y) dy dx \right). \end{aligned}$$

g.ii) $m < 0$:

$$\begin{aligned} \frac{1}{h^2} \int_C f(x, y) dx dy &= \frac{1}{h^2} \left(\int_{x_0}^{x_l} \int_{y_0}^{y_1} p_2(x, y) dy dx + \int_{x_l}^{x_r} \int_{y_0}^{mx+n} p_2(x, y) dy dx \right. \\ &\quad \left. + \int_{x_l}^{x_r} \int_{mx+n}^{y_1} p_1(x, y) dy dx + \int_{x_r}^{x_1} \int_{y_0}^{y_1} p_1(x, y) dy dx \right). \end{aligned}$$

h) Si $mx_0 + n > y_1$ y $y_0 \leq mx_1 + n \leq y_1$:, sea $x_l = \frac{y_1 - n}{m}$ y $y_r = mx_1 + n$, obteniendo:

$$\begin{aligned} \frac{1}{h^2} \int_C f(x, y) dx dy &= \frac{1}{h^2} \left(\int_{x_0}^{x_l} \int_{y_0}^{y_1} p_2(x, y) dy dx + \int_{x_l}^{x_1} \int_{y_0}^{mx+n} p_2(x, y) dy dx \right. \\ &\quad \left. + \int_{x_l}^{x_1} \int_{mx+n}^{y_1} p_1(x, y) dy dx \right). \end{aligned}$$

i) Si $mx_0 + n > y_1$ y $mx_1 + n > y_1$:, tenemos:

$$\frac{1}{h^2} \int_C f(x, y) dx dy = \frac{1}{h^2} \int_{x_0}^{x_1} \int_{y_0}^{y_1} p_2(x, y) dy dx.$$

6. Multirresolución 2D no separable para medias en celda 179

En los casos en el que $\{px + qy = c\}$ tengamos $p = 0$ o $q = 0$, podemos calcular la media de $f(x, y)$ sencillamente procediendo como en los casos anteriores.

Los polinomios $p_1(x)$ y $p_2(x)$ los construiremos de modo que interpolen las medias de la función $f(x, y)$ en unos determinados *stencils* 3×3 . Los *stencils* que consideraremos, dependiendo de si la celda ha sido marcada como integrante de un perfil horizontal, vertical, diagonal positivo o negativo, podemos verlos en las figuras 6.5, 6.6, 6.7 y 6.8, respectivamente.

De este modo tenemos tres posibles *stencils* para construir $p_1(x, y)$ y $p_2(x, y)$, de forma que entre las tres posibilidades elegiremos aquel *stencil* $\{C_{n,l}^k, C_{n+1,l}^k, C_{n+2,l}^k, C_{n,l+1}^k, C_{n+1,l+2}^k, C_{n+2,l+1}^k, C_{n,l+2}^k, C_{n+1,l+2}^k, C_{n+2,l+2}^k\}$ que haga mínimo el siguiente valor:

$$\sum_{h=0}^2 \sum_{i=0}^1 |\bar{f}_{n+i+1,l+h}^k - \bar{f}_{n+i,l+h}^k| + \sum_{i=0}^2 \sum_{h=0}^1 |\bar{f}_{n+i,l+h+1}^k - \bar{f}_{n+i,l+h}^k|.$$

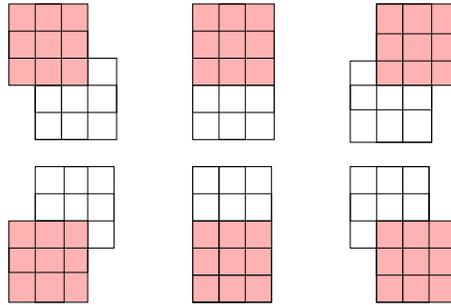


Figura 6.5: Posibles stencils para la construcción de $p_1(x, y)$ y $p_2(x, y)$ con contorno horizontal. La celda central del stencil blanco representa a $C_{i,j}^k$.

6.2.3

Consistencia de la predicción

Notar que a partir de la definición de $\bar{f}_{i,j}^k$ y $C_{i,j}^k$ es inmediato que los datos interpolados, $\{\hat{f}_{i,j}^{k+1}\}_{i,j=0}^{2^{k+1}}$, deben verificar la relación

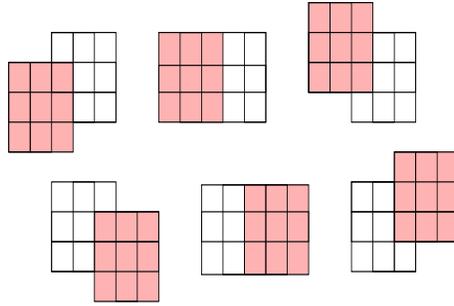


Figura 6.6: Posibles stencils para la construcción de $p_1(x, y)$ y $p_2(x, y)$ con contorno vertical. La celda central del stencil blanco representa a $C_{i,j}^k$.

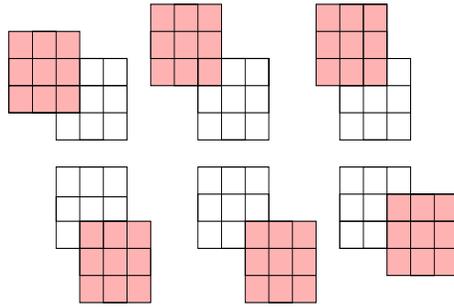


Figura 6.7: Posibles stencils para la construcción de $p_1(x, y)$ y $p_2(x, y)$ con contorno diagonal positivo. La celda central del stencil blanco representa a $C_{i,j}^k$.

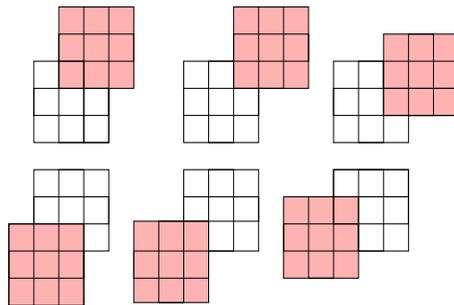


Figura 6.8: Posibles stencils para la construcción de $p_1(x, y)$ y $p_2(x, y)$ con contorno diagonal negativo. La celda central del stencil blanco representa a $C_{i,j}^k$.

6. Multirresolución 2D no separable para medias en celda 181

((2.15)):

$$\bar{f}_{i,j}^k = \frac{1}{4}(\hat{f}_{2i-1,2j-1}^{k+1} + \hat{f}_{2i,2j-1}^{k+1} + \hat{f}_{2i-1,2j}^{k+1} + \hat{f}_{2i,2j}^{k+1}) \quad (6.9)$$

El hecho de no tener en consideración el valor $\bar{f}_{i,j}^k$ al obtener las aproximaciones en el nivel de resolución $k+1$, $\{\bar{f}_{2i-h,2j-l}^{k+1}\}_{h,l=0,1}$, en aquellas celdas afectadas por un contorno, implica que no se verifique necesariamente (6.9). Para que los valores aproximados sean consistentes, modificamos la predicción de aquellas celdas por las que pasa la aproximación al perfil calculada previamente del siguiente modo: sea $d_{i,j}$ la diferencia entre $4\bar{f}_{i,j}^k$ y la suma de las aproximaciones obtenidas para las celdas $C_{n,m}^{k+1} \subset C_{i,j}^k$, i.e.

$$d_{i,j} = 4\bar{f}_{i,j}^k - (\hat{f}_{2i,2j}^{k+1} + \hat{f}_{2i-1,2j}^{k+1} + \hat{f}_{2i,2j-1}^{k+1} + \hat{f}_{2i-1,2j-1}^{k+1}).$$

Caso 1: Si la recta calculada no pasa por ninguna de las celdas $C_{2i-h,2j-l}^{k+1}$, $h, l = 0, 1$. Redefinimos las cuatro aproximaciones de la siguiente forma,

$$\tilde{f}_{2i-h,2j-l}^{k+1} = \hat{f}_{2i-h,2j-l}^{k+1} + \frac{d_{i,j}}{4}, \quad h, l = 0, 1. \quad (6.10)$$

Caso 2: Si la recta calculada pasa por tan solo una celda, sea ésta $C_{n_1,m_1}^{k+1} \subset C_{i,j}^k$, modificamos la media de dicha celda:

$$\tilde{f}_{n_1,m_1}^{k+1} = \hat{f}_{n_1,m_1}^{k+1} + d_{i,j}.$$

Caso 3: Si la recta cruza dos celdas, por ejemplo $C_{n_1,m_1}^k, C_{n_2,m_2}^k \subset C_{i,j}^{k+1}$, modificamos las medias correspondientes a C_{n_1,m_1}^{k+1} y C_{n_2,m_2}^{k+1} :

$$\tilde{f}_{n_l,m_l}^{k+1} = \hat{f}_{n_l,m_l}^{k+1} + \frac{d_{i,j}}{2}, \quad l = 1, 2.$$

Caso 4: Si la recta atraviesa todas las celdas excepto a C_{n_1,m_1}^{k+1} , modificamos las medias de las celdas $C_{n,m}^{k+1} \subset C_{i,j}^k$, $(n, m) \neq (n_1, m_1)$

$$\tilde{f}_{n,m}^{k+1} = \hat{f}_{n,m}^{k+1} + \frac{d_{i,j}}{3}.$$

De este modo es inmediato comprobar que $\{\tilde{f}_{2i-h,2j-l}^{k+1}\}_{h,l=0,1}$ verifican (6.9).

6.3

Detección de contornos

A continuación describimos brevemente el procedimiento para detectar los contornos de la imagen. En la sección 4.6 podemos encontrar de modo más detallado dicho proceso.

La detección realizada consta de tres fases:

Diferencias divididas : A partir de las diferencias divididas de primer orden, detectamos aquellas celdas susceptibles de contener un perfil.

Filtros de Sobel : Haciendo uso de los filtros de Sobel descartamos aquellas celdas marcadas en la fase anterior como consecuencia de la textura y clasificamos las celdas en cuatro tipos (contorno vertical, horizontal, diagonal positivo o diagonal negativo).

Eliminar celdas aisladas : Rechazamos aquellas celdas que son las únicas marcadas en una ventana de tamaño 5×5 .

6.4

Interpolación adaptada

Para calcular las aproximaciones de las medias de la función $f(x, y)$ en las celdas $\{C_{2i-h, 2j-l}^{k+1}\}_{h,l=0,1}$, tendremos en cuenta si $C_{i,j}^k$ pertenece a un contorno, está en las proximidades de un perfil o si podemos considerar que está lo suficientemente aislada de los perfiles de la imagen (figura 6.9). De este modo, clasificamos las celdas en tres tipos distintos, realizando en cada uno de ellos una predicción diferente:

Tipo I : Celdas pertenecientes a contornos. Las aproximaciones son calculadas del modo descrito en la sección 6.2, bien suponiendo que la imagen es constante, o bien que coincide con un polinomio a ambos lados del perfil

6. Multirresolución 2D no separable para medias en celda 183

Tipo II : Celdas que no pertenecen a contornos pero existe un contorno que cruza una ventana 3×3 centrada en ella. En este caso las aproximaciones se calculan a partir de una predicción ENO 2D con *stencils* 3×3 (sección 2.5).

Tipo III : Celdas que no pertenecen a contornos y la ventana 3×3 centrada en ella tampoco contiene ningún perfil. En este caso se aplica interpolación lineal dos-dimensional con *stencils* 3×3 centrados(ver ecuación (2.17)).

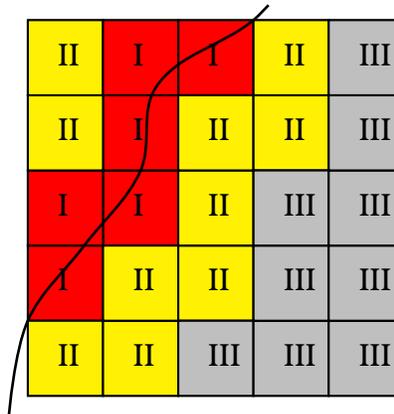


Figura 6.9: Interpolación empleada según la clasificación de la celda. Tipo I: interpolación adaptada a contornos. Tipo II: ENO 2D. Tipo III: Lineal con *stencils* 3×3 .

6.5

Resultados numéricos

Los apartados 6.5.1 y 6.5.2 muestran los resultados obtenidos al interpolar y comprimir imágenes aplicando diferentes técnicas interpolatorias. Las técnicas interpolatorias empleadas son las siguientes:

LIN3 : Interpolación lineal empleando *stencils* 3×3 (sección 2.4).

ENOSR : Interpolación ENO Subcell Resolution aplicada tensorialmente empleando polinomios interpoladores de grado 2.

EACT : Interpolación adaptada tal como se ha explicado en la sección 6.4. La predicción de las celdas *tipo III* (figura 6.9) se realiza suponiendo la imagen constante a ambos lados del contorno (Sección 6.2.1)

EA2P : Interpolación adaptada tal como se ha explicado en la sección 6.4. La predicción de las celdas *tipo III* (figura 6.9) se realiza suponiendo que la imagen es polinómica a trozos a ambos lados del perfil (Sección 6.2.2).

6.5.1

Interpolación de imágenes

Para realizar los tests hemos considerado cuatro imágenes distintas: dos de carácter geométrico (figura 6.10) y dos reales (figura 6.13). De las diferentes imágenes test, hemos tomado ventanas de tamaño $2^6 \times 2^6$ y se han aplicado recursivamente las diferentes técnicas interpolatorias hasta ampliar la ventana original en un factor de 8, i.e. $2^9 \times 2^9$ píxeles. Los resultados obtenidos en interpolación de imágenes han sido publicados en [6, 7] y presentados en el congreso Cedyá 2003 [9].

Imágenes geométricas

La figura 6.10 muestra las imágenes geométricas test y las ventanas empleadas para realizar las pruebas numéricas. La imagen *Geoma* es una imagen puramente geométrica mientras que la imagen *Geobl* es una imagen geométrica a la que se le ha añadido ruido.

En la figura 6.11 se puede observar que las técnicas EACT y EA2P presentan un comportamiento similar, obteniendo unos perfiles más definidos que al emplear LIN3 y eliminando además el efecto escalera que produce ENOSR como consecuencia de su aplicación tensorial.

6. Multirresolución 2D no separable para medias en celda 185

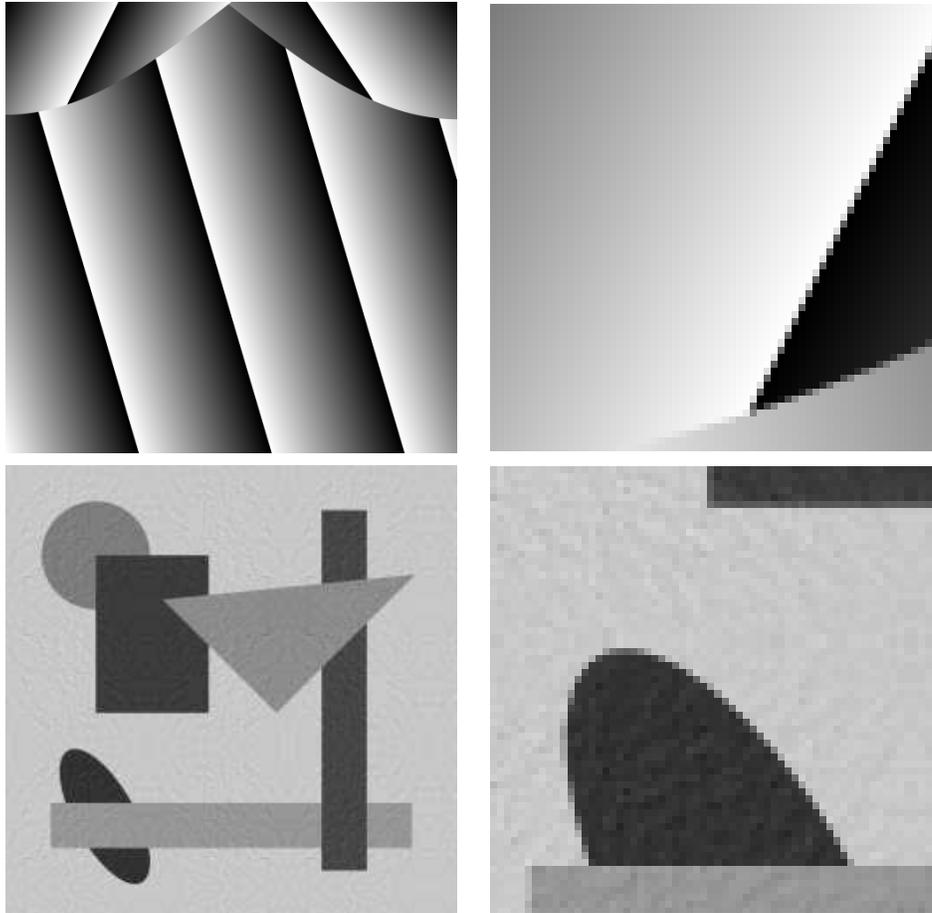


Figura 6.10: Derecha: de arriba a abajo. Geoma, Geobl. Izquierda: ventanas de 64×64 píxeles

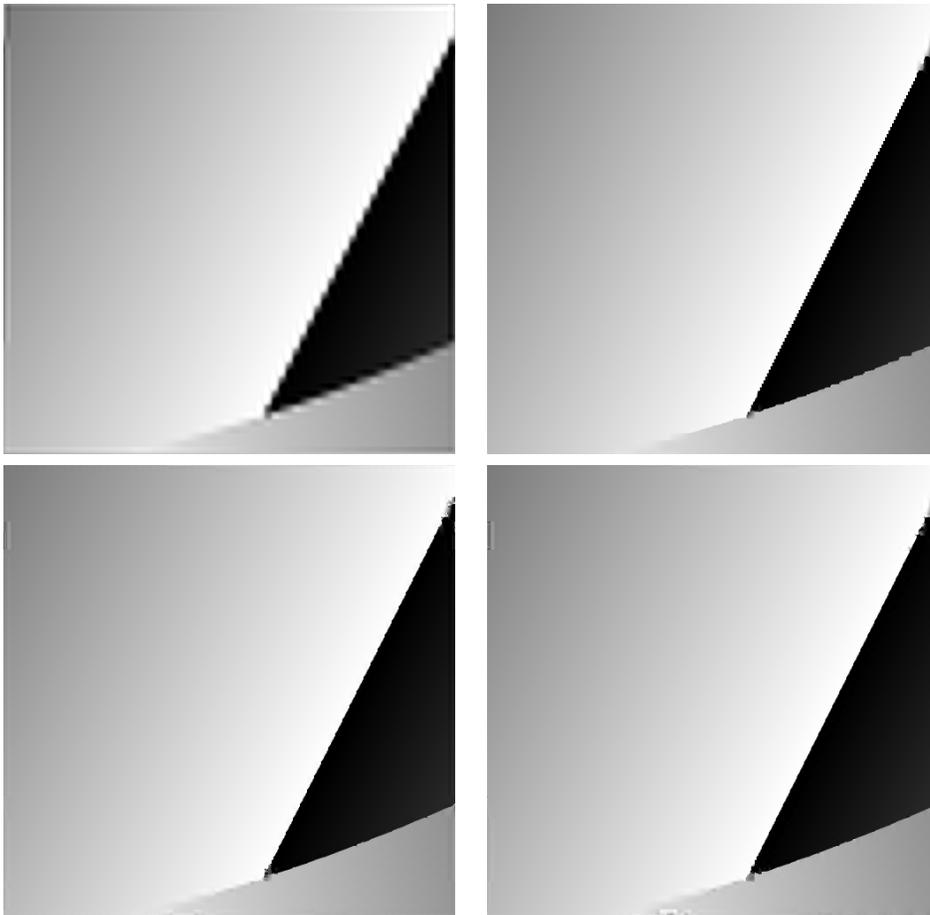


Figura 6.11: *Imagen Geoma. Arriba: LIN3, ENOSR. Abajo: EACT, EA2P*

6. Multirresolución 2D no separable para medias en celda 187

Al aplicar las diferentes técnicas interpolatorias a la imagen geométrica con ruido, figura 6.12, observamos un comportamiento similar: EACT y EA2P reducen el efecto escalera que produce ENOSR en la semielipse y consiguen unos contornos más nítidos. Notar que en este caso, EACT consigue un mejor resultado que EA2P al ampliar la parte inferior del cuadrado.

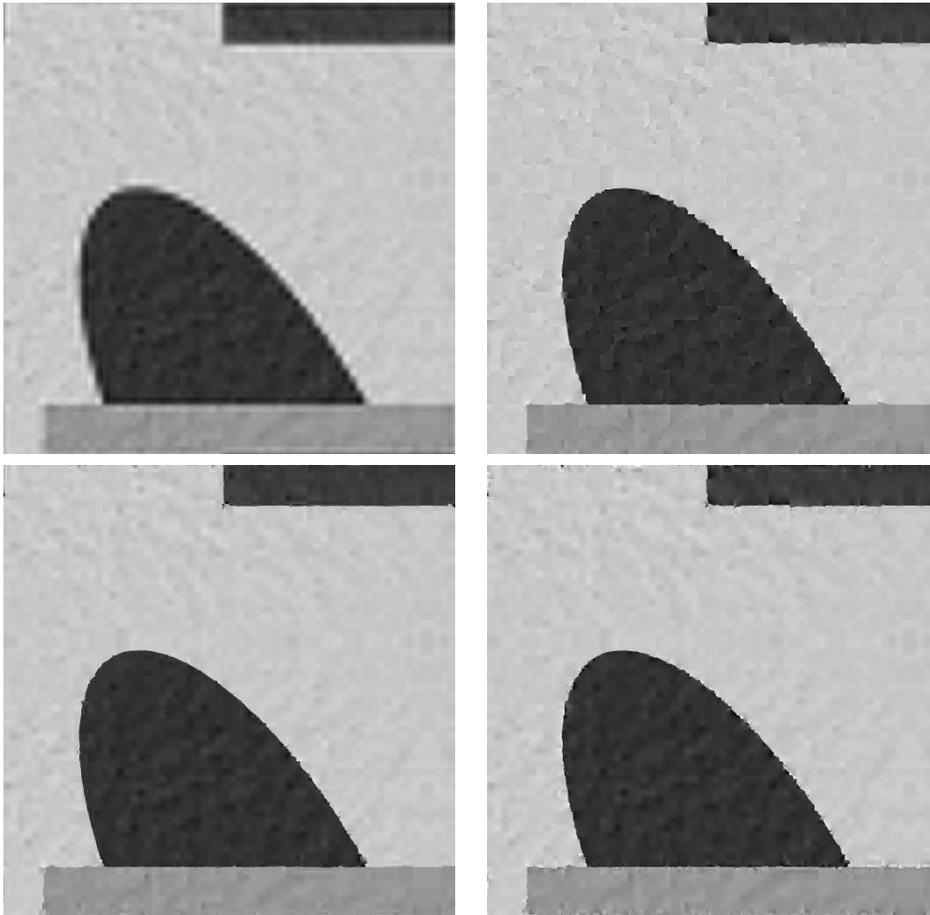


Figura 6.12: *Imagen Geobl. Arriba: LIN3, ENOSR. Abajo: EACT, EA2P*

Imágenes reales

La figura 6.13 muestra las imágenes test y las correspondientes ventanas de 64×64 píxeles.



Figura 6.13: Derecha: de arriba a abajo. Peppers, Camera. Izquierda: ventanas de 64×64 píxeles

6. Multirresolución 2D no separable para medias en celda 189

Los resultados obtenidos al ampliar las imágenes *Peppers* y *Camera* (figuras 6.14 y 6.15) son similares a los obtenidos con las imágenes geométricas. EACT y EA2P reducen la difusión producida por LIN3 y el efecto escalera de ENOSR. No obstante se puede observar que EACT produce unos artefactos en la imagen *Peppers* como consecuencia de suponer la imagen constante a ambos lados del contorno y aproximar los perfiles por líneas rectas.

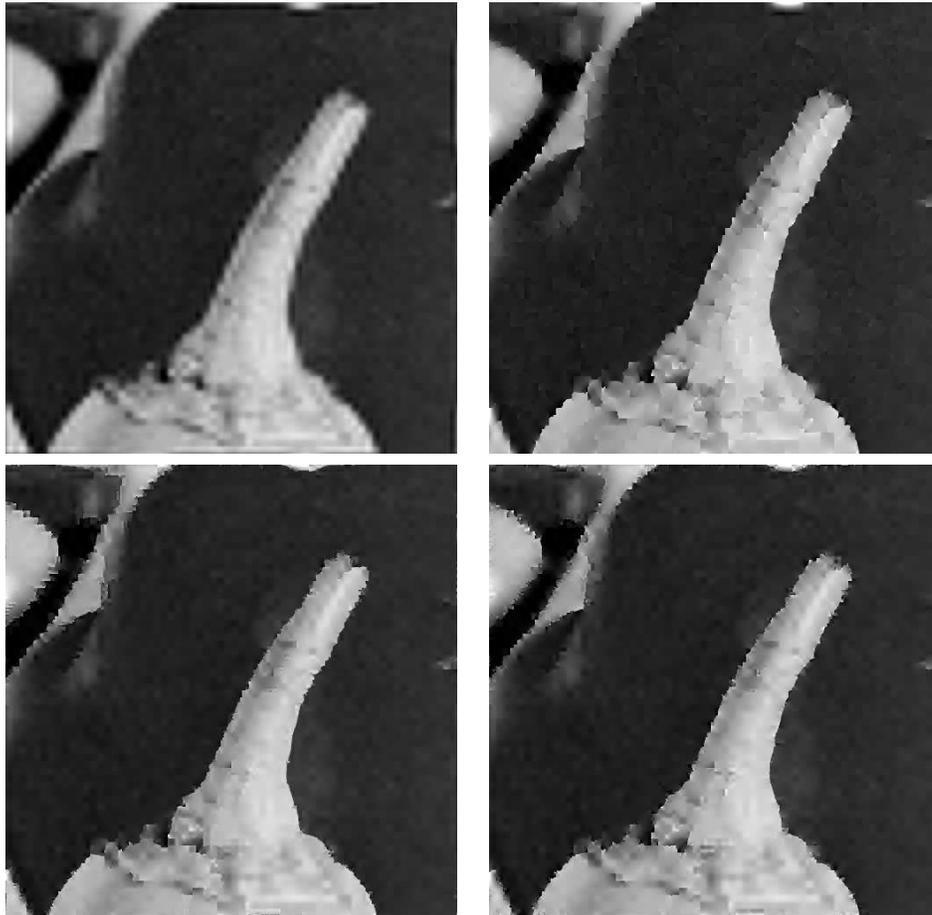


Figura 6.14: *Imagen Peppers. Arriba: LIN3, ENOSR. Abajo: EACT, EA2P*

Las mismas conclusiones son válidas para la imagen *Camera*.



Figura 6.15: *Imagen Camera. Arriba: LIN3, ENOSR. Abajo: EACT, EA2P*

Compresión de imágenes

A continuación mostramos los resultados obtenidos al emplear las técnicas interpolatorias LIN3, ENOSR, EACT y EA2P como operadores predicción en la multirresolución de Harten.

Para asegurar la estabilidad de las técnicas no lineales se ha utilizado el algoritmo de control del error ([1, 10, 11]). Los coeficientes *detalle* correspondientes al nivel de resolución k han sido cuantizados por el parámetro ϵ_k , i.e.

$$\hat{d}_{i,j}^k = qu(d_{i,j}^k; \epsilon_k) = 2\epsilon_k \left[\frac{d_{i,j}^k}{2\epsilon_k} \right],$$

donde $[\cdot]$ no es más que el entero obtenido por redondeo. Notar que así definidos se verifica $|\hat{d}_{i,j}^k - d_{i,j}^k| \leq \epsilon_k$. Fijado el parámetro de cuantización para el nivel más fino, ϵ_L , definimos los parámetros de cuantización de los niveles de resolución inferiores del siguiente modo:

$$\epsilon_{k-1} = \begin{cases} \epsilon_k/2 & \text{si } \epsilon_k \geq 1/2, \\ 1/2 & \text{otro caso.} \end{cases}$$

Los coeficientes correspondientes a la imagen en el nivel de resolución inferior más los detalles cuantizados han sido codificados con el algoritmo SPIHT [43] (Set Partitioning In Hierarchical Trees). En todas las pruebas las imágenes originales son de 512×512 píxeles y hemos considerado $k = 5$ el nivel de resolución inferior, i.e. 32×32 píxeles.

Imagen geométrica

La tabla 6.1 muestra los resultados obtenidos al aplicar los diferentes tests numéricos a la imagen geométrica *Geo*. En ella se puede observar que los resultados obtenidos con los métodos EACT y EA2P son mejores que los obtenidos con LIN3, tanto si comparamos el PSNR como si comparamos las tasas de compresión. Respecto al método ENOSR, los resultados obtenidos son similares a

los obtenidos por EACT, si bien EACT es ligeramente superior a ENOSR.

Las figuras 6.16 y 6.17 muestran, respectivamente, las reconstrucciones obtenidas para un PSNR similar y un detalle de éstas. Se puede observar que EACT y EA2P reconstruye de un modo más exacto la circunferencia y el triángulo. Notar que la reconstrucción de la parte superior del rectángulo hecha por EA2P no es del todo satisfactoria.

	$\ \cdot\ _\infty$	$\ \cdot\ _2$	PSNR	Nnz	bytes	bpp
LIN3	5,04	0,47	54,66	13654	12394	0,38
	7,43	0,64	52,01	12259	10169	0,31
	14,19	1,11	47,23	9755	7763	0,24
	31,39	2,02	42,04	7342	5472	0,17
	58,67	3,73	36,74	3892	3354	0,10
ENOSR	4,78	0,36	57,16	5099	5940	0,18
	8,93	0,44	55,21	4127	4788	0,15
	16	0,70	51,22	3418	3928	0,12
	32,01	1,33	45,69	2840	2840	0,10
	72,82	2,46	40,36	2213	2597	0,08
EACT	5,28	0,35	57,21	4087	4890	0,15
	8,60	0,43	55,56	3086	3854	0,12
	15,39	0,64	52,10	2535	3190	0,10
	31,61	1,20	46,60	2162	2688	0,08
	66,45	2,25	41,11	1815	2208	0,07
EA2P	5,30	0,38	56,58	5419	6227	0,19
	7,92	0,47	54,68	4136	4871	0,15
	16,83	0,74	50,75	3305	3920	0,12
	31,00	1,40	45,25	2673	3158	0,10
	64,82	2,59	39,91	2275	2565	0,08

Tabla 6.1: Imagen Geo. $\|\cdot\|_\infty$, $\|\cdot\|_2$, PSNR, coeficientes no nulos (Nnz), bytes y bits por pixel (bpp)

6. Multirresolución 2D no separable para medias en celda 193

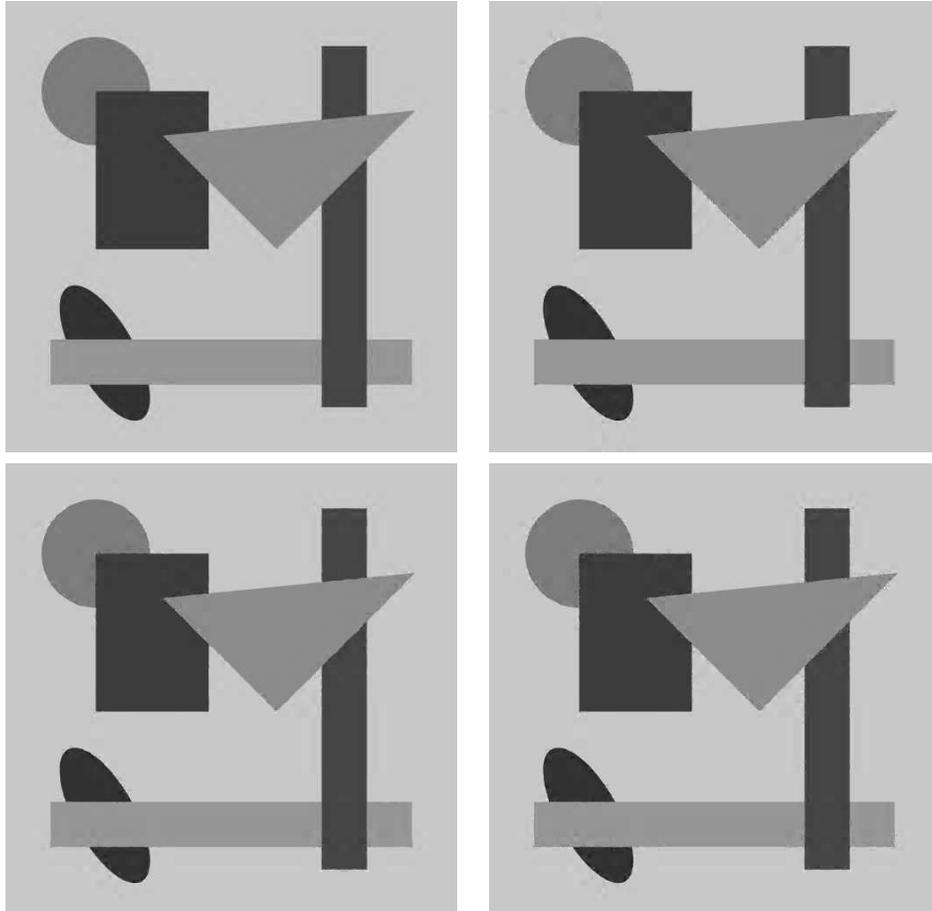


Figura 6.16: Reconstrucciones de Geo. Arriba: LIN3: PSNR 42,04, *bpp* 0,17. ENOSR: PSNR: 40,36, *bpp* 0,08. Abajo: EACT: PSNR 41,11, *bpp* 0,07. EA2P: PSNR 39,91, *bpp* 0,08.

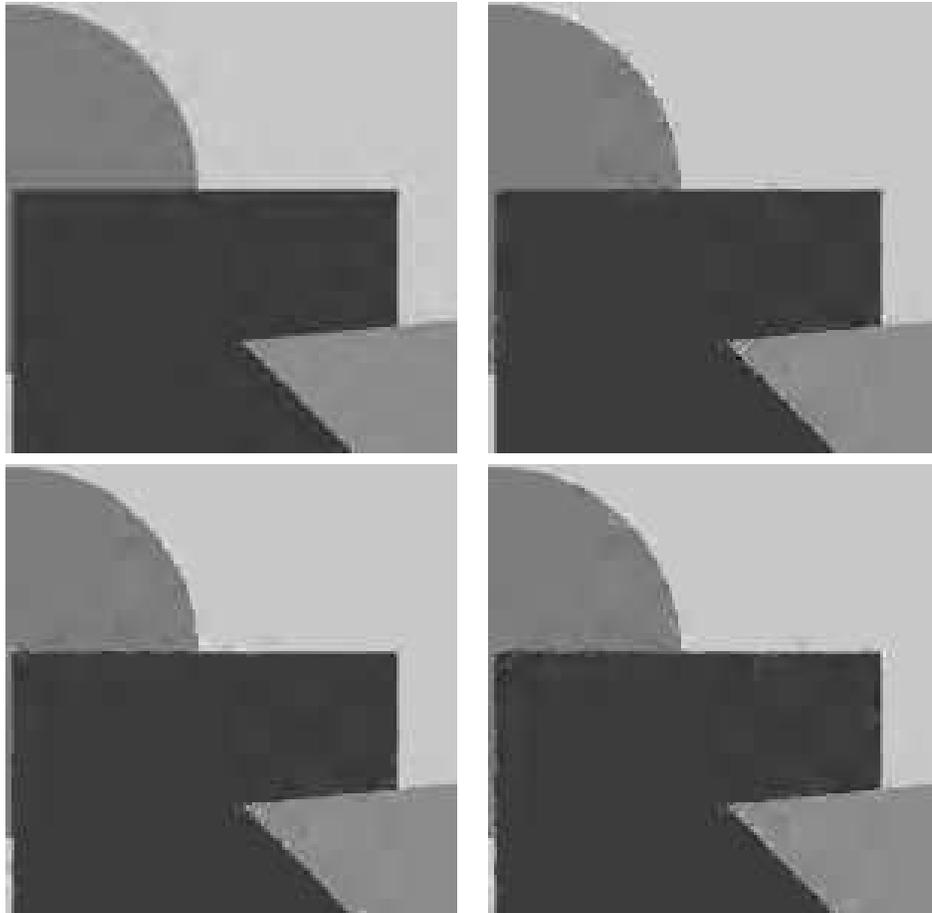


Figura 6.17: *Detalle de la reconstrucción de Geo. Arriba: LIN3, ENOSR. Abajo: EACT, EA2P*

6. Multirresolución 2D no separable para medias en celda 195

Imagen real

La tabla 6.2 recoge los resultados obtenidos con la imagen real *Peppers*. En ella podemos ver que los resultados obtenidos por LIN3, en términos de PSNR, son ligeramente superiores que los obtenidos por el resto de métodos. Lo mismo ocurre si miramos las tasas de compresión. Los métodos ENOSR, EACT y EA2P obtienen resultados parecidos.

	$\ \cdot\ _\infty$	$\ \cdot\ _2$	PSNR	Nnz	bytes	bpp
LIN3	5,45	1,26	46,15	124015	73102	2,23
	9,43	2,01	42,12	80221	49033	1,50
	16,41	3,30	37,78	42732	28852	0,88
	29,77	5,08	34,04	20634	15187	0,46
	59,91	7,33	30,86	10353	8043	0,25
ENOSR	5,87	1,29	45,95	135879	83947	2,56
	8,95	2,08	41,79	90873	57785	1,76
	17,45	3,50	37,28	49246	34830	1,06
	33,174	5,41	33,50	23916	18914	0,58
	60,42	7,96	30,14	12047	10375	0,32
EACT	5,46	1,26	46,16	126461	78956	2,41
	8,26	2,01	42,10	83744	54578	1,67
	17,18	3,34	37,68	46848	33756	1,03
	35,12	5,23	33,79	24619	19364	0,59
	62,40	7,88	30,24	13109	11023	0,34
EA2P	5,47	1,26	46,16	126691	78855	2,41
	9,65	2,01	42,10	83769	54355	1,66
	16,40	3,35	37,67	47002	33729	1,03
	32,79	5,24	33,77	24612	19384	0,59
	64,94	7,92	30,19	13242	11193	0,34

Tabla 6.2: Imagen Peppers. $\|\cdot\|_\infty$, $\|\cdot\|_2$, PSNR, coeficientes no nulos (Nnz), bytes y bits por pixel (bpp)

Si observamos las figuras 6.18 y 6.19, podemos ver las reconstrucciones obtenidas para un PSNR similar y un detalle de dichas reconstrucciones. Se puede apreciar que LIN3 presenta una mayor difusión en los perfiles de la imagen, siendo este reducido conside-

rablemente por el resto de métodos. Por otra parte la reconstrucción de los contornos obtenida por los métodos EACT y EA2P es mejor que la obtenida por ENOSR. Se observa también que EACT y EA2P reduce el número de artefactos producidos por el método ENOSR.



Figura 6.18: Reconstrucciones de Peppers. Arriba: LIN3: PSNR 27,69, *bpp* 0,14. ENOSR: PSNR: 26,71, *bpp* 0,18. Abajo: EACT: PSNR 26,67, *bpp* 0,19. EA2P: PSNR 26,56, *bpp* 0,19.

6. Multirresolución 2D no separable para medias en celda 197

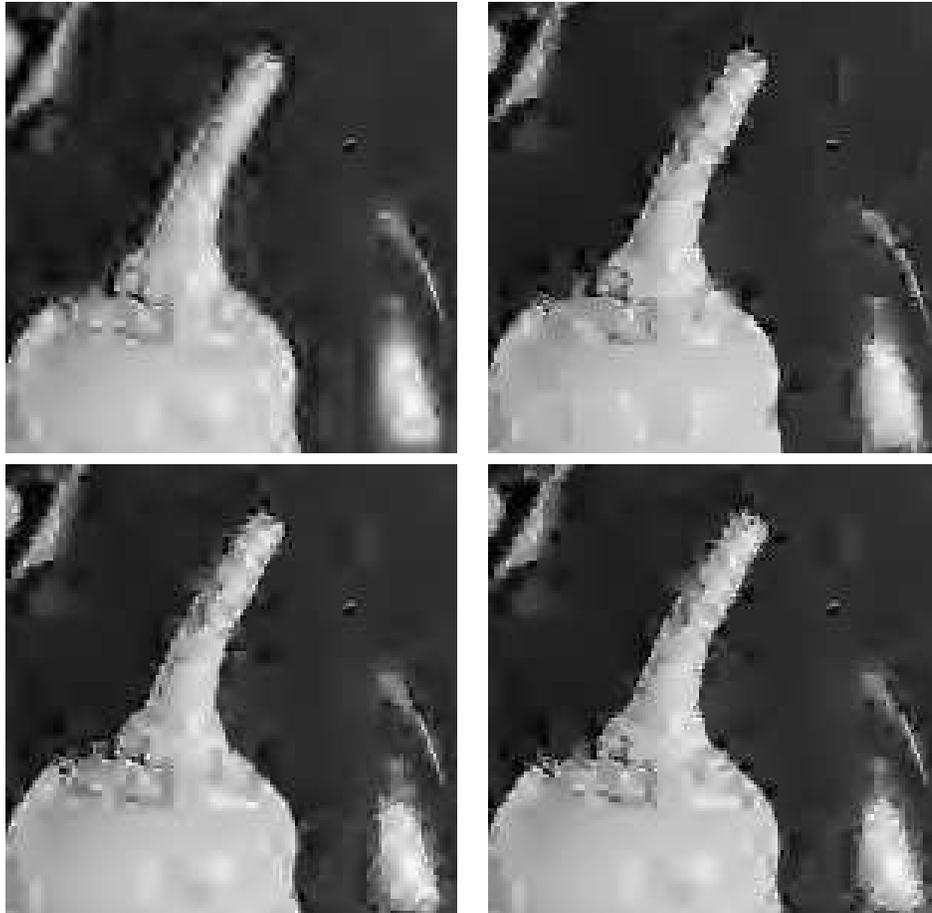


Figura 6.19: *Detalle de la reconstrucción de Peppers. Arriba: LIN3, ENOSR. Abajo: EACT, EA2P*

6.6

Conclusiones

En este capítulo hemos presentado un operador predicción dos-dimensional que utiliza la geometría de la imagen. Para ello detecta los contornos de la imagen a partir de la información proporcionada por el uso conjunto de las diferencias divididas de los datos y los filtros de Sobel (ver apartado 4.6), y posteriormente en aquellas celdas marcadas por la detección de contornos se aplica un operador predicción dos-dimensional en el que calcularemos las medias en las celdas que contienen los perfiles bajo dos supuestos: un primer caso, suponiendo que la imagen es constante a ambos lados del contorno, y un segundo caso, bajo el supuesto de que a ambos lados del perfil la imagen toma la forma de dos polinomios $p_1(x, y)$ y $p_2(x, y)$. En [22, 39, 6]) podemos encontrar algunos resultados en la misma línea que los presentados en este capítulo.

Los resultados obtenidos al ampliar imágenes proporcionan unos perfiles más definidos que los obtenidos mediante técnicas lineales o interpolación ENO-SR, eliminando a la vez el efecto escalera obtenido con ENO SR como producto de su aplicación tensorial. Al aplicar la predicción para comprimir imágenes los resultados no representan una gran mejora respecto el PSNR y la tasa de compresión, sin embargo si que se aprecia una mayor calidad visual en la reconstrucción de los contornos.

7

Conclusiones y perspectivas

En el presente trabajo de tesis doctoral se proponen diversas técnicas interpolatorias con el objetivo de aplicarlas a la ampliación y compresión de imágenes. Se han estudiado técnicas interpolatorias tanto en el contexto de interpolación de valores puntuales, como en el de medias en celda, aplicando en ambos casos, bien técnicas interpolatorias uno dimensionales de modo tensorial, o bien técnicas interpolatorias dos dimensionales.

En el contexto de interpolación de valores puntuales, se han propuesto diversas modificaciones de la interpolación racional propuesta por Ramponi et al. [19, 20, 21, 42]. En el Capítulo 1 se estudian los órdenes de interpolación de las modificaciones propuestas y se comparan los resultados obtenidos con otras técnicas, observando como los métodos derivados de la interpolación racional disminuyen el fenómeno de Gibbs que se presenta en

las discontinuidades al aplicar técnicas interpolatorias lineales. Se observa también, en la ampliación de imágenes, que las técnicas propuestas reducen la difusión de los contornos propia de las técnicas lineales, consiguiendo resultados comparables a los obtenidos por técnicas no lineales como ENO o WENO. Respecto a las técnicas interpolatorias dos dimensionales, se proponen diversas formas de seleccionar los *stencils*, así como la implementación de dichas técnicas a partir de los filtros empleados por técnicas uno dimensionales. Los resultados obtenidos por la interpolación dos dimensional propuesta, ENO2DT, mejoran ligeramente los resultados conseguidos con la aplicación tensorial de la técnica ENO.

En el campo de la compresión de imágenes, se comparan los resultados al emplear las diferentes técnicas como operadores predicción en la multirresolución de Harten. En éstos se observa como las modificaciones de la interpolación racional reducen la difusión de los contornos producida por las técnicas lineales y obtienen resultados similares a las técnicas no lineales con las que se comparan. Además se propone una modificación del algoritmo de control del error empleado en la multirresolución de Harten para asegurar la estabilidad al emplear operadores predicción no lineales. Al comparar los resultados obtenidos por la modificación del algoritmo de error control con el algoritmo sin modificar, se observa una mejora en la tasa de compresión.

Respecto a la interpolación de medias en celda, se proponen dos modificaciones de la interpolación racional, originalmente diseñada para la interpolación de valores puntuales, analizándose en cada caso los órdenes de interpolación alcanzados. Se observa que al aplicarlas en la ampliación de señales 1D, desaparece el fenómeno de Gibbs, y en la ampliación de imágenes, se reduce la difusión de los contornos presentes al aplicar interpolación lineal. Los resultados obtenidos en la compresión de imágenes se muestran en el Capítulo 3. En el caso de la imagen real, se observa una disminución del error en $\|\cdot\|_2$ y de la tasa de compresión al emplear las técnicas racionales.

Con el objetivo de desarrollar técnicas interpolatorias 2D no separables, en el Capítulo 4 se proponen dos algoritmos para la detección de contornos. Dichos algoritmos son aplicados en los Capítulos 5 y 6.

El Capítulo 5 muestra un algoritmo 2D no separable, para el contexto de valores puntuales. Los resultados obtenidos se comparan con los conseguidos al aplicar interpolación lineal e interpolación ENO. En imágenes geométricas, se consigue tanto visualmente como en términos de tasa de compresión una mejora significativa respecto las otras técnicas. Al aplicar el algoritmo a una imagen real, si bien numéricamente los resultados son similares a los obtenidos con ENO, si que supone visualmente una mejora respecto la interpolación lineal y ENO.

El Capítulo 6 desarrolla un algoritmo de predicción 2D no separable, en este caso diseñado para la interpolación de medias en celda. Se estudian dos formas diferentes de realizar la predicción en aquellas celdas que contienen un perfil de la imagen: la primera suponiendo la imagen constante a ambos lados del perfil, y la segunda suponiéndola polinómica a trozos. En la ampliación de imágenes, se observa una disminución del efecto escalera presente en los contornos de la imagen al aplicar tensorialmente técnicas uno dimensionales. Respecto a la compresión de imágenes, si bien los resultados obtenidos, en términos de PSNR y tasa de compresión, no suponen una gran mejora respecto a otras técnicas, las reconstrucciones poseen una mayor calidad visual.

Respecto al trabajo futuro, sería interesante estudiar nuevos pesos para las técnicas interpolatorias racionales, de modo que la contribución de los polinomios afectados por discontinuidades fueran lo menor posible.

Por otra parte, en la modificación del algoritmo de control del error para valores puntuales se podría estudiar el incorporar un proceso detección de contornos, para así poder emplear la información dada por la localización de los perfiles en la segunda etapa de la predicción, esperando obtener así una mejora tanto en la tasa de compresión como en el PSNR.

La extensión de las aplicaciones vistas a las imágenes en color constituiría otra posible ampliación del trabajo, en este caso se debería aplicar la técnica interpolatoria deseada a los diferentes canales de color de la imagen.

Bibliografía

- [1] S. Amat, F. Aràndiga, A. Cohen, and R. Donat. Tensor product multiresolution analysis with error control for compact image representation. *Signal Processing*, 82:587–608, 2002.
- [2] S. Amat, F. Aràndiga, A. Cohen, R. Donat, G. Garcia, and M.v. Oehnsen. Data compression with ENO Schemes: A case study. *Applied and Computational Harmonic Analysis*, 11:273–288, 2001.
- [3] F. Aràndiga and A. Belda. Weighted ENO interpolation and applications. *Communications in Nonlinear Science and Numerical Simulation*, 9(2):187–195, 2003.
- [4] F. Aràndiga, A. Belda, and P. Mulet. Point-value WENO multi-resolution applications to stable image compression. *Journal of Scientific Computing*, 43(2):158–182, 2010.
- [5] F. Aràndiga, J. Baccou, M. Doblas, and J. Liandrat. Image compression based on a multi-directional map-dependent algorithm. *Applied and Computational Harmonic Analysis*, 23:181–197, 2007.
- [6] F. Aràndiga, A. Cohen, M. Doblas, R. Donat, and B. Matei. Edge-adapted nonlinear multiscale transform for compact representations of images. In *IEEE International Conference on Image Processing*, volume 1, pages 701–704, Barcelona, 2003.

- [7] F. Aràndiga and M. Doblas. Interpolación adaptativa de imágenes. *Revista de la Facultat de Matemàtiques de València*, 2(1):99–114, 2004.
- [8] F. Aràndiga and M. Doblas. Compresión de imágenes mediante multirresolución de Harten. In *Cedyà 2005, XIX Congreso de Ecuaciones Diferenciales y Aplicaciones, IX Congreso de la Sociedad Española de Matemática Aplicada*, Madrid, 2005.
- [9] F. Aràndiga, M. Doblas, and P. Mulet. Interpolación adaptativa de imágenes. In *Cedyà 2003, XVIII Congreso de Ecuaciones Diferenciales y Aplicaciones, VIII Congreso de Matemática Aplicada*, Tarragona, 2003.
- [10] F. Aràndiga and R. Donat. Nonlinear multiscale decompositions: The approach of Ami Harten. *Numerical Algorithms*, 23:175–216, 2000.
- [11] F. Aràndiga and R. Donat. Cell average data compression algorithms with exact error control. *GrAN report*, 2003.
- [12] F. Aràndiga, R. Donat, and P. Mulet. Adaptive interpolation of images. *Signal Processing*, 83:459–464, 2003.
- [13] J. Baccou. *Analyses Multirésolutions et problèmes de bord: applications au Traitement d’Images et à la Résolution d’Equations aux Dérivées Partielles*. PhD thesis, Université de Provence, 2004.
- [14] J. Baccou and J. Liandrat. Position dependent Lagrange interpolating multi-resolutions. *Int. J. Wavelets Multiresolut. Inf. Proc.*, 5(4):1–27, 2007.
- [15] A. Belda. *Técnicas de interpolación WENO y su aplicación al procesamiento de imágenes*. PhD thesis, Universitat de València, 2010.
- [16] B.L. Bihari and A. Harten. Multiresolution schemes for the numerical solution of 2-D conservation laws I. *J. Sci. Comput.*, 18(2):315–354, 1997.

- [17] C. Bloom. Solving the problems of context modelling. <http://www.cbloom.com/papers>, 1998.
- [18] J.F. Canny. Finding Edges and Lines in images. Technical Report 720, MIT Artificial Intelligence Laboratory, 1983.
- [19] S. Carrato and G. Ramponi. Interpolation of the DC Component of Coded Images Usin a Rational Filter. *Proc. Fourth IEEE Intern. Conf. on Image Processing - 97*, pages 26–29, S. Barbara, CA, Oct. 1997.
- [20] R. Castagno, S. Marsi, and G. Ramponi. A Simple Algorithm for the Reduction of Blocking Artifacts in Images and its Implementation. *IEEE Trans. on Consumer Electronics*, 44(3):1062–1070, Aug. 1998.
- [21] R. Castagno and G. Ramponi. A rational filter for the removal of blocking artifacts in image sequences coded at low bitrate. *Proc. Eighth European Signal Processing Conf., EUSIPCO-96*, Trieste, Italy, Sept. 10-13 1996.
- [22] A. Cohen and B. Matei. Compact representations of images by edge adapted multiscale transforms. In *IEEE-ICIP conference*, Thessaloniki, 2001.
- [23] A. Cohen, B. Matei, and N. Dyn. Quasilinear subdivision schemes with applications to ENO interpolation. *Appl. Comput. Harmon. Anal.*, 15:89–116, 2003.
- [24] G. Deslauriers and S. Dubuc. Interpolation dyadique. In *Fractals, dimensions non entières et applications*, pages 44–45, Masson, Paris 1987.
- [25] M.N. Do and M. Vetterli. The contourlet transform: an efficient directional multiresolution image representation. *IEEE-Trans. on image processing*, 14:2091–2106, 2005.
- [26] N. Dyn. *Subdivision schemes in computer-aided geometric design*. In W.A Light, editor, *Advances in Numerical Analysis II, Wavelets, Subdivision algorithms and Radial Basis functions*. Clarendon Press, Oxford, 1992.

- [27] M. Gasca and T. Sauer. On the history of multivariate polynomial interpolation. *Journal of computational and applied mathematics*, pages 23–35, 2000.
- [28] A. Harten. ENO schemes with subcell resolution. *Journal of Computational Physics*, 71:148–184, 1989.
- [29] A. Harten. Discrete multiresolution analysis and generalized wavelets. *J. Applied Num. Math.*, 12:153–193, 1993.
- [30] A. Harten. Multiresolution representation of data: a general framework. *SIAM J. Numer. Anal.*, 33(3):1205–1256, 1996.
- [31] A. Harten, B. Engquist, S. Osher, and S. Chakravarthy. Uniformly high order accurate essentially non-oscillatory schemes III. *J. Comput. Physics*, 71:231–303, 1987.
- [32] G-S. Jiang and C-W. Shu. Efficient implementation of Weighted ENO Schemes. *J. Comput. Physics*, 115:202–228, 1994.
- [33] R. Estes Jr. and V. Ralph Agazzi. Efficient error free chain coding of binary documents. *Proc. Data Compression Conference, Snowbird, Utah, IEE Computer Society Press, Los Alamitos, CA*, pages 122–132, 1995.
- [34] X-D. Liu, S. Osher S., and T. Chan. Weighted Essentially Non-oscillatory Schemes. *J. Comput Physics*, 115:200–212, 1994.
- [35] S. Mallat. A theory for multiscale signal decomposition: The wavelet representation. *IEEE Trans. on Pattern. and Machine Intelligence*, 11(7):674–693, 1989.
- [36] S. Mallat. *A Wavelet Tour of Signal Processing*. Academic Press, San Diego, 1998.
- [37] S. Mallat and W. L. Hwang. Singularity detection and processing with wavelet. *IEEE Trans. Info. Theory.*, 38(2):617–643, 1992.
- [38] S. Mallat and S. Zhong. Characterization of signals from multiscale edges. *IEEE Trans. Patt. Anal. and Mach. Intell.*, 14(7):710–732, 1992.

- [39] B. Matei. *Methodes multiresolutions non-linéaires et applications au traitement d'image*. PhD thesis, Université Paris VI, 2002.
- [40] E. Le Pennec and S. Mallat. Sparse geometric image representations with bandelets. *IEEE-Trans. on image processing*, 14(4):423–438, 2005.
- [41] A. Peyronne. Utilisation de méthodes adaptatives pour l'agrandissement d'images. *GrAN report*, 01-02, 2001.
- [42] G. Ramponi. Image Processing using Rational Functions. *Proc. Second IEEE Intern. Conf. on Image Processing - 95*, pages 22–25, Washington, DC, USA, Oct 1995.
- [43] A. Said and W. A. Pearlman. A new fast and efficient image codec based on set partitioning in hierarchical trees. *IEEE Transactions on Circuits and Systems for Video Technology*, 2:243–250, 1996.
- [44] T. Sauer and Y. Xu. On multivariate Lagrange interpolation. *Math. Comp.*, 64:1147–1170, 1995.
- [45] J.M. Shapiro. Embedded image coding using zerotrees of wavelet coefficients. *IEEE Trans. Signal Processing*, 41(12):3445–3462, 1993.
- [46] D. Taubman. Embedded block coding with optimized truncation. *Signal Processing: Imagen Communication*, Elsevier, 1998.
- [47] S. Thurnhofer, M. Lighstone, and S.K. Mitra. Adaptive Interpolation of Images with Application to Interlaced-to-Progressive Conversion. *Proc. SPIE Symp. Visual Communications and Image Processing*, pages 614–625, 1993.
- [48] M. J. Weinberg, G. Seroussi, and G. Sapiro. From loco-i to the jpeg-ls standard. *Proc. of the International Conference on Image Processing*, 4:68–72, 1999.