

## Tratamiento de imágenes mediante algoritmos basados en el cálculo de áreas

MARÍA MONCAYO<sup>1</sup>, ANTONIO MOLINA<sup>2</sup> Y JUAN FRANCISCO REINOSO<sup>3</sup>

1. Departamento de Matemática Aplicada y Estadística.  
Universidad Politécnica de Cartagena.
2. Departamento de Matemáticas.  
Universidad de Jaén.
3. Departamento de Ingeniería Cartográfica, Geodésica y Fotogrametría.  
Universidad de Granada.

maria.moncayo@upct.es; amtebar@ujaen.es; jreinoso@ugr.es

### Resumen

Codificar una imagen básicamente consiste en obtener una representación de la misma que esté bien adaptada al problema que se maneje. Por ejemplo, comprimirla, eliminar el ruido o borrosidades que pueda presentar, detectar bordes o cambios de color, restaurar partes que estén deterioradas, etc. Entre los algoritmos que suelen aplicarse con éxito en varios de los problemas mencionados, se encuentra un tipo de algoritmos multirresolución cuya definición es posible gracias al teorema fundamental del Cálculo. Este teorema constituye una valiosa joya del análisis infinitesimal y determina el hecho de que derivadas e integrales son procesos inversos. El objetivo del artículo en curso es presentar de manera gradual este algoritmo y realizar con él alguna aplicación propia del tratamiento de imágenes.

**Proyecto/Grupo de investigación:** Aproximaciones no lineales para la reconstrucción de datos discontinuos. Entidad financiadora: Ministerio de Ciencia e Innovación, MICINN-FEDER. Código: MTM2010-17508.

**Líneas de investigación:** *Tratamiento de imágenes; Análisis multirresolución; Teoría de aproximación; Análisis infinitesimal.*

## 1 Imágenes y matrices de números naturales

Toda aquella persona que haya reducido la calidad de una fotografía presente en la pantalla de su ordenador, habrá podido comprobar la aparición de unos pequeños cuadrados a lo largo y ancho de la imagen. Si se subdivide la imagen

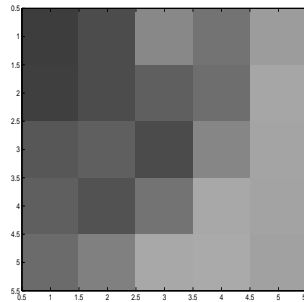


Figura 1: Ojo izquierdo de Lena (detalle) y niveles de gris de cada píxel

en  $NM$  cuadrados, con  $N, M \in \mathbb{N}$ , y a cada cuadrado se le asigna un número, entonces tendremos una codificación de la imagen en términos de una matriz, con  $N$  filas y  $M$  columnas. Con idea de que la matriz arroje cierta información sobre la imagen, el número asignado no es arbitrario y suele estar relacionado con el nivel de gris presente en el cuadrado correspondiente. En la figura 1 se representa parte del ojo izquierdo de Lena<sup>1</sup> con los distintos píxeles que definen el ojo.

Se suele convenir que el número 0 represente al color blanco y el 255 represente al negro. De este modo, los números enteros  $1, 2, \dots, 254$  definen los niveles de gris intermedio. La matriz  $5 \times 5$  de la tabla 1 corresponde al valor de los niveles de gris de los píxeles de la figura 1.

195	179	120	141	100
193	179	161	146	90
169	160	181	122	92
160	174	141	88	93
149	127	88	86	95

Tabla 1. Matriz con los niveles de gris de cada píxel de la figura 1.

De este modo, la imagen se define matemáticamente como una aplicación, cuyo dominio es ahora una matriz arbitraria:

$$I_d : \{0, 1, \dots, N - 1\} \times \{0, 1, \dots, M - 1\} \mapsto [0, 255] \cap \mathbb{N}$$

Es habitual considerar  $M = N = 2^L$ . El número  $L$  se conoce como nivel de resolución. En el caso de trabajar con entradas naturales, la representación se denomina digital. La generalización al caso real proporciona la definición de imagen analógica y se representa mediante la aplicación siguiente:

$$I_a : D \subset \mathbb{R} \times \mathbb{R} \mapsto [0, 255] \subset \mathbb{R}.$$

<sup>1</sup>Lena aparece en todas las librerías de imágenes como figura test.

En estas líneas consideraremos imágenes codificadas mediante matrices de números naturales. Aunque parezca una obviedad, debemos tener presente que la representación unidimensional –por ejemplo de una de las filas de la matriz– no es más que una función constante a trozos o *escalonada*. La figura 2 representa las cinco componentes de la primera fila de la matriz definida en la Tabla 1. Si el soporte de cada *escalón* tiene longitud igual a la unidad, el área que encierra coincide con el valor presente en la matriz, i.e., el valor que toma la función constante a trozos sobre el eje de ordenadas.

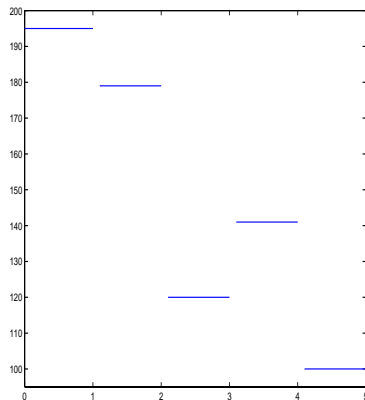


Figura 2: Función escalonada definida con las cinco componentes de la primera fila de la matriz

Los algoritmos lineales que consideran datos y salidas definidos mediante matrices suelen estudiarse en primer lugar para el caso unidimensional. Posteriormente se llevan hasta dimensión dos aplicando un razonamiento estándar de producto tensor. En esta ocasión nos vamos a ceñir a este guión. Por tanto, el algoritmo objeto de este artículo se va a definir en dimensión uno. De modo genérico puede decirse que un algoritmo multirresolución es aquél que conecta –a dos resoluciones consecutivas– los datos de un problema. Un proceso iterativo nos permite conectar los datos a distintos niveles de resolución. La palabra *multirresolución* queda, de esta manera, sobradamente justificada. En la siguiente sección se introducen con más precisión las claves del análisis multirresolución.

## 2 Análisis multirresolución

Las técnicas multiescala desempeñan un papel importante en el Análisis Numérico. La idea original de la definición de transformadas multirresolución consiste en comparar –vía producto escalar– una señal dada con funciones muy localizadas en tiempo y en frecuencia (wavelets). De este modo es posible obtener información conjunta en las dos variables. La comparación con exponenciales

dependientes únicamente de la variable frecuencia nos lleva directamente hasta la clásica transformada de Fourier.

Esa idea original se ha desarrollado de forma espectacular desde que se introdujo a finales de los años 80 [4], [7]. Lo que parecía una teoría repleta de artificios y *tricks* resultó ser algo susceptible de ser generalizado dentro de un marco muy flexible y natural. Es lo que se ha venido haciendo en lo que se conoce como la segunda generación de wavelets o wavelets generalizadas [5], [6], [8].

Analizar una señal a más o menos resolución depende del número de coeficientes *wavelet* (no de Fourier) utilizados en su representación en series de wavelets. El número de coeficientes está directamente relacionado con la dimensión del espacio vectorial al que pertenecen las funciones empleadas en la serie. Estos espacios vectoriales se denominan más específicamente espacios de aproximación. Pertenecer a uno de ellos implica poseer una determinada resolución.

Más precisamente, si se considera el caso unidimensional y el espacio de señales de energía finita  $L^2(\mathbb{R})$ , los espacios de aproximación se definen formando una cadena:

$$\cdots \subset V_{-2} \subset V_{-1} \subset V_0 \subset V_1 \subset V_2 \subset \cdots \subset V_k \subset V_{k+1} \subset \cdots \subset L^2(\mathbb{R}),$$

y asume que

$$\overline{\bigcup_{k \in \mathbb{Z}} V_k} = L^2(\mathbb{R}), \quad \bigcap_{k \in \mathbb{Z}} V_k = \{0\}.$$

Los espacios  $V_k$  son espacios de aproximación en los que se proyectan señales que tienen resolución  $2^k$ ,  $k \in \mathbb{Z}$ . A medida que  $k$  crece, los espacios ganan en resolución. De este modo, si  $f \in L^2(\mathbb{R})$ , la proyección de  $f$  sobre  $V_k$  y que denotaremos  $f^k$ , tiene menor resolución que  $f^{k+1}$ , es decir, que su proyección sobre  $V_{k+1}$ . Aunque no es el objetivo de estas páginas entrar minuciosamente en definiciones y teoremas, no podemos dejar de mencionar que la definición matemática de resolución tiene que ver con el número de elementos que conforman una base de cada espacio de aproximación. Por comodidad de notación, identificaremos las proyecciones  $f^k$  con los coeficientes que la determinan.

En el análisis multirresolución tanto o más importantes que los espacios  $V_k$  son los espacios intermedios entre dos espacios de aproximación consecutivos. Son los denominados *espacios de detalle*. Es decir, los espacios  $W_k$  tales que

$$V_k + W_k = V_{k+1}. \quad (1)$$

En el caso de que la suma anterior sea suma directa, entonces el marco de análisis multirresolución que se define es ortogonal. En general  $W_j$  no tiene por

qué ser el complemento ortogonal de  $V_k$  en  $V_{k+1}$ . Al igual que  $f^k$  representan<sup>2</sup> los coeficientes de la proyección de  $f$  sobre  $V_k$ , tenemos también que  $d^k$  representan los coeficientes de la proyección de  $f$  sobre  $W_j$ . De este modo, y de acuerdo con (1), se tiene que:

$$\begin{aligned} \{f^k, d^k\} & \text{ contiene la misma información que } f^{k+1} \text{ y} \\ \{f^k, d^k, \dots, d^0\} & \text{ contiene la misma información que } f^{k+1}. \end{aligned}$$

Dado un vector original de datos,  $f^k = (f_j^k)$ , con  $k \in \mathbb{N}$ , una representación multirresolución de  $f^k$  es una sucesión

$$\{f^{k-1}, d^{k-1}, d^{k-2}, \dots, d^1, d^0\}.$$

De modo genérico, un algoritmo multirresolución relaciona los coeficientes  $f_j^k$  y  $d_j^k$  con  $f_j^{k+1}$ . Teniendo en cuenta (1), el procedimiento para obtener  $f_k^k$  a partir de  $\{f_k^{k-1}, d_k^{k-1}\}$  se denomina reconstrucción. Recíprocamente, la obtención de  $\{f_k^{k-1}, d_k^{k-1}\}$  a partir de  $f_k^k$  se denomina descomposición. Esquemáticamente, tenemos los algoritmos de descomposición y reconstrucción como sigue:

$$\begin{aligned} f^k & \longrightarrow \{f^{k-1}, d^{k-1}\} \text{ Descomposición (pérdida de resolución),} \\ f^k & \longleftarrow \{f^{k-1}, d^{k-1}\} \text{ Reconstrucción (incremento de resolución).} \end{aligned}$$

Hasta ahora, todo lo que se ha comentado no deja de ser un ejercicio puramente algebraico. Es muy relevante el hecho de que verdaderamente existan bases de funciones que generan a los espacios de aproximación y otras bases, distintas a las anteriores pero relacionadas entre sí, tales que generan a los espacios de detalle. Justamente, éstas últimas son las bases wavelet. La construcción de tales bases conforman la teoría relacionada con la primera generación de wavelets, siendo referencias obligadas [4] y [7]. Alguna de estas construcciones son poco amigables en el sentido de ser muy artificiosas. Nada que ver con lo que sucede en el marco que generaliza a esta primera generación de wavelets. Si bien es cierto que “no debe generalizarse por generalizar”, en muchas ocasiones la generalización permite encontrar teorías más flexibles que dan respuestas a problemas de muy diversa índole. La idea que hay detrás de la superación de la primera generación de wavelets es la siguiente: Puede pensarse también que -en definitiva- trabajar con más o menos resolución depende simplemente de la “cantidad de información” que se emplee. La cantidad de información fiable en una señal unidimensional puede venir dada por la cantidad de evaluaciones de las que dispongamos [1], [2]. Un proceso -por ejemplo de interpolación- entre valores conocidos nos permite ganar información (o resolución). Es lo que se conoce como análisis multirresolución mediante valores puntuales y está directamente relacionado con el análisis multirresolución basado en el cálculo de áreas. Los concretamos en la siguiente sección. Un estudio más completo lo encontramos en [5] y [6].

<sup>2</sup>Es importante resaltar que los valores  $f^k$  representan determinada cantidad relacionada con  $f$  (no necesariamente evaluaciones de  $f$ ). Esto es algo que no debe sorprender si pensamos que los procesos de discretización dependen de la naturaleza de los datos.

## 2.1 Análisis multirresolución mediante cálculo de áreas

Sea  $N \in \mathbb{N} \cup \{0\}$ . Se define una partición uniforme del intervalo real  $[0, 1]$  que emplea  $2^N + 1$  puntos distanciados entre sí una cantidad fija  $h_N = 2^{-N}$ . Denotaremos a esta partición  $X^N$ . Así,

$$X^N := \{x_j^N\}_{j=0}^{2^N} \quad \text{y} \quad x_j^N = jh_N.$$

Al variar el número natural  $N$  se consiguen definir más o menos puntos en  $[0, 1]$ , originando mallas de mayor o menor resolución. En concreto, las mallas de menor resolución que  $X^N$  se definen como particiones  $X^k$  con  $k = 0, 1, \dots, N - 1$ . Las mallas a dos resoluciones consecutivas se relacionan mediante

$$x_{2j}^k = x_j^{k-1}; \quad (j = 0, \dots, 2^{k-1}). \quad (2)$$

Por tanto, basta con eliminar las componentes impares de la malla de mayor resolución para obtener las componentes de la malla de menor resolución. Conviene notar que, al igual que sucede con los espacios de aproximación  $V_k$ , las mallas  $X^k$  también están *anidadas*, i.e., forman la cadena

$$X^0 \subset X^1 \subset X^2 \subset \dots \subset X^{k-1} \subset X^k \subset \dots$$

Según sea el problema en el que se trabaje, esto es, según sea la naturaleza de los datos de los que disponemos, nos puede interesar conseguir fórmulas de descomposición y reconstrucción que involucren a valores de una cierta función  $F$  sobre las mallas  $X^k$ , variando  $k$ . La multirresolución mediante valores puntuales constituye el ejemplo más sencillo de multirresolución discreta. Basta con definir

$$F_j^k = F(x_j^k) \quad (3)$$

y, en base a (2), se sigue la siguiente **fórmula de descomposición**:

$$F_j^{k-1} = F_{2j}^k; \quad (j = 0, \dots, 2^{k-1}).$$

Para reconstruir valores ausentes de  $F$  sobre la malla  $X^k$  a partir de los valores sobre la malla  $X^{k-1}$  se puede emplear con los valores  $F^{k-1}$  una sencilla técnica de interpolación –pongamos lineal–. Si denotamos al interpolante  $I(x; F^{k-1})$ , entonces se obtienen aproximaciones de los valores ausentes en la malla  $X^k$ :

$$\tilde{F}_{2j-1}^k := I(x_{2j-1}^k; F^{k-1}). \quad (4)$$

En principio, cualquier técnica de interpolación es susceptible de ser utilizada. Suele emplearse interpolación central polinomial. En ese caso el interpolante es un polinomio de grado  $2p - 1$ , únivocamente determinado por los  $2p$  datos  $(F_{j-p}^{k-1}, \dots, F_{j+p-1}^{k-1})$ . Nótese que, en ese caso, el orden de la aproximación es  $r = 2p - 1$ . Así,

$$I(x_{2j-1}^k; F^{k-1}) = \sum_{m=1}^p \Lambda_m(F_{j+m-1}^{k-1} + F_{j-m}^{k-1}), \quad (5)$$

donde el cálculo de  $\Lambda_m$  se realiza teniendo en cuenta la fórmula del polinomio interpolador de Lagrange. Como consecuencia, si  $F$  es ya un polinomio de grado menor o igual que  $2p - 1$ , entonces la reconstrucción reproduce la propia función  $F$  y los errores de interpolación son nulos. En caso contrario, en las componentes impares de la malla sí se producen errores de interpolación. Estos errores juegan el papel de coeficientes de detalle entre dos representaciones a resoluciones consecutivas. Si se denota

$$D_j^{k-1} = F_{2j-1}^k - \tilde{F}_{2j-1}^k; \quad (j = 0, \dots, 2^{k-1}),$$

entonces  $\{F^k, D^k\}$  contiene la misma información que  $F^{k+1}$ . La **fórmula de reconstrucción** en este caso viene dada, para  $j = 0, \dots, 2^{k-1}$ , mediante:

$$\begin{aligned} F_{2j}^k &= F_j^{k-1}, \\ F_{2j-1}^k &= \sum_{m=1}^p \Lambda_m (F_{j+m-1}^{k-1} + F_{j-m}^{k-1}) + D_j^{k-1}. \end{aligned}$$

Las fórmulas de descomposición y reconstrucción (2.1) y (6) concretan el análisis multirresolución para valores puntuales de una función  $F$ . Sin embargo, y volviendo al inicio del artículo, a la vista de la gráfica de la Figura 3, no son precisamente los valores puntuales de la función escalonada lo que podría interesar reconstruir o descomponer. Más bien interesa relacionar la altura de cada escalón a resoluciones distintas. Puesto que la base de cada escalón tiene de longitud igual a la unidad, la altura coincide con el área de cada trozo constante. Interesa, pues, conseguir fórmulas de descomposición y reconstrucción que relacionen áreas o, por el Teorema del valor medio, que relacionen valores medios. Más precisamente, el valor medio que encierra una función  $f(x)$  sobre el intervalo  $(x_{j-1}^k, x_j^k)$  viene dado por

$$f_j^k := \frac{1}{h_k} \int_{x_{j-1}^k}^{x_j^k} f(x) dx.$$

Teniendo en cuenta (2) y el hecho de que la integral sobre un intervalo se puede descomponer como suma de integrales sobre una partición del mismo, se deduce fácilmente una **fórmula de descomposición** en el contexto de áreas:

$$\begin{aligned} f_j^{k-1} &= \frac{1}{h_{k-1}} \int_{x_{j-1}^{k-1}}^{x_j^{k-1}} f(x) dx \\ &= \frac{1}{2h_k} \left( \int_{x_{2j-2}^k}^{x_{2j-1}^k} f(x) dx + \int_{x_{2j-1}^k}^{x_{2j}^k} f(x) dx \right) \\ &= \frac{1}{2} (f_{2j-1}^k + f_{2j}^k). \end{aligned}$$

Así,

$$f_j^{k-1} = \frac{1}{2} (f_{2j-1}^k + f_{2j}^k); \quad (j = 1, \dots, 2^{k-1}) \tag{6}$$

es la fórmula de descomposición que sirve de base para la escritura del algoritmo de descomposición. Respecto a la fórmula de reconstrucción, se hace necesario recurrir a uno de los resultados más importantes del análisis infinitesimal, el *punteo* entre derivadas e integrales. Más precisamente, si  $a, b \in \mathbb{R}, a < b$  y queremos calcular la integral

$$\int_a^b f(x) dx,$$

el teorema fundamental del cálculo establece que es suficiente con calcular una función  $F$  tal que  $F'(x) = f(x)$ , para cada número  $x$  entre  $a$  y  $b$ ; entonces<sup>3</sup>

$$\int_a^b f(x) dx = F(b) - F(a).$$

Esta última igualdad permite ver un área como diferencia de evaluaciones puntuales de una función  $F$ . Justamente es la clave para relacionar el análisis multirresolución mediante cálculo de áreas con el análisis multirresolución mediante valores puntuales. De este modo, reconstruir valores medios de  $f$  equivale –salvo constante multiplicativa– a hacer la resta entre las reconstrucciones de los valores puntuales de  $F$ . Concretamente, si  $F(x) = \int_0^x f(t) dt$  y se sigue la notación introducida en (3), entonces

$$\begin{aligned} F_j^{k-1} &= \int_0^{x_j^{k-1}} f(t) dt = \sum_{i=1}^j \int_{x_{i-1}^{k-1}}^{x_i^{k-1}} f(t) dt \\ &= h_{k-1} \sum_{i=1}^j f_i^{k-1}. \end{aligned}$$

De la expresión anterior, se obtiene que  $F_j^{k-1} - F_{j-1}^{k-1} = h_{k-1} f_j^{k-1}$ , ya que en la suma se van cancelando todos los términos salvo  $f_j^{k-1}$ . Por tanto,

$$f_j^{k-1} = \frac{1}{h_{k-1}} (F_j^{k-1} - F_{j-1}^{k-1}). \quad (7)$$

Introducimos un interpolante para reconstruir la función primitiva de la manera introducida en (4). Recordemos también que  $F_{j-1}^{k-1} = F_{2j-2}^k$ . De esta manera, la expresión (7) sirve para encontrar una reconstrucción de los valores impares de

---

<sup>3</sup>No es cuestión, en un artículo como éste, de ir más allá de unas breves notas introduciendo exigencias sobre la función  $f$  y detallando métodos del cálculo infinitesimal. Sin embargo, sí queremos resaltar que la manera en que ese conjunto de resultados fueron descubiertos y estudiados constituye una aventura intelectual que tardó casi dos mil años en dar rigor a los conceptos de derivada, integral o límite y llevarlos hasta la forma en la que los conocemos hoy. Algunas de estas ideas, como por ejemplo considerar un área como una colección (infinita) de segmentos o un volumen como una colección (infinita) de áreas, fueron razonamientos que Arquímedes, allá por el siglo III a.C., incluyó en su libro *El Método*, saltándose de camino la prohibición aristotélica del uso del infinito en acto. Sorprendentemente esta misma idea no apareció en las Matemáticas hasta el siglo XVII, dos mil años después de Arquímedes.



$f$  a resolución  $k$ . Esta reconstrucción la denotaremos mediante  $\tilde{f}_j^k$  y viene dada por

$$\tilde{f}_{2j-1}^k = \frac{1}{h_{k-1}} (I(x_{2j-1}^k; F^{k-1}) - F_{j-1}^{k-1}). \quad (8)$$

Análogamente se obtiene también que

$$\tilde{f}_{2j}^k = \frac{1}{h_{k-1}} (F_j^{k-1} - I(x_{2j-1}^k; F^{k-1})). \quad (9)$$

De (8) y (9) se sigue que

$$\frac{1}{2} (\tilde{f}_{2j-1}^k + \tilde{f}_{2j}^k) = \frac{1}{h_{k-1}} (F_j^{k-1} - F_{j-1}^{k-1}).$$

Teniendo en cuenta (7), y rehaciendo un poco los términos, tenemos que

$$\tilde{f}_{2j}^k = 2f_j^{k-1} - \tilde{f}_{2j-1}^k. \quad (10)$$

Por tanto, la reconstrucción de los valores pares se hace mediante (10) en lugar de emplear (9). Es posible dar una expresión equivalente a (8) en términos únicamente de los valores medios de  $f$  (no de su primitiva). Básicamente, utilizando (5), y simplificando algunos términos, se tiene que

$$\tilde{f}_{2j-1}^k = f_j^{k-1} + \sum_{m=1}^{p-1} \lambda_m (f_{j+m}^{k-1} - f_{j-m}^{k-1}), \quad (11)$$

donde  $\lambda_m$  y  $\Lambda_m$  están relacionados según la fórmula de recurrencia

$$\begin{aligned} \lambda_0 &= 1, \\ \lambda_m &= \lambda_{m-1} - 2\Lambda_m; \quad (m \geq 1). \end{aligned}$$

En definitiva, el **algoritmo de reconstrucción** se programa en base a (10) y (11). En este caso, los coeficientes de detalle están dados por

$$\begin{aligned} d_j^{k-1} &= f_{2j-1}^k - \tilde{f}_{2j-1}^k \\ &= f_{2j-1}^k - f_j^{k-1} - \sum_{m=1}^{p-1} \lambda_m (f_{j+m}^{k-1} - f_{j-m}^{k-1}); \quad (j = 1, \dots, 2^{k-1}). \end{aligned}$$

Los algoritmos son estables en el sentido de que una pequeña variación en los datos iniciales (a descomponer o a reconstruir) produce una variación controlable mediante constantes en las sucesivas descomposiciones o reconstrucciones. Es decir, supongamos que  $f^0$  es un conjunto inicial de datos que sufren una perturbación  $\hat{f}^0$ . Si  $\{f^k, d^k, d^{k-1}, \dots, d^1\}$  y  $\{\hat{f}^k, \hat{d}^k, \hat{d}^{k-1}, \dots, \hat{d}^1\}$  son sus correspondientes representaciones multiescala a resolución  $k$ , entonces existen unas constantes  $C_1$  y  $C_2$  tales que

$$\|f^k - \hat{f}^k\|_{\mathbb{Z}} \leq C_1 \|f^0 - \hat{f}^0\|_{\mathbb{Z}} + C_2 \sum_{l=1}^k \|d^l - \hat{d}^l\|_{\mathbb{Z}},$$

donde, en general, para cualquier sucesión  $(u_n)_{n \in \mathbb{Z}}$  se define  $\|u\|_{\mathbb{Z}} = \sup_{n \in \mathbb{Z}} |u_n|$ .

La estabilidad de los distintos algoritmos multirresolución (lineales y no lineales) constituye una línea de investigación muy activa. En [3] se ha logrado, entre otras cosas, sintetizar y ajustar más aún las constantes de estabilidad presentes en una multitud de resultados conocidos. Para ello se empleó una formulación más general de uno de los esquemas de reconstrucción no lineales más utilizados a día de hoy. Como pequeño homenaje a Arquímedes, cerramos esta sección con una imagen de una escultura suya.

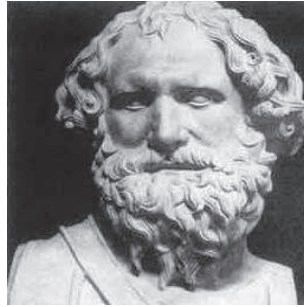


Figura 3: Arquímedes de Siracusa

### 2.1.1 Aplicaciones al análisis de la regularidad y a la compresión de datos

En las proximidades de las discontinuidades de la función  $f$ , los coeficientes de detalle  $d_j^k$  decaen. La velocidad de decaimiento está relacionada con la regularidad de  $f$  y el orden de precisión  $r$  que tenga el interpolante utilizado en la reconstrucción. Más precisamente, la función  $f(x)$  en  $x = x^*$  tiene  $q - 1$  derivadas continuas y una discontinuidad de salto en la derivada  $q$ -ésima de magnitud  $[f^{(q)}]$ , entonces en los puntos  $x_j^k$  cercanos a  $x^*$  se tiene que

$$d_j^k \sim \begin{cases} h_k^q [f^{(q)}], & \text{si } 0 \leq q < r \\ h_k^q f^{(q)}, & \text{si } q \geq r. \end{cases}$$

Como consecuencia de lo anterior se derivan, además, las siguientes conclusiones:

- (i) Lejos de las discontinuidades, los coeficientes  $d_j^k$  decrecen a medida que se va a niveles más finos de resolución.
- (ii) En un entorno de una irregularidad de  $f(x)$ , los coeficientes  $d_j^k$  son de orden  $\mathcal{O}([f])$ .

De este modo, es posible estudiar la regularidad local de una función  $f$  a través de la multirresolución de las medias. Otra de las aplicaciones de este tipo

de algoritmos consiste en utilizarlos para comprimir datos. La técnica que suele emplearse es una técnica de truncamiento de los coeficientes de detalle  $d_j^k$ . Para ello se redefinen como nulos aquellos coeficientes que están por debajo de una tolerancia prefijada  $\varepsilon_k > 0$ . Más precisamente, los nuevos coeficientes de detalle se definen mediante

$$\hat{d}_j^k = \begin{cases} 0, & \text{si } |d_j^k| \leq \varepsilon_k \\ d_j^k, & \text{en caso contrario.} \end{cases}$$

La reconstrucción utilizando los coeficientes truncados  $\hat{d}_j^k$  es estable -en base a los resultados de estabilidad-. Por tanto, la compresión que se obtenga tras una sucesión de reconstrucciones “no se aleja” de la reconstrucción que se obtiene sin truncamiento. En la práctica, se consigue eliminar el ruido presente en los datos reconstruyendo con detalles truncados. Realizamos tres descomposiciones sucesivas de la imagen original de la Figura 3 ( que corresponde a una matriz de tamaño  $256 \times 256$ ).



Figura 4: Izquierda: Primera descomposición de la imagen de la Figura 3. Derecha: Suma de los detalles al primer nivel

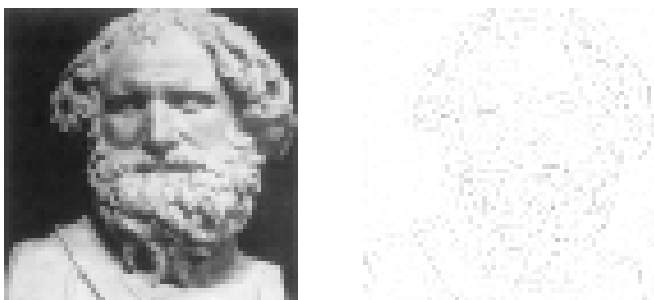


Figura 5: Izquierda: Segunda descomposición de la imagen de la Figura 3. Derecha: Suma de los detalles al segundo nivel

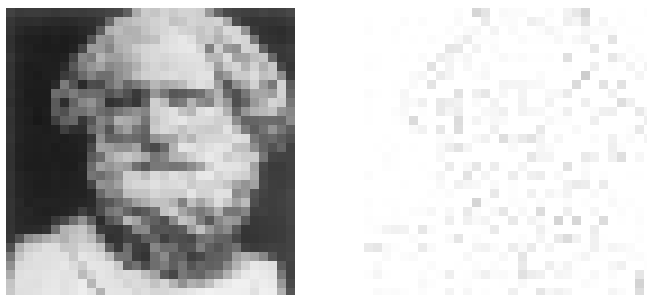


Figura 6: Izquierda: Tercera descomposición de la imagen de la Figura 3. Derecha: Suma de los detalles al tercer nivel

Finalmente, la Figura 7 (izquierda) muestra la versión comprimida de la imagen de la Figura 3. La reconstrucción se obtiene mediante interpolación cúbica con la imagen de la Figura 4 y todos los detalles igual a cero. La Figura 7 (derecha) muestra la reconstrucción obtenida a partir de la imagen original.

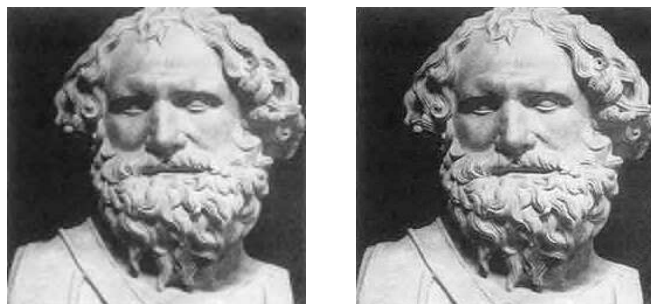


Figura 7: Izquierda: Reconstrucción de tamaño  $256 \times 256$  a partir de la primera descomposición con detalles nulos. Derecha: Reconstrucción de tamaño  $512 \times 512$  a partir de la imagen original de la Figura 3.

La reconstrucción de la imagen de partida produce un aumento de resolución que permite disponer de una imagen más nítida que la original.

## Referencias

- [1] D. Donoho, Interpolating wavelet transforms. Preprint, Department of Statistics, Stanford University, 1992.
- [2] G. Deslaurier and S. Dubuc, Symmetric iterative interpolation processes, *Constr. Approx.* (5), 49–68 (1989).

- [3] A. Guessab, M. Moncayo and G. Schmeisser, A class of nonlinear four-point subdivision schemes Properties in terms of conditions. *Adv. Comput. Math.* (September 2011), DOI: 10.1007/s10444-011-9199-y.
- [4] I. Daubechies, Ten lectures on wavelets. SIAM, Philadelphia, PA, 1992.
- [5] A. Harten, Discrete multiresolution analysis and generalized wavelets. *J. Appl. Numer. Math.*, (12), 153–192 (1993).
- [6] A. Harten, Multiresolution representation of data: General framework. *SIAM J. Numer. Anal.*, (33), 1205–1256 (1996).
- [7] S. Mallat, *A wavelet tour of signal processing*. Academic Press, 1999.
- [8] W. Sweldens, The lifting scheme: A construction of second generation wavelets. *SIAM J. Numer. Anal.*, (29) 511-546 (1998).