

Tema 7. Eliminación del ruido en imágenes mediante filtros no lineales y filtros no locales

Gonzalo Galiano Casas

Departamento de Matemáticas
Universidad de Oviedo
galiano@orion.ciencias.uniovi.es

Introducción

En este tema introduciremos algunas técnicas matemáticas para el filtrado de ruidos en imágenes (*denoising*). Esta tarea es una de las fundamentales en el preprocesamiento de imágenes, habiendo dado lugar a una gran variedad de algoritmos provenientes de diferentes disciplinas matemáticas: teoría de Fourier y Wavelets, optimización y métodos varacionales, ecuaciones en derivadas parciales, probabilidad y procesos estocásticos, y otras.

Una imagen puede interpretarse como una función, u , que está definida en el espacio de los píxeles $\Omega \subset \mathbb{R}^2$, y toma valores en cierto subconjunto de \mathbb{R} (imagen de intensidades o grises) o de \mathbb{R}^3 (imagen vectorial o en color). En lo que sigue, nos centraremos en las imágenes de intensidad, aunque la mayor parte de nuestra discusión se extiende a imágenes vectoriales y, de hecho, a *imágenes* definidas en un sentido más amplio, tanto por el espacio de salida, por ejemplo, los vídeos (\mathbb{R}^4), como de llegada, por ejemplo, los volúmenes de resonancias magnéticas de difusión DTI (\mathbb{R}^6).

Partimos de la situación en la que nuestra imagen contiene ruido. Las fuentes de ruido son variadas. En fotografía, puede deberse a una baja iluminación o a una incorrecta exposición. En imágenes médicas, suele ser debido a la interferencia del captador (escáner) con campos magnéticos de baja intensidad generados por el propio escáner, o por aparatos electrónicos cercanos al mismo. En imágenes de satélite, la heterogeneidad atmosférica produce ruido.

Matemáticamente, el ruido en una imagen se relaciona con su irregularidad, con la existencia de componentes frecuenciales de alta frecuencia, en términos de la teoría de Fourier. En la Figura 1 vemos la gráfica de una señal unidimensional (sección de una imagen) limpia y de la misma señal con un ruido añadido. Una opción para recuperar la señal limpia a partir de la ruidosa será suavizar la última, es decir, eliminar las pequeñas variaciones (gradientes) de la señal ruidosa. Explotaremos esta sencilla idea mediante dos tipos de métodos matemáticos:

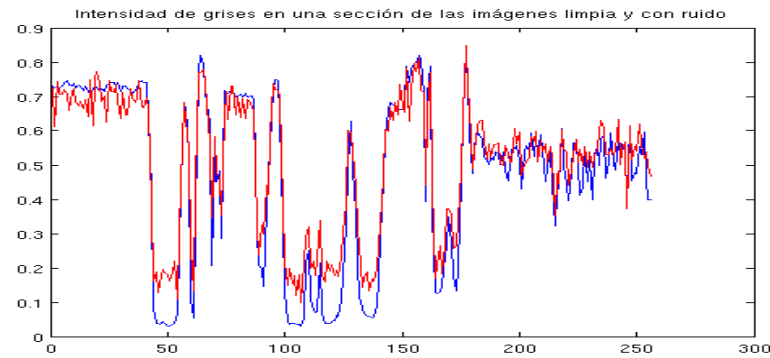


Figura 1: El ruido introduce *irregularidad* en una señal.

- Método variacional. La imagen restaurada se obtiene mediante un proceso que involucra la minimización del gradiente de la imagen ruidosa. Dependiendo de la medida que se utilice para evaluar las fluctuaciones del gradiente, el filtro resultante puede ser lineal (norma $H^1(\Omega)$) o no lineal (otras normas, por ejemplo, la *variación total*).

- Métodos basados en entornos. En este caso, en cada píxel, la imagen restaurada se obtiene como un promedio de la imagen original en un entorno de dicho píxel. La peculiaridad de estos métodos es que el entorno se construye en el espacio conjunto de los píxeles y de los valores de intensidad de la imagen. De ahí el otro nombre por el que se conoce a estos métodos: *no locales*.

Métodos variacionales

Introducción

Las variaciones de una imagen, u_0 , a lo largo y ancho del dominio de píxeles que, de momento, suponemos continuo, vienen dadas por su gradiente

$$\nabla u_0(\mathbf{x}) = (\partial_x u_0(\mathbf{x}), \partial_y u_0(\mathbf{x})), \quad \mathbf{x} = (x, y).$$

El tamaño de dichas variaciones puede capturarse usando el módulo del gradiente, esto es

$$|\nabla u_0(\mathbf{x})| = \left((\partial_x u_0(\mathbf{x}))^2 + (\partial_y u_0(\mathbf{x}))^2 \right)^{1/2}.$$

De modo que si lo que queremos es eliminar las variaciones de u_0 , podemos proponernos minimizar el módulo de su gradiente, es decir, hallar otra imagen v tal que $|\nabla v|$ sea mínimo (en algún sentido a precisar) y que, al mismo tiempo, se parezca lo más posible a la imagen original, u_0 (de otro modo tomaríamos $v(\mathbf{x}) = \text{const.}$, que no tendría mucho sentido para aproximar a u_0).

Así llegamos a una *formulación variacional* del problema de filtrado de ruidos: Hallar $v : \Omega \rightarrow \mathbb{R}$ que realice el mínimo de

$$J(v) = \text{medida de } (|\nabla v|) + \lambda \text{ medida de } (u_0 - v).$$

Al primer término de la derecha se le denomina *término de regularidad*, y al segundo *término de fidelidad*. El parámetro $\lambda > 0$ nos da una idea del peso relativo entre la regularización o suavizado de la imagen y la cercanía con la imagen inicial.

A J se le denomina *funcional*. Es un caso particular de función que aplica un espacio de funciones sobre otro espacio de funciones. En nuestro caso, transforma imágenes en imágenes.

La riqueza de este método reside en la variedad de *medidas* que podemos utilizar para determinar J . El ejemplo más notable es el siguiente

$$J(v) = \frac{1}{2} \int_{\Omega} |\nabla v(\mathbf{x})|^2 d\mathbf{x} + \frac{\lambda}{2} \int_{\Omega} |u_0 - v|^2 d\mathbf{x}. \quad (1)$$

Obviamente, siempre tendremos $J(v) \geq 0$. Para que $J(v)$ no pueda ser infinito, basta con exigir, por una parte,

$$\int_{\Omega} |u_0|^2 d\mathbf{x} < \infty, \quad (2)$$

y por otra

$$\int_{\Omega} |\nabla v(\mathbf{x})|^2 d\mathbf{x} < \infty, \quad \int_{\Omega} |v|^2 d\mathbf{x} < \infty. \quad (3)$$

Al espacio de funciones que satisface (2) se le denota por $L^2(\Omega)$, y al que satisface (3), por $H^1(\Omega)$. Es decir, $H^1(\Omega)$ es el espacio de funciones que tienen tanto a la propia función como al módulo de su gradiente en $L^2(\Omega)$. Por tanto, $H^1(\Omega) \subset L^2(\Omega)$.

Como en la teoría de funciones, para calcular un mínimo del funcional J podemos inspeccionar sus puntos críticos, es decir, los puntos (funciones) para los que se anula la derivada de J . Derivar funcionales es algo más delicado que derivar funciones, pero la idea es la misma: medir pequeñas variaciones del funcional cuando nos desviamos del punto de referencia. Esto lo hacemos, formalmente, mediante la derivada direccional

$$\langle J'(v), \varphi \rangle = \lim_{h \rightarrow 0} \frac{J(v + h\varphi) - J(v)}{h},$$

donde φ es una función que pertenece al mismo espacio que v , en nuestro ejemplo $\varphi \in H^1(\Omega)$. Aplicado

a (1), obtenemos

$$\begin{aligned}
\langle J'(v), \varphi \rangle &= \lim_{h \rightarrow 0} \frac{1}{2h} \left(\int_{\Omega} (|\nabla(v + h\varphi)|^2 - |\nabla v|^2) d\mathbf{x} + \lambda \int_{\Omega} (|u_0 - v + h\varphi|^2 - |u_0 - v|^2) d\mathbf{x} \right) \\
&= \lim_{h \rightarrow 0} \frac{1}{2h} \left(\int_{\Omega} (2h \nabla v \cdot \nabla \varphi + h^2 |\nabla \varphi|^2) d\mathbf{x} + \lambda \int_{\Omega} (2h(u_0 - v)\varphi + h^2 |\varphi|^2) d\mathbf{x} \right) \\
&= \int_{\Omega} \nabla v \cdot \nabla \varphi d\mathbf{x} + \lambda \int_{\Omega} (u_0 - v)\varphi d\mathbf{x}.
\end{aligned}$$

De modo que v será un punto crítico de J si $J'(v)$ es nulo en cualquier dirección, φ , que tomemos, es decir, si

$$\int_{\Omega} \nabla v \cdot \nabla \varphi d\mathbf{x} + \lambda \int_{\Omega} (u_0 - v)\varphi d\mathbf{x} = 0, \quad \forall \varphi \in H^1(\Omega).$$

Usando en esta expresión la regla de integración por partes, obtenemos que debe satisfacerse

$$-\int_{\Omega} \Delta v \varphi d\mathbf{x} + \int_{\partial\Omega} \varphi \nabla v \cdot \mathbf{n} d\mathbf{x} + \lambda \int_{\Omega} (u_0 - v)\varphi d\mathbf{x} = 0, \quad \forall \varphi \in H^1(\Omega). \quad (4)$$

Aquí hemos introducido alguna notación: \mathbf{n} es la normal a la frontera, $\partial\Omega$, del dominio de los píxeles, Ω , y

$$\Delta v(\mathbf{x}) = \partial_{xx}^2 v(\mathbf{x}) + \partial_{yy}^2 v(\mathbf{x})$$

es el *laplaciano* de v . Suponiendo que nuestro espacio de funciones incorpora la condición $\nabla v \cdot \mathbf{n} = 0$ y reordenando (4) obtenemos

$$\int_{\Omega} (-\Delta v + \lambda(u_0 - v))\varphi d\mathbf{x} = 0, \quad \forall \varphi \in H^1(\Omega).$$

Puesto que esta integral es nula para toda φ , deducimos (Lema fundamental del cálculo variacional) que nuestro punto crítico debe satisfacer

1. $v \in H^1(\Omega)$
2. $\nabla v(\mathbf{x}) \cdot \mathbf{n}(\mathbf{x}) = 0$ para $\mathbf{x} \in \partial\Omega$, y
3. $-\Delta v(\mathbf{x}) + \lambda(u_0(\mathbf{x}) - v(\mathbf{x})) = 0$ para $\mathbf{x} \in \Omega$.

El problema 1)-3) es un problema de ecuaciones en derivadas parciales muy estudiado en otros ámbitos, como la teoría del calor (también con importantes contribuciones de Fourier). Un método muy utilizado en la práctica para resolver este problema es el llamado método del gradiente o de descenso, el cual añadiendo un tiempo artificial t , plantea la siguiente ecuación

$$\partial_t u(t, \mathbf{x}) - \Delta u(t, \mathbf{x}) + \lambda(u_0(\mathbf{x}) - u(t, \mathbf{x})) = 0, \quad \text{para } t > 0, \mathbf{x} \in \Omega, \quad \text{y con } u(0, \mathbf{x}) = u_0(\mathbf{x}). \quad (5)$$

Puede demostrarse que, cuando $t \rightarrow \infty$, se tiene $u(t, \mathbf{x}) \rightarrow v(\mathbf{x})$, donde $v(\mathbf{x})$ es la solución del problema 1)-3). La ventaja de este enfoque es que la solución de (5) puede aproximarse fácilmente. Por ejemplo, discretizando la derivada temporal, obtenemos

$$u(t + \tau, \mathbf{x}) \approx u(t, \mathbf{x}) + \tau(\Delta u(t, \mathbf{x}) + \lambda(u_0(\mathbf{x}) - u(t, \mathbf{x}))),$$

de modo que llamando $u_n(\mathbf{x}) = u(n\tau, \mathbf{x})$ producimos la sucesión

$$\begin{aligned} u_0(\mathbf{x}) &= u_0(\mathbf{x}), \\ u_{n+1}(\mathbf{x}) &= u_n(\mathbf{x}) + \tau(\Delta u_n(\mathbf{x}) + \lambda(u_0(\mathbf{x}) - u_n(\mathbf{x}))), \quad n = 0, 1, \dots \end{aligned} \quad (6)$$

que nos aproxima la solución $v(\mathbf{x})$ que buscamos.

Resumiendo, hemos dado los siguientes pasos:

1. Hemos definido un funcional, J , que mide la regularidad de la imagen restaurada, v , y su similitud con la imagen inicial, u_0 .
2. Hemos planteado el problema de ecuaciones en derivadas parciales que satisfacen los puntos críticos de J .
3. Hemos introducido un algoritmo de descenso para aproximar dichos puntos críticos por una sucesión de imágenes que, esperamos, atenúen el ruido de la imagen inicial.

Hay algunas cuestiones que no hemos tratado. Por ejemplo,

1. Como el funcional J es no negativo (y continuo), siempre existe, al menos, un mínimo. Ahora, ¿es posible que existan más de uno? Es decir, ¿puede este procedimiento conducirnos a distintas restauraciones de la imagen inicial?
2. ¿Cómo de satisfactoria es la imagen restaurada respecto a los detalles que queremos conservar de la imagen inicial?
3. Suponiendo que el procedimiento sea satisfactorio, ¿cuándo detener el cálculo de la sucesión de imágenes restauradas, u_n , en la práctica? Es decir, ¿con qué u_N quedarnos, para un N fijo?

Estas preguntas no tienen, en general, una respuesta sencilla. Forman parte de la investigación matemática en la teoría del procesamiento de imágenes.

La primera cuestión tiene una respuesta negativa cuando el funcional J es convexo, lo cual asegura la unicidad de la solución v del problema 1)-3). Sin embargo, para funcionales no convexos, esta afirmación no es, en general, válida.

Para responder a la segunda cuestión no existe una metodología general. Así, lo común es utilizar técnicas diversas para el análisis de la eficiencia del método. Estas técnicas van desde las de tipo subjetivo, como el análisis visual directo del resultado, es decir, la percepción que tenemos de *mejora*, hasta técnicas objetivas basadas en el cálculo de la diferencia entre la imagen inicial sin ruido y el resultado obtenido. Por ejemplo, mediante el error cuadrático medio (*mean square error*-MSE), o el *peak signal to noise ratio* PSNR, dados por

$$MSE = \|u_n - u_0\|_{L^2(\Omega)}, \quad PSNR = 20 \log_{10} \left(\frac{\max(u_0)}{\sqrt{MSE}} \right).$$

Finalmente, la tercera pregunta se refiere al *criterio de parada* en el cálculo de la sucesión u_n . Nuevamente, no existe un criterio universal. Lo más habitual es manejar varios criterios simultáneamente, por ejemplo, detener la iteración cuando el error relativo entre dos iteraciones es menor que cierto umbral, *tol*, es decir, cuando se satisface

$$\frac{\|u_n - u_{n-1}\|_{L^2(\Omega)}}{\|u_{n-1}\|_{L^2(\Omega)}} < tol,$$

al mismo tiempo que se limita el número máximo de iteraciones, para evitar bucles infinitos en caso de que no se satisfaga el criterio del error.

Minimización de la variación total

Como puede apreciarse en la Figura 2, la minimización del funcional J de la subsección anterior, con el término de regularidad

$$\int_{\Omega} |\nabla v|^2 d\mathbf{x},$$

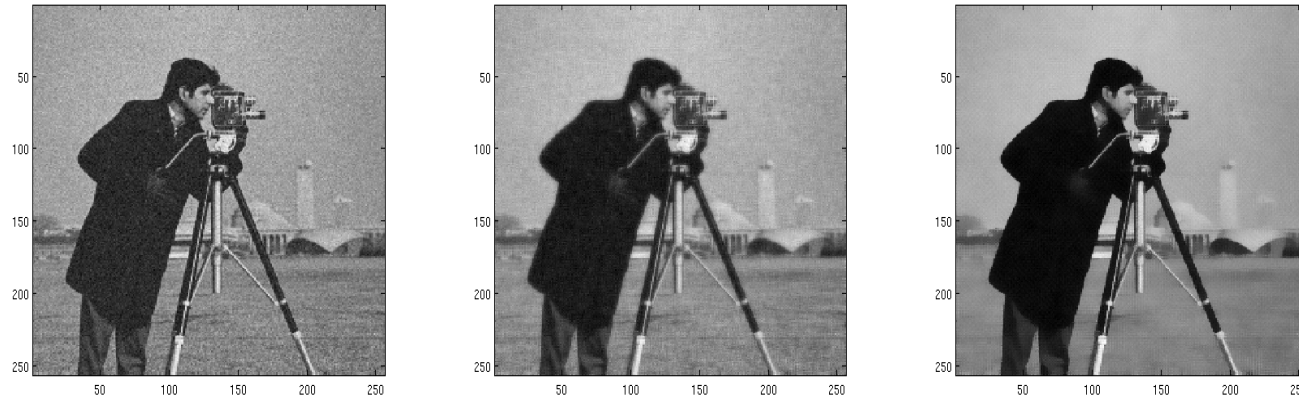


Figura 2: De izquierda a derecha: imagen con ruido, imagen restaurada minimizando en $H^1(\Omega)$ (laplaciano), e imagen restaurada minimizando la variación total.

elimina ruidos, pero produce una difusión no deseada de los bordes de los objetos contenidos en la imagen. Este efecto no es, en general, deseable dado que una de las aplicaciones básicas del procesamiento de imágenes es la identificación de objetos dentro de las imágenes (segmentación), para lo que necesitamos que dichos bordes estén claramente delimitados.

Dada la flexibilidad de la aproximación variacional al problema de la restauración, podemos considerar otros términos de regularización más convenientes. Entre los más utilizados se encuentra la *variación total*, que podemos definir informalmente como

$$\int_{\Omega} |\nabla v| d\mathbf{x}.$$

Aunque el cambio parezca pequeño, este término dificulta enormemente el análisis matemático del problema, y también introduce ciertos inconvenientes en su aproximación numérica. Sin embargo, los

resultados merecen la pena.

Siguiendo el progrma introducido en la subsección anterior, comenzamos definiendo el nuevo funcional:

$$J(v) = \int_{\Omega} |\nabla v(\mathbf{x})| d\mathbf{x} + \frac{\lambda}{2} \int_{\Omega} |u_0 - v|^2 d\mathbf{x}.$$

Seguimos teniendo $J(v) \geq 0$. Ahora, para que $J(v)$ sea finito, exigimos

$$\int_{\Omega} |u|^2 d\mathbf{x} < \infty,$$

y

$$\int_{\Omega} |\nabla v(\mathbf{x})| d\mathbf{x} < \infty, \quad \int_{\Omega} |v|^2 d\mathbf{x} < \infty.$$

A continuación, calculamos la derivada direccional de J (queda como ejercicio)

$$\langle J'(v), \phi \rangle = \int_{\Omega} \frac{\nabla v}{|\nabla v|} \cdot \nabla \phi d\mathbf{x} + \lambda \int_{\Omega} (u_0 - v) \phi d\mathbf{x}.$$

Obviamente, esta expresión es problemática si $|\nabla v| = 0$. En cualquier caso, seguimos nuestra discusión anterior para llegar a la ecuación

$$-\operatorname{div} \left(\frac{\nabla v}{|\nabla v|} \right) + \lambda(u_0 - v) = 0,$$

a partir de la cual, tras considerar el método del descenso, llegamos al problema

$$\partial_t u(t, \mathbf{x}) - \operatorname{div} \left(\frac{\nabla u(t, \mathbf{x})}{|\nabla u(t, \mathbf{x})|} \right) + \lambda(u_0(\mathbf{x}) - u(t, \mathbf{x})) = 0, \quad \text{para } t > 0, \mathbf{x} \in \Omega, \quad \text{y con } u(0, \mathbf{x}) = u_0(\mathbf{x}). \quad (7)$$

En la práctica, la división por el gradiente da lugar a *overflows*, por lo que se introduce una nueva aproximación que da lugar a la sucesión $u_n(\mathbf{x}) = u(n\tau, \mathbf{x})$, con

$$\begin{aligned} u_0(\mathbf{x}) &= u(\mathbf{x}), \\ u_{n+1}(\mathbf{x}) &= u_n(\mathbf{x}) + \tau \operatorname{div} \left(\frac{\nabla u(t, \mathbf{x})}{\sqrt{|\nabla u(t, \mathbf{x})|^2 + \varepsilon^2}} \right) + \tau \lambda(u_0(\mathbf{x}) - u_n(\mathbf{x})), \quad n = 0, 1, \dots \end{aligned} \quad (8)$$

para cierto $\varepsilon > 0$ fijado, la cual nos aproxima la solución $v(\mathbf{x})$ que buscamos.

Discretización espacial

Las iteraciones (6) y (8) surgen de una discretización temporal de sus correspondientes versiones continuas (5) y (7). Sin embargo, para el cálculo computacional de estos filtros, aún debemos abordar la aproximación espacial de los operadores diferenciales que aparecen en estas ecuaciones.

Debido al carácter simple del dominio de definición de las imágenes, un rectángulo, y al equiespaciamento natural entre píxeles, cuya distancia puede normalizarse a uno, lo más común es discretizar dichos operadores mediante el método de diferencias finitas.

Consideremos, para empezar, una función de variable real, $f : [a, b] \subset \mathbb{R} \rightarrow \mathbb{R}$, y una malla equiespaciada de $[a, b]$, con $N + 1$ nodos: $x_j = a + j \frac{(b-a)}{N}$, para $j = 0, \dots, N$. Denotemos $h = \frac{(b-a)}{N}$. Existen

tres maneras fundamentales de aproximar la derivada de $f(x_j)$,

$$f'(x_j) \approx \frac{f(x_{j+1}) - f(x_j)}{h} \quad (\text{fórmula progresiva}),$$

$$f'(x_j) \approx \frac{f(x_j) - f(x_{j-1}))}{h} \quad (\text{fórmula regresiva}),$$

$$f'(x_j) \approx \frac{f(x_{j+1}) - f(x_{j-1}))}{2h} \quad (\text{fórmula centrada}).$$

La fórmula centrada es la que tiene un mejor orden de aproximación al resultado exacto. Componiendo las fórmulas progresiva y regresiva obtenemos una aproximación centrada para la segunda derivada:

$$f''(x_j) \approx \frac{f(x_{j+1}) - 2f(x_j) + f(x_{j-1}))}{h^2}.$$

Todas estas fórmulas tienen el llamado *problema del borde* debido a que no pueden ser evaluadas en uno o ambos bordes del intervalo, es decir, para $j = 0$ y $j = N$. Este problema puede resolverse usando otras fórmulas, de tipo interpolatorio, para aproximar las derivadas en los bordes usando únicamente nodos pertenecientes al intervalo. Sin embargo, en nuestro caso, las condiciones de frontera nos fijarán estos valores sin tener que recurrir a cálculos adicionales.

Volviendo a nuestro problema, podemos extender las ideas en una dimensión a varias dimensiones y obtener la discretización

$$\Delta u(i, j) = \partial_{xx}u(i, j) + \partial_{yy}u(i, j) \approx u(i+1, j) + u(i-1, j) + u(i, j+1) + u(i, j-1) - 4u(i, j), \quad (9)$$

donde $u(i, j)$ denota el valor de u en el pixel de coordenadas (i, j) , y hemos normalizado a uno la distancia entre píxeles consecutivos, es decir, $h = 1$. Un pseudocódigo para implementar el filtro de minimización de la norma $H^1(\Omega)$, problema (6), viene dado en Pseudocódigo 1.

Para discretizar el operador de variación total o, en general, una expresión del tipo $\text{div}(b(\mathbf{x})\nabla u(\mathbf{x}))$ podemos utilizar la concatenación de la fórmula centrada para obtener

$$\begin{aligned} \text{div}(b(i, j)\nabla u(i, j)) \approx \frac{1}{4} \Big(& b(i+1, j)u(i+2, j) + b(i-1, j)u(i-2, j) + b(i, j+1)u(i, j+2) \\ & + b(i, j-1)u(i, j-2) - u(i, j)(b(i+1, j) + b(i-1, j) + b(i, j+1) + b(i, j-1)) \Big). \end{aligned}$$

Estas aproximaciones que hemos mostrado son las más sencillas de entre las que pueden generarse a partir de las fórmulas progresiva, regresiva y centrada. Existen otras posibilidades más sofisticadas y robustas frente al ruido. Una primera lectura sobre este tema puede realizarse en [1, Appendix A]. Este libro también incluye un tratamiento amplio de los métodos varacionales en procesamiento de imágenes. Para el caso particular de la variación total, véase [2].

Pseudocódigo 1 Minimización de la norma $H^1(\Omega)$

- 1: Lee imagen u_0 . Fija valor de parámetros τ y λ . Establece criterio de parada
 - 2: **while** criterio de parada no se satisface **do**
 - 3: **for** cada píxel interior de la imagen, (i, j) , **do**
 - 4: $D2u(i, j) \leftarrow u(i+1, j) + u(i-1, j) + u(i, j+1) + u(i, j-1) - 4u(i, j)$
 - 5: $u(i, j) \leftarrow u(i, j) + \tau(D2u(i, j) + \lambda(u_0(i, j) - u(i, j)))$
 - 6: **for** cada píxel de la frontera de la imagen, (i, j) , **do**
 - 7: $u(i, j) \leftarrow$ valor del píxel interior más cercano
 - 8: Evalúa criterio de parada
-



Figura 3: De izquierda a derecha: imagen con ruido, imagen restaurada minimizando la variación total, e imagen restaurada usando el filtro bilateral.

Métodos basados en entornos o métodos no locales

Los algoritmos introducidos en la sección anterior han dado lugar a la consideración de operadores diferenciales de segundo orden, que se conocen como operadores de *difusión (local)*. Por ejemplo, si consideramos la discretización más elemental del laplaciano dada por la fórmula (9), vemos que el valor de $\Delta u(i, j)$ se obtiene como un promedio de los valores de u en un entorno cercano del punto (i, j) . De ahí el nombre de *difusión local*.

La idea de los algoritmos basados en difusión no local es la de considerar la cercanía no solo en el sentido local, entre píxeles, sino también en el sentido del valor de la intensidad de grises que les corresponde, es decir, si los correspondientes valores de u están cerca.

Esta idea da lugar a una familia de filtros, los filtros basados en entornos (*neighborhood filters*), entre

los que destacan los casos particulares de los filtros bilaterales y el filtro *Nonlocal Means*. Todos ellos tienen la siguiente forma general

$$N_{h,\rho}u(\mathbf{x}) = \frac{1}{C(\mathbf{x})} \int_{\Omega} T_h(|u(\mathbf{y}) - u(\mathbf{x})|) w_{\rho}(|\mathbf{x} - \mathbf{y}|) u(\mathbf{y}) d\mathbf{y}, \quad (10)$$

siendo $C(\mathbf{x})$ el factor de normlización

$$C(\mathbf{x}) = \int_{\Omega} T_h(|u(\mathbf{y}) - u(\mathbf{x})|) w_{\rho}(|\mathbf{x} - \mathbf{y}|) d\mathbf{y}.$$

En esta expresión w_{ρ} representa la difusión local y T_h la no local, y son llamados *núcleo espacial* y *núcleo de rango*, respectivamente.

Por ejemplo, si tomamos $T_h \equiv 1$ y $w_{\rho}(s) = \exp(-s^2/\rho^2)$ entonces obtenemos el clásico filtro gaussiano

$$N_{h,\rho}u(\mathbf{x}) = \frac{1}{C} \int_{\Omega} \exp\left(-\frac{|\mathbf{x} - \mathbf{y}|^2}{\rho^2}\right) u(\mathbf{y}) d\mathbf{y},$$

donde C es una constante. Puede demostrarse que este filtro es una aproximación al filtrado dado por el laplaciano (minimización de la norma $H^1(\Omega)$).

Para generar filtrado no local debemos dotar de contenido a la función T_h . Las elecciones más habituales, que dan lugar a los filtros bilateral y Yaroslavsky, respectivamente, son

$$T_h(s) = \exp(-s^2/h^2) \quad \text{y} \quad w_{\rho}(s) = \exp(-s^2/\rho^2) \quad \text{o} \quad w_{\rho}(s) = \chi_{(0,\rho)}(s),$$

donde χ denota la función característica de un conjunto. De este modo, obtenemos los filtros bilateral

$$B_{h,\rho}u(\mathbf{x}) = \frac{1}{C(\mathbf{x})} \int_{\Omega} \exp\left(-\frac{|u(\mathbf{x}) - u(\mathbf{y})|^2}{h^2}\right) \exp\left(-\frac{|\mathbf{x} - \mathbf{y}|^2}{\rho^2}\right) u(\mathbf{y}) d\mathbf{y},$$

y de Yaroslavsky

$$Y_{h,\rho}u(\mathbf{x}) = \frac{1}{C(\mathbf{x})} \int_{B(\mathbf{x},\rho)} \exp\left(-\frac{|u(\mathbf{x}) - u(\mathbf{y})|^2}{h^2}\right) u(\mathbf{y}) d\mathbf{y},$$

donde $B(\mathbf{x},\rho)$ denota un cuadrado de radio ρ centrado en x .

En general, estos filtros se utilizan en un solo paso. Sin embargo, dependiendo de los núcleos particulares que se utilicen, puede ser conveniente convertir la expresión (10) en una fórmula iterativa. Así, dada la imagen inicial u_0 , definimos la iteración

$$u_{n+1}(\mathbf{x}) = \frac{1}{C_n(\mathbf{x})} \int_{\Omega} T_h(|u_n(\mathbf{y}) - u_n(\mathbf{x})|) w_{\rho}(|\mathbf{x} - \mathbf{y}|) u_n(\mathbf{y}) d\mathbf{y},$$

siendo $C_n(\mathbf{x})$ el factor de normlización

$$C_n(\mathbf{x}) = \int_{\Omega} T_h(|u_n(\mathbf{y}) - u_n(\mathbf{x})|) w_{\rho}(|\mathbf{x} - \mathbf{y}|) d\mathbf{y}.$$

La principal ventaja de los filtros bilaterales es que, proporcionando unos resultados cualitativamente parecidos a los de los filtros obtenidos mediante consideraciones variacionales, como el de minimización de la variación total, el tiempo de ejecución puede reducirse enormemente.

En estas notas utilizaremos la discretización más simple de los filtros bilaterales, cuyos tiempos de ejecución se asemejan a los de los métodos variacionales. Utilizaremos un algoritmo de *fuerza bruta*. Algoritmos más sofisticados que reducen enormemente el tiempo de ejecución pueden encontrarse en [3] y sus referencias. Muchas de ellas pueden descargarse libremente en internet.

Discretización

La discretización más simple de la fórmula (10) se plantea del siguiente modo. Para un píxel $\mathbf{x} = (i_1, i_2)$ dado de la imagen, aproximamos la integral de (10) como

$$\begin{aligned} \int_{\Omega} T_h(|u(\mathbf{y}) - u(\mathbf{x})|) w_{\rho}(|\mathbf{x} - \mathbf{y}|) u(\mathbf{y}) d\mathbf{y} \\ \approx \sum_{j_1=1}^N \sum_{j_2=1}^M T_h(|u(j_1, j_2) - u(i_1, i_2)|) w_{\rho}(((i_1 - j_1)^2 + (i_2 - j_2)^2)^{1/2}) u(j_1, j_2), \end{aligned}$$

donde, por convención, tomamos el área de un píxel igual a uno. El factor de normalización, $C(\mathbf{x})$, se discretiza de forma análoga.

En principio, la fórmula anterior tiene una gran carga computacional dado que, para cada píxel \mathbf{x} , debemos realizar del orden de $N \times M$ operaciones para calcular su valor filtrado. Sin embargo, el núcleo espacial w_{ρ} suele ser una función con un soporte pequeño. Por ejemplo, para el filtro de Yaroslavsky tenemos que $w_{\rho} = 0$ para los píxeles \mathbf{y} que estén a una distancia mayor que ρ del píxel de referencia \mathbf{x} . Además, dentro del cuadrado $B_{\rho}(\mathbf{x})$, el núcleo vale uno, de modo que el cálculo se reduce a

$$\int_{\Omega} T_h(|u(\mathbf{y}) - u(\mathbf{x})|) w_{\rho}(|\mathbf{x} - \mathbf{y}|) u(\mathbf{y}) d\mathbf{y} \approx \sum_{j_1=i_1-\rho}^{i_1+\rho} \sum_{j_2=i_2-\rho}^{i_2+\rho} \exp\left(-\frac{|u(j_1, j_2) - u(i_1, i_2)|^2}{h^2}\right) u(j_1, j_2),$$

teniendo una simplificación similar el factor $C(\mathbf{x})$.

En el caso del filtro bilateral tenemos en cuenta que el núcleo espacial, dado por la función gaussiana, decae rápidamente a cero, y que su soporte puede aproximarse por un cuadrado de tamaño 2ρ , véase la

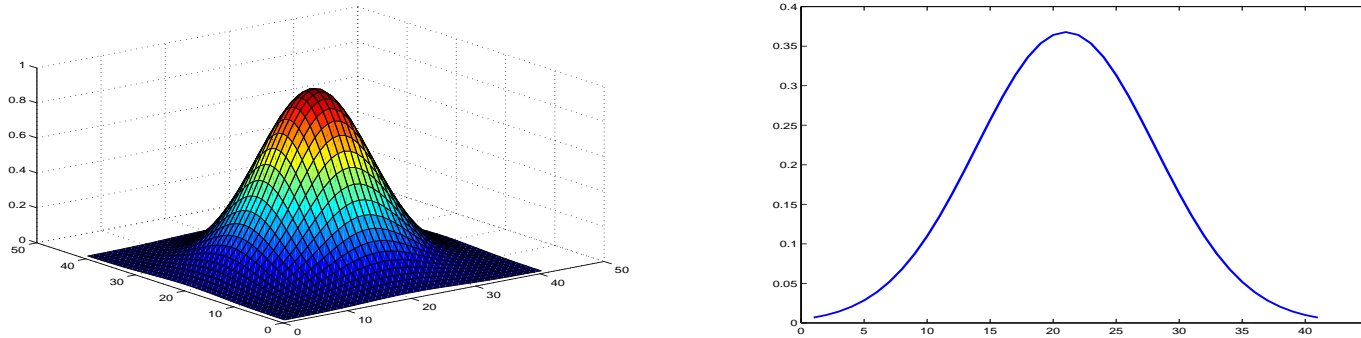


Figura 4: La función Gaussiana y su rápido decaimiento a cero ($\rho = 10$).

Figura 4. De este modo, obtenemos

$$\int_{\Omega} T_h(|u(\mathbf{y}) - u(\mathbf{x})|) w_{\rho}(|\mathbf{x} - \mathbf{y}|) u(\mathbf{y}) d\mathbf{y} \\ \approx \sum_{j_1=i_1-2\rho}^{i_1+2\rho} \sum_{j_2=i_2-2\rho}^{i_2+2\rho} \exp\left(-\frac{|u(j_1, j_2) - u(i_1, i_2)|^2}{h^2}\right) \exp\left(-\frac{(i_1 - j_1)^2 + (i_2 - j_2)^2}{\rho^2}\right) u(j_1, j_2),$$

y análogamente para $C(\mathbf{x})$.

Como puede observarse, en estas fórmulas tenemos un problema en el borde parecido al de las fórmulas de ecuaciones en derivadas parciales. En este caso, resolvemos el problema extendiendo la imagen por cero en una franja del mismo tamaño que el del radio del núcleo espacial. Los pseudocódigos correspondientes a estos filtros vienen dados por Pseudocódigo 2 y Pseudocódigo 3.

Pseudocódigo 2 Filtro Yaroslavsky

- 1: Lee imagen u_0 . Fija valor de parámetros h y ρ .
 - 2: **for** cada píxel de la imagen, (i, j) , **do**
 - 3: Extraer la región local: $iMin = \max(i - \rho, 1)$, $iMax = \min(i + \rho, N)$, etc.
 - 4: $I \leftarrow u_0(iMin : iMax, jMin : jMax)$
 - 5: $H \leftarrow T_h(I - u_0(i, j))$
 - 6: $u(i, j) \leftarrow \text{sum}(H * I) / \text{sum}(H)$
-

Pseudocódigo 3 Filtro bilateral

- 1: Lee imagen u_0 . Fija valor de parámetros h y ρ .
 - 2: Calcula núcleo espacial:
 - 3: $[X, Y] \leftarrow \text{meshgrid}(-2\rho : 2\rho, -2\rho : 2\rho)$;
 - 4: $G \leftarrow w_\rho(X, Y)$
 - 5: **for** cada píxel de la imagen, (i, j) , **do**
 - 6: Extraer la región local: $iMin = \max(i - \rho, 1)$, $iMax = \min(i + \rho, N)$, etc.
 - 7: $I \leftarrow u_0(iMin : iMax, jMin : jMax)$
 - 8: $H \leftarrow T_h(I - u_0(i, j))$
 - 9: $F \leftarrow H * G$
 - 10: $u(i, j) \leftarrow \text{sum}(F * I) / \text{sum}(F)$
-

Bibliografía

- [1] Aubert G, Kornprobst P (2006) Mathematical problems in image processing: partial differential equations and the calculus of variations, Springer.
- [2] Chambolle A, Caselles V, Cremers D, Novaga M, Pock T (2010) An introduction to total variation for image analysis, Theoretical foundations and numerical methods for sparse recovery, Radon S Comput Appl Math 9: 263–340
Versión *libre* en: <https://hal.archives-ouvertes.fr/hal-00437581>
- [3] Galiano G, Velasco J (2014) Neighborhood filters and the decreasing rearrangement. J Math Imaging Vision, DOI: 10.1007/s10851-014-0522-3.
Versiones *libres* en: <http://arxiv.org/abs/1311.2191>, <http://arxiv.org/abs/1406.7128>