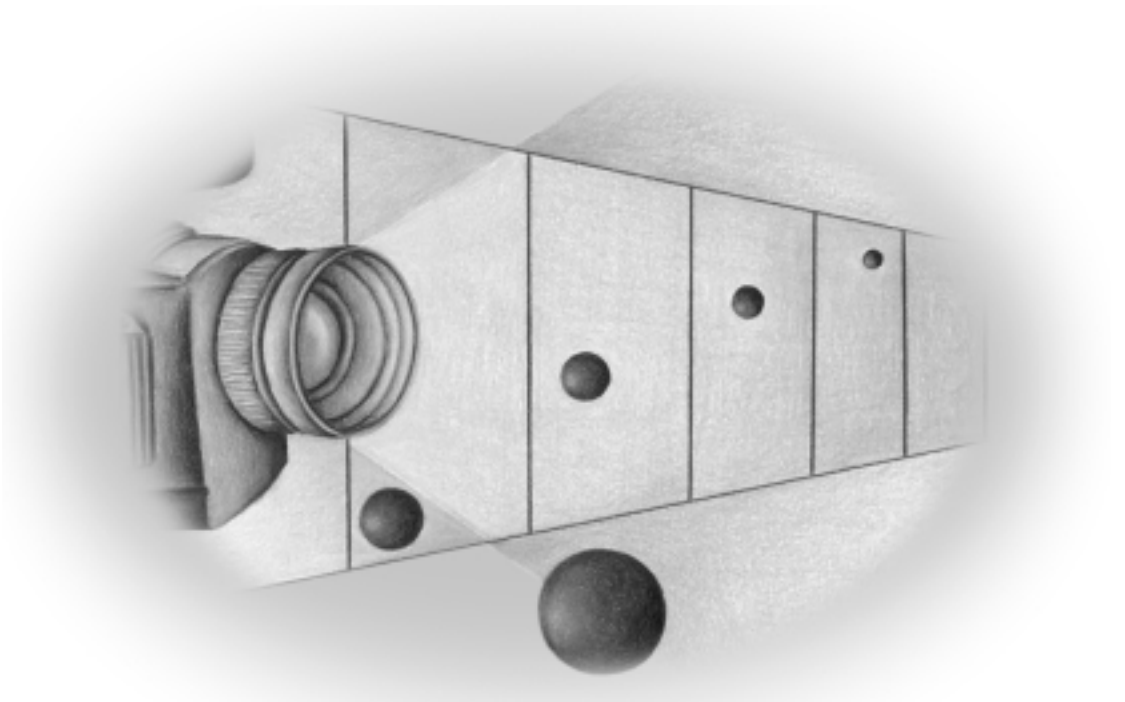


Principios de tratamiento de la imagen (2^a parte)



Índice

Etapa 1: Transformaciones espaciales	5
Introducción	5
Concepto de plantilla y su aplicación	6
Carga computacional asociada a una transformación espacial	7
Tratamiento de los píxeles del contorno	8
Filtrado paso bajo de la imagen	9
Aplicaciones de los filtros paso bajo	9
Filtros de promedio	10
Aplicación de las plantillas a la imagen	11
Filtros ponderados	13
Filtros paso bajo direccionales	14
Algoritmo básico	15
Resultados	17
Filtros dependientes de un parámetro: filtros gaussianos	18
Filtrado paso alto	21
Obtención de la plantilla del filtro paso alto	21
Representación de la imagen resultante en un filtro paso alto	22
Otras plantillas para el filtrado paso alto	23
Estimación del gradiente y detección de contornos	24
Estimación del módulo del gradiente a partir de ejes de coordenadas	25
Plantillas para estimar los componentes del gradiente	26
Plantillas comunes para estimar derivadas direccionales	28
Derivada de la plantilla gaussiana	31
Operadores de Laplace y cruces por cero para la estimación de contornos	32
Filtros paso alto para el realce de contornos	37
Etapa 2: Transformaciones espaciales no lineales y procesado morfológico	40
Introducción	40
Filtros de mediana	40
Elemento estructurante	42
Filtros de valor máximo y de valor mínimo	44
Procesado morfológico	45
Dilatación y erosión en imágenes binarias	45
Dilatación binaria	46
Erosión binaria	47
Combinando las operaciones de dilatación y erosión	49
Operaciones de opening y closing	50
Gradiente morfológico	51
Gradiente morfológico en imágenes de nivel de gris	52

Etapa 3: Transformaciones geométricas	53
Introducción	53
Representación gráfica de las transformaciones geométricas	55
Metodología de la transformación	55
Corrección hacia atrás (<i>backward</i>)	56
Interpolación de valores	57
Interpolación de orden cero	58
Interpolación bilineal.....	59
Interpolación bicúbica.....	59
Transformaciones más habituales	60
Cambio de escala	61
Rotación	61
Sesgo (<i>skew</i>)	62
Transformación afín	62
Transformación bilineal.....	62
Deformaciones puntuales	63
Etapa 4: Otras aplicaciones	65
Introducción	65
Metamorfosis de imágenes	65
<i>Morphing</i> utilizando deformaciones puntuales.....	66
Fusión de imágenes	68
Fotografía panorámica	69

Etapa 1: Transformaciones espaciales

Introducción

En general, utilizamos las transformaciones espaciales para obtener versiones filtradas de la imagen que enfatizen o atenúen algunas de sus características. Los filtrados más habituales son el paso bajo, con el que por norma general se pretende reducir el ruido presente en la imagen, y el paso alto, con el que se busca enfatizar la definición de la imagen o extraer los contornos.

En este módulo consideraremos las transformaciones espaciales utilizadas de manera más usual para la mejora y acondicionamiento de imágenes. Hemos visto que en este tipo de transformaciones cada píxel de la imagen resultante depende de los píxeles de la imagen original situados en la proximidad (vecindad) del elemento que tiene sus mismas coordenadas. Sólo vamos a tratar con transformaciones de una imagen estática, por lo que no consideraremos la posible dependencia temporal que puede aparecer en una secuencia de vídeo.

Atendiendo al tipo de relación utilizado en la transformación las podemos clasificar en lineales y no lineales.

En los filtros lineales, la imagen resultante se puede expresar como una combinación lineal de los píxeles de la imagen original que se sitúan en la proximidad del elemento de imagen que se está transformando. En consecuencia, las transformaciones espaciales lineales responden a la ecuación:

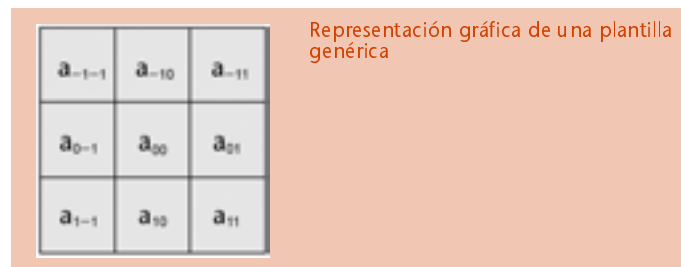
$$w[n, m] = \sum_{(i, j) \in I} a_{ij} \cdot u[n - i, m - j].$$

En esta ecuación, a_{ij} representa los coeficientes del filtro e I , el dominio de valores (i, j) sobre los que se realiza la transformación. El resultado de esta transformación debe ser cuantificado para que se corresponda con los niveles de gris con los que deseamos trabajar. Así pues, al resultado obtenido deberíamos aplicarle un cuantificador $Q(\cdot)$ que, por simplicidad, hemos omitido en esta ecuación.

La relación de la ecuación anterior corresponde a un filtro FIR (*finite impulse response*) bidimensional, ya que la salida corresponde a una combinación lineal de un número finito de elementos de entrada. Por tanto, es posible utilizar técnicas de diseño paralelas a las empleadas en el diseño de filtros digitales unidimensionales, generalizándolas para el caso bidimensional. En este módulo sólo vamos a considerar filtros lineales cuyos coeficientes toman valores fácilmente interpretables desde un punto de vista intuitivo.

En algunas aplicaciones, es conveniente utilizar relaciones **no lineales** entre la imagen original y la transformada. El caso más habitual es el conocido **filtro de mediana** que se utiliza para la reducción de ruido impulsional y que será examinado con detalle más adelante. En algunos casos, también es posible modificar los coeficientes de un filtro lineal con el objeto de adaptarnos a las propias características de la imagen. Este tipo de filtros se denominan **adaptativos o variantes**.

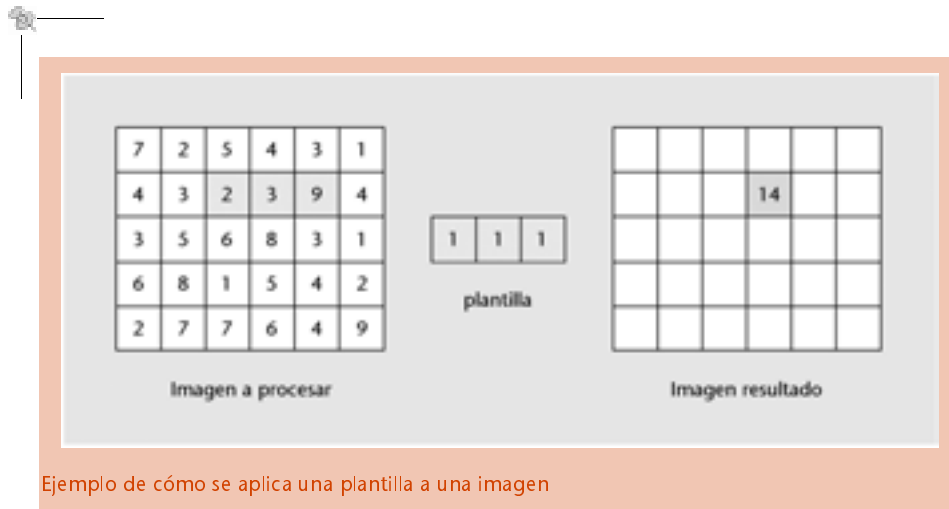
Concepto de plantilla y su aplicación



Si queremos representar de manera gráfica la transformación lineal asociada a un filtro, podemos hacerlo mediante el uso de plantillas. La plantilla representa los coeficientes con los que ponderaremos los píxeles situados en el entorno del elemento de imagen que estamos transformando.

La figura anterior representa una plantilla genérica en la que se procesa un entorno de 3×3 píxeles alrededor del elemento que se está transformando.

La interpretación de la plantilla es relativamente simple y, con cierta práctica, permite tener una idea muy aproximada de cuál será el resultado visual de una determinada transformación. En esta representación debemos tener en cuenta que para obtener el valor de un determinado elemento en la imagen transformada tendremos que situar la plantilla sobre el mismo elemento de la imagen original. El resultado de la transformación consiste en realizar el producto entre el valor del coeficiente y el valor del píxel correspondiente en la imagen original y, después, sumar todos estos resultados parciales.



Ejemplo de cómo se aplica una plantilla a una imagen

Si queréis entender mejor el proceso, considerad el ejemplo de la figura adjunta, en el que deseamos calcular el resultado que obtenemos al aplicarle la plantilla al elemento marcado en la imagen resultante. En este ejemplo, la plantilla asociada a la transformación espacial tiene un tamaño de 1×3 píxeles. Para determinar el resultado, superponemos la plantilla sobre la imagen original, centrada en el píxel que queremos calcular (en la figura se marcan los píxeles que intervienen en el cálculo), y efectua-

mos las operaciones convenientes. En este caso, como todos los coeficientes son iguales a la unidad, sólo tendremos que sumar los valores de los píxeles para obtener el resultado final.

Con el fin de obtener el resultado de toda la imagen procesada, iremos desplazando la plantilla (centrándola sobre todos los píxeles de la imagen original) y repetiremos la operación anterior.

Carga computacional asociada a una transformación espacial

La carga computacional asociada a un filtrado espacial de toda una imagen se puede estimar de una manera bastante sencilla. Si la imagen es de $N \times M$ píxeles y la plantilla es de $K \times L$ coeficientes, podemos determinar el número de productos y sumas totales que vamos a realizar:

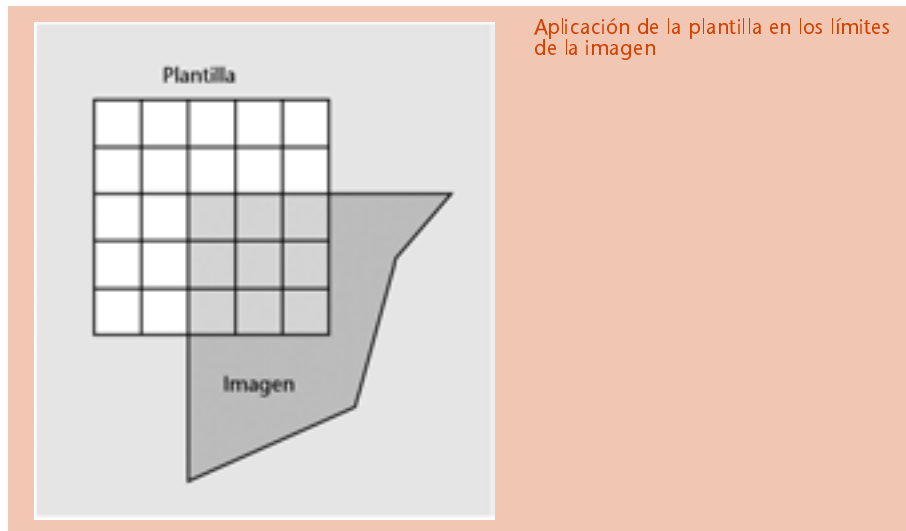
- Para cada elemento de la imagen resultante realizaremos $K \times L$ productos (el número de coeficientes de la plantilla).
- Para cada elemento de la imagen resultante, efectuaremos $K \times L - 1$ sumas.

Por lo tanto, el número total de productos es $(N \times M) \times (K \times L)$ y el número de sumas, $(N \times M) \times (K \times L - 1)$.

Si tomamos como referencia una imagen de 600×800 píxeles y una plantilla de 3×3 , obtenemos un total de 4.320.000 productos y 3.840.000 sumas. En algunos casos, estas operaciones se pueden reducir si algunos de los coeficientes de la plantilla son nulos (no es necesario multiplicar ni sumar) o iguales a la unidad (no es necesario realizar el

producto). Cuando el coeficiente toma el valor $1/2$ o $1/4$, la operación de multiplicación también puede verse simplificada mediante su sustitución por desplazamientos binarios.

Tratamiento de los píxeles del contorno



Uno de los problemas intrínsecos a cualquier transformación espacial es el de definir qué tipo de procesado se realiza cuando una parte de la plantilla queda fuera de los límites de la imagen. En efecto, cuando calculamos los elementos situados en los extremos de la imagen (podéis ver la figura anterior), parte de la plantilla aparece situada fuera de los límites de la imagen, por lo que no dispondremos del valor del elemento de imagen.

Existen varias soluciones prácticas:

- 1) Suponer que todos los elementos situados fuera del contorno de la imagen son nulos.
- 2) El tamaño de la imagen resultante es algo menor que el de la imagen original, de manera que la plantilla siempre se encuentra completamente contenida en la imagen.
- 3) Extender el tamaño de la imagen original replicando las primeras y últimas filas y columnas, de modo que la imagen resultante tenga siempre el mismo tamaño que la original (sin extender) y que la plantilla siempre quede incluida en la extensión de la primera.

A pesar de que todas estas soluciones son del mismo modo válidas, probablemente la que se utiliza con mayor frecuencia es la tercera.

Filtrado paso bajo de la imagen



Se entiende como filtrado paso bajo de una señal bidimensional cualquier transformación que deje inalteradas las regiones uniformes de la imagen y atenúe las transiciones bruscas.

En la figura hallamos un ejemplo de una imagen que ha sido filtrada paso bajo, haciendo posible observar una considerable pérdida de nitidez debido a la aparición de transiciones graduales entre los distintos objetos que componen la escena. El filtrado paso bajo se interpreta, por norma general, como un desenfoque de la imagen, ya que la sensación que produce es que la escena ha sido captada por una cámara que no se ha enfocado de la manera correcta. De hecho, es posible comprobar analíticamente que un desenfoque de la cámara se corresponde con una reducción del contenido espectral de la imagen en las regiones de alta frecuencia.

El filtrado paso bajo equivale a realizar un desenfoque de la imagen.

Aplicaciones de los filtros paso bajo

El interés en filtrar una imagen paso bajo se puede deber a:

a) **La existencia de ruido en la imagen original.** En algunos casos aparece un cierto nivel de ruido en el dispositivo de adquisición. Este ruido se aprecia en forma de cambios bruscos y aleatorios que se superponen con la señal de interés, motivo por el cual, un filtrado controlado de la imagen puede reducir la visibilidad del ruido (se atenúan las transiciones bruscas del nivel), por lo que, en general, deberemos buscar un compromiso adecuado entre la reducción de nitidez y ruido en la imagen.

b) La composición de imágenes capturadas por distintos dispositivos. En muchos trabajos de retoque fotográfico o composición de imágenes es habitual disponer de imágenes con calidades distintas. Considerad, por ejemplo, que queremos integrar un fragmento de una fotografía capturada por una cámara digital moderna junto a una imagen escaneada de una fotografía antigua. La resolución del fotograma digital será mucho mayor que la que disponemos en la fotografía antigua, de modo que, si hay que componer ambas imágenes, tendremos que llevar a cabo un filtrado paso bajo de la parte digital para que se integren con naturalidad.

Las distintas alternativas para el filtrado paso bajo que se proponen en este apartado buscan optimizar el compromiso entre la reducción de ruido, la pérdida de nitidez y la complejidad computacional.

Filtros de promedio

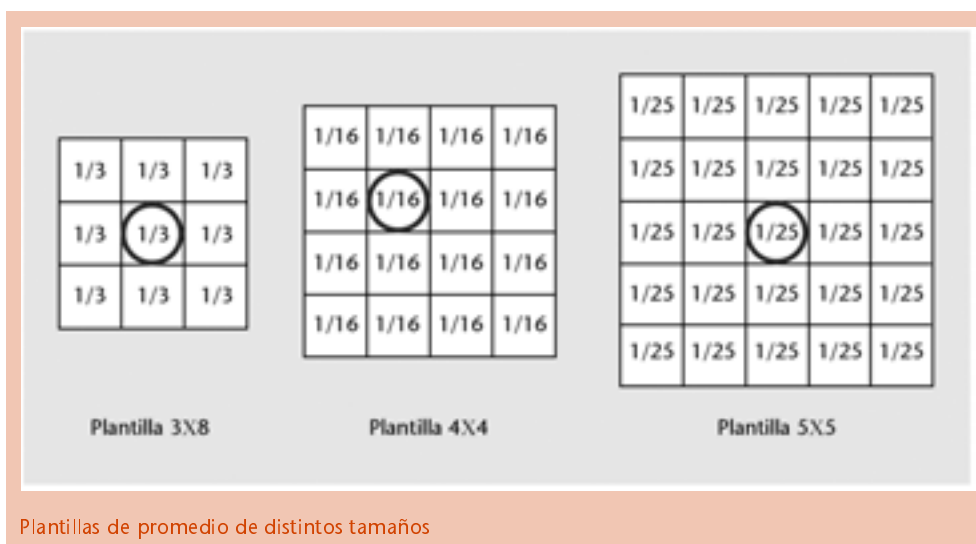
Los filtros de promedio (*averaging*, en inglés) consisten simplemente en sustituir cada píxel de la imagen por el valor medio del nivel de gris en un dominio centrado sobre el punto que estamos procesando. Podemos expresar la ecuación de filtrado como:

$$v[n, m] = \frac{1}{N_p} \cdot \sum_{(i, j) \in I} u[n-i, m-j]$$

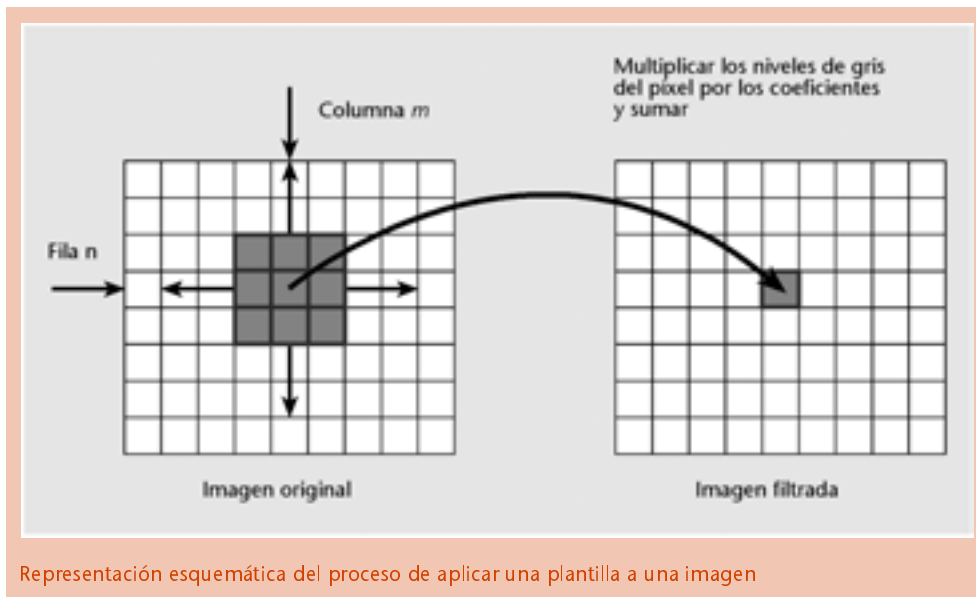
En esta ecuación observamos que todos los coeficientes del filtro son constantes y toman el valor $1/N_p$, siendo N_p el número de píxeles del dominio donde se realiza el promedio.

Para simplificar la representación gráfica del filtro, solemos indicar la ecuación anterior mediante una plantilla que indica de forma simultánea el dominio sobre el que se realiza el promedio y el valor de los coeficientes.

En la figura anterior hallamos la representación de plantillas de distintos tamaños correspondientes a filtros de promedio. El elemento central sobre el que se aplica la plantilla aparece señalado con un círculo.



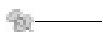
Aplicación de las plantillas a la imagen



La interpretación de cómo operan las plantillas sobre la imagen tiene que quedar bien clara: para obtener el valor transformado correspondiente a la posición con coordenadas (n, m) , hay que centrar la plantilla sobre el píxel situado en estas coordenadas de la imagen original, multiplicar cada uno de los coeficientes por los elementos de imagen situados en las mismas posiciones que la plantilla y, finalmente, sumar todos estos resultados. La plantilla se irá desplazando a lo largo de todo el dominio de definición de la imagen original para determinar el resultado de la transformación.

La figura anterior nos muestra el proceso mencionado.

Por norma general, las dimensiones de las plantillas de promedio son idénticas en las dos variables (horizontal y vertical), con lo que pérdida de definición también es equivalente en las dos direcciones principales. Para plantillas de tamaño $N \times N$, con N impar, podemos identificar el píxel central sin ambigüedades. Por otra parte, en el supuesto de que N sea par, podemos utilizar como píxel de referencia cualquiera de los cuatro que quedan en el centro. (En la figura anterior hemos indicado con un círculo el píxel que hemos tomado como referencia aunque, normalmente, ésta es una información que no se suele proporcionar).



En el siguiente ejemplo aparece el resultado de haber aplicado diferentes plantillas paso bajo sobre una imagen en la que aparece un cierto nivel de ruido gaussiano. El ruido tiene una desviación típica con un nivel de gris igual a 32 (con respecto a una codificación de la imagen de 256 niveles).

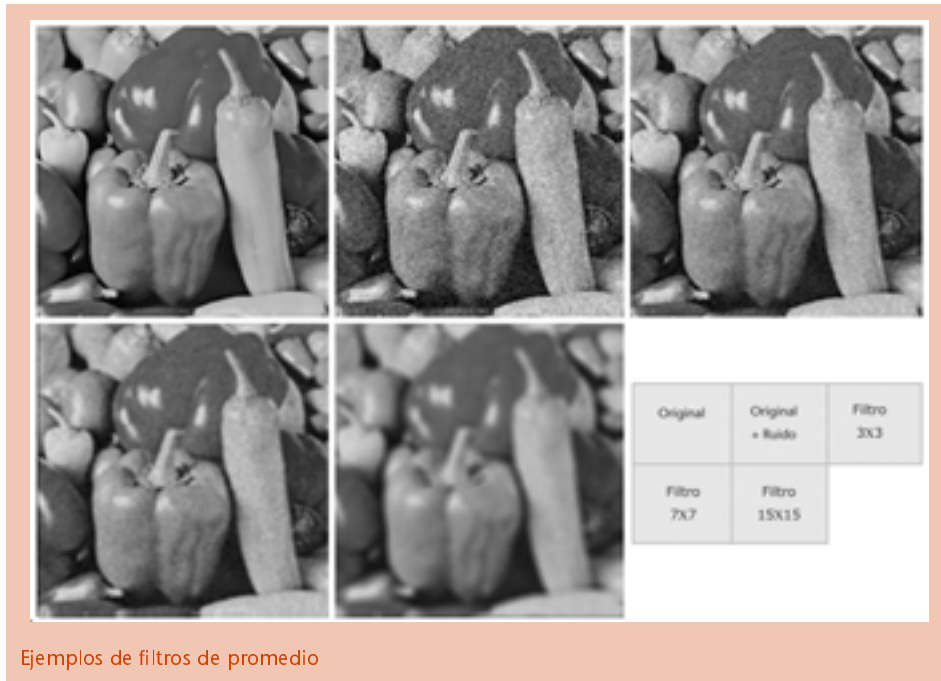
Este nivel es equivalente a una relación señal a ruido de aproximadamente 12 dB. En efecto:

$$SNR = 10 \cdot \log_{10} \left(\frac{P_{señal}}{P_{ruido}} \right) = 10 \cdot \log_{10} \left(\frac{128^2}{32^2} \right) \approx 12 \text{ dB},$$

donde hemos calculado la potencia de la señal como el cuadrado de la desviación del margen dinámico con respecto a la media de la señal, media que hemos supuesto de 128 (256 – valor medio)².

En estos ejemplos podemos observar el compromiso entre la pérdida de nitidez de la imagen y la reducción del nivel de ruido. El desenfoque producido por una plantilla 3×3 no es demasiado notorio, aunque no produce una suficiente atenuación de ruido. Con la plantilla de 15×15 , el ruido queda prácticamente eliminado, pero el desenfoque de la imagen es considerable.

Es preciso que tengamos en cuenta que el desenfoque introducido en la imagen por un filtro de promedio depende tanto del tamaño de la plantilla, como del tamaño de la imagen original. Así pues, una plantilla 3×3 puede introducir una pérdida de nitidez considerable en imágenes de baja resolución. En nuestro ejemplo, la imagen original es de 512×512 píxeles, por lo que el desenfoque introducido por una plantilla 3×3 no es significativo.



Reducción del nivel de ruido en un filtro de promedio

Resulta interesante evaluar la mejora en relación señal a ruido que introduce un filtro de promedio. Para independizar este cálculo de la pérdida de definición en la imagen, supondremos que el filtro se aplica sobre una imagen con nivel de gris uniforme, por lo que la plantilla de promedio carece de efecto alguno sobre la señal (el valor

medio de una señal uniforme coincide con la propia señal). Supondremos, también, que a la señal útil se le superpone un ruido blanco gaussiano con media cero y varianza σ^2 ; en consecuencia, la potencia de ruido de la señal sin filtrar coincide con la varianza. Al filtrar la señal, el nivel de ruido vendrá dado por:

$$w_{filt}[n, m] = \frac{1}{N_p} \cdot \sum_{(i, j) \in I} w[n-i, m-j],$$

donde $w_{filt}[n, m]$ representa el componente de ruido filtrado y $w[n, m]$ el componente de

ruido original. Podemos estimar la potencia de ruido después del filtrado como:

$$\begin{aligned} E[w_{filt}^2[n, m]] &= E\left[\left(\frac{1}{N_p} \cdot \sum_{(i, j) \in I} w[n-i, m-j]\right)^2\right] = \\ &= \frac{1}{N_p^2} \cdot \sum_{(i, j) \in I} E[(w[n-i, m-j])^2] = \frac{1}{N_p^2} \cdot \sum_{(i, j) \in I} \sigma^2 = \frac{\sigma^2}{N_p}, \end{aligned}$$

donde hemos supuesto que no existe correlación entre el ruido de entrada y muestras

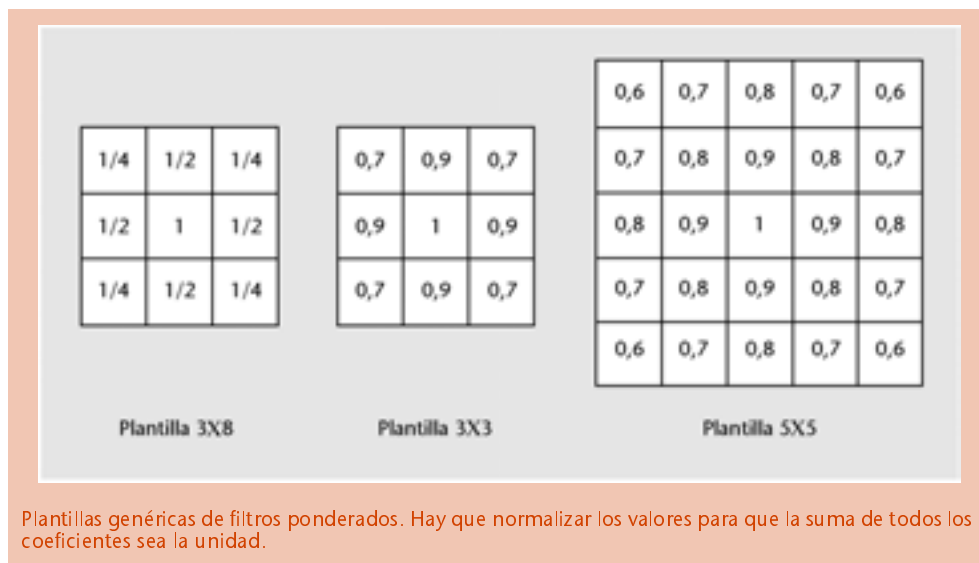
correspondientes a posiciones espaciales distintas.

Así pues, el filtrado de promedio de la imagen reduce la potencia de ruido en un factor igual al número de elementos de imagen promediados. Para una plantilla 3×3 , la mejora en relación señal a ruido será, por tanto, de unos 9,54 dB (se promedian 9 píxeles).

Las mejoras conseguidas con una plantilla 5×5 , 7×7 y 15×15 son, respectivamente, de 13,9 dB, 16,9 dB y 23,5 dB. En general, para estimar la mejora en decibelios debida al promedio, haremos uso de la siguiente ecuación:

$$SNR_{mejora} = 10 \cdot \log_{10} N_p$$

Filtros ponderados



Una alternativa a los filtros de promedio estricto consiste en ponderar las muestras que intervienen en el promedio de forma distinta, en función de su proximidad al elemento central. En la figura mostramos algunas plantillas en las que incluimos los pesos relativos con los que se pondera cada elemento. Es preciso escalar estas plantillas con una constante multiplicativa, con objeto de que al aplicarlas a una región con nivel de gris uniforme su ganancia sea exactamente igual a la unidad.

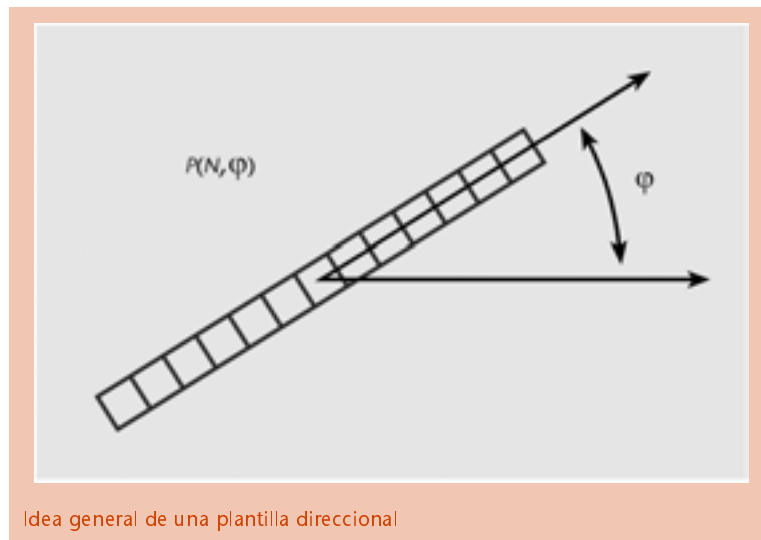
Para determinar el valor de esta constante multiplicativa basta con sumar todos los coeficientes de la plantilla y tomar el inverso de esta suma como una constante multiplicativa que se aplica directamente al resultado. Así, si aplicamos la plantilla de la izquierda a una región con nivel de gris uniforme p_k , deberemos multiplicar el resultado final por $1/4$ para que el resultado final sea el mismo nivel de gris:

$$P_{out} = \frac{1}{4} \cdot (p_k \cdot 1 + 4 \cdot p_k \cdot 1/2 + 4 \cdot p_k \cdot 1/4) = p_k$$

El objetivo de estas plantillas de ponderación es obtener un compromiso más estrecho entre desenfoque de la imagen y atenuación de componentes de ruido.

Los filtros de ponderación pueden introducir un cierto incremento de la carga computacional con respecto a los filtros de promedio. En efecto, los filtros de promedio estricto se pueden implementar de forma muy simple ya que sólo requieren la realización de sumas. Además, si la plantilla es de un tamaño considerable, puede resultar rentable actualizar el resultado del filtro sumando en exclusiva los nuevos elementos que intervienen en la plantilla y restando los salientes. En el caso de filtros con ponderaciones no uniformes, esta simplificación no es evidente y hay que llevar a cabo multiplicaciones relativamente complejas. Por todo esto, su uso en tratamiento de imagen es menos habitual y sólo se utilizan en algunas aplicaciones específicas. Entre los filtros de ponderación más utilizados está el que hemos representado a la izquierda de la figura, cuya principal ventaja respecto del resto de los ejemplos es que los coeficientes multiplicativos y el factor de escala son siempre potencias de 2, de forma que podemos realizar las operaciones mediante simples desplazamientos de bits. Otros filtros con coeficientes ponderados que se utilizan con cierta frecuencia son los filtros gaussianos u otras variantes, cuya plantilla final depende de un parámetro, de modo que podemos adaptarlas con facilidad a las características de la imagen.

Filtros paso bajo direccionales



Idea general de una plantilla direccional

Los filtros direccionales pretenden obtener un mejor compromiso entre la pérdida de nitidez y el grado de atenuación del ruido mediante la realización de varios filtros direccionales de la imagen y la selección, *a posteriori*, de cuál de ellos produce un mejor resultado en cada parte de la imagen.

Los filtros direccionales obtienen un mejor compromiso entre la reducción del ruido y la pérdida de nitidez.

Los filtros direccionales son similares a los de promedio, pero la plantilla no es simétrica en ambas variables, de modo que la pérdida de definición sólo se produce en la dirección en que actúa el filtro. En la figura aparece representada una plantilla estándar para un filtro direccional que actúa promediando únicamente los elementos de imagen que se encuentran alineados en la dirección asociada al filtro.

Algoritmo básico

La idea básica para realizar un filtrado direccional de una imagen se puede resumir en los siguientes puntos:

a) Calcular para cada punto de la imagen el resultado de la aplicación de un filtro direccional:

$$v_{\phi_k}[n, m] = P_{\phi_k}(u[n, m]),$$

donde $P_{\phi_k}^N$ representa la plantilla orientada según el ángulo ϕ_k y que realiza el promedio de N puntos de la imagen. En general, el conjunto de ángulos sobre los que se calcula el promedio direccional es reducido. Los más habituales son $\{0^\circ, 30^\circ, 60^\circ, 90^\circ, 120^\circ, 150^\circ\}$, o bien $\{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$.

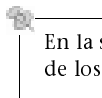
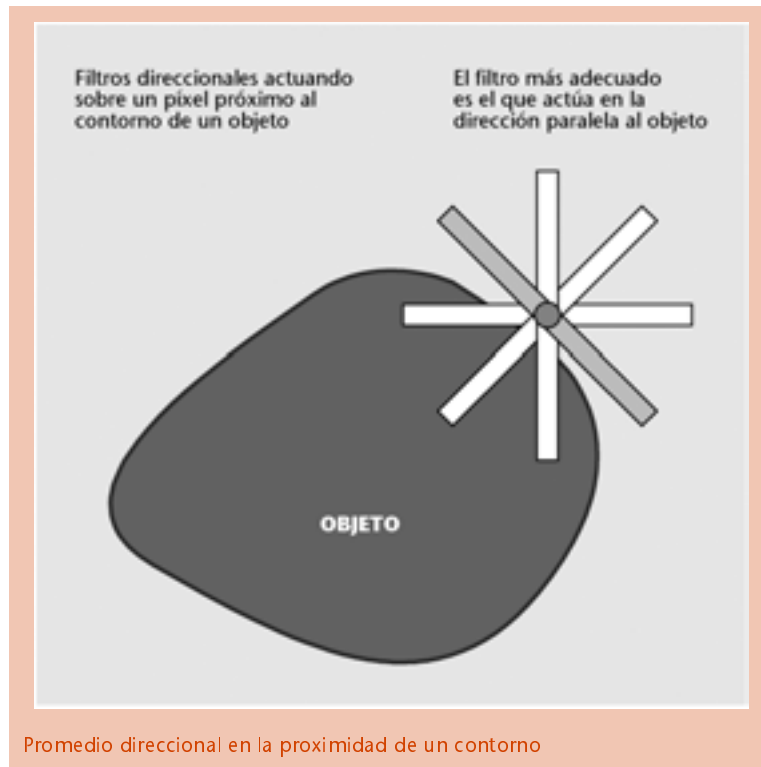
b) Para obtener el valor del píxel transformado, elegiremos el resultado direccional que ha proporcionado un valor más cercano al nivel de gris que tenía el píxel en la imagen original.

$$v[n, m] = \{v_{\phi_k}[n, m] | \min(\|u[n, m] - v_{\phi_k}[n, m]\|), \forall \phi_k\}$$

La justificación de cómo este proceso permite reducir el nivel de ruido sin afectar de forma tan directa a la definición de la imagen se ilustra a modo de esquema en la figura adjunta.

Suponed un punto situado cerca de un contorno de la imagen; en tal caso, el filtro que proporcionará una respuesta óptima es el que actúa en una dirección paralela al contorno. Notad, también, que si el ruido se mantiene dentro de unos límites tolerables, éste será el filtro que producirá un resultado más próximo al nivel de gris que teníamos en la imagen original, ya que sólo está promediando elementos de imagen que, en condiciones normales, se asociarían al mismo objeto. El resto de los filtros direccionales promedian elementos de imagen que están en parte fuera y en parte dentro de la región, por lo que es de esperar que el nivel de gris obtenido difiera bastante del de la imagen original. En regiones uniformes, todos los filtros direccionales producirán resultados de salida parecidos, motivo por el que se hace relevante determinar cuál de ellos escogemos para obtener la imagen original. Así

pues, el criterio de tomar como resultado el filtro direccional que produzca un nivel de gris más próximo al de la imagen original parece coherente y es, por otra parte, simple de implementar.



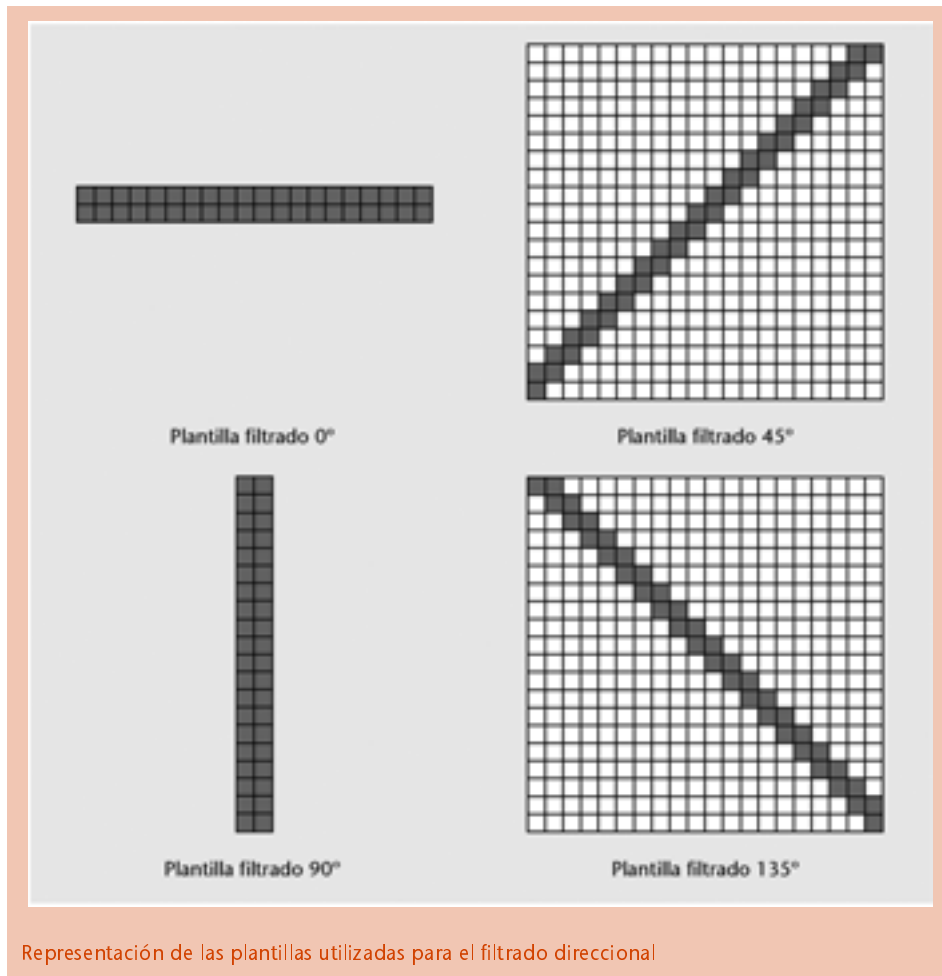
En la siguiente figura vemos la comparación de los resultados obtenidos mediante un fil-

tro direccional y uno de promediado con la misma imagen del ejemplo anterior.



Hemos obtenido el filtrado direccional utilizando 4 direcciones principales $\{0^\circ, 45^\circ, 90^\circ, 135^\circ\}$. Las plantillas utilizadas para realizar el filtrado en las direcciones horizontal y vertical promediaban un total de 40 píxeles en plantillas de 2×20 y 20×2 , respectivamente. Los filtrados en las direcciones de 45° y 135° se han efectuado con plantillas de 20×20 que contie-

nen ceros en todas las posiciones exceptuando la diagonal principal y la superior, en las que se ponderan por igual todos los elementos de imagen. El número total de elementos de imagen considerados en estas dos direcciones es, por tanto, de 39. En el gráfico adjunto hallaremos una representación esquemática de las plantillas usadas para el filtrado direccional.



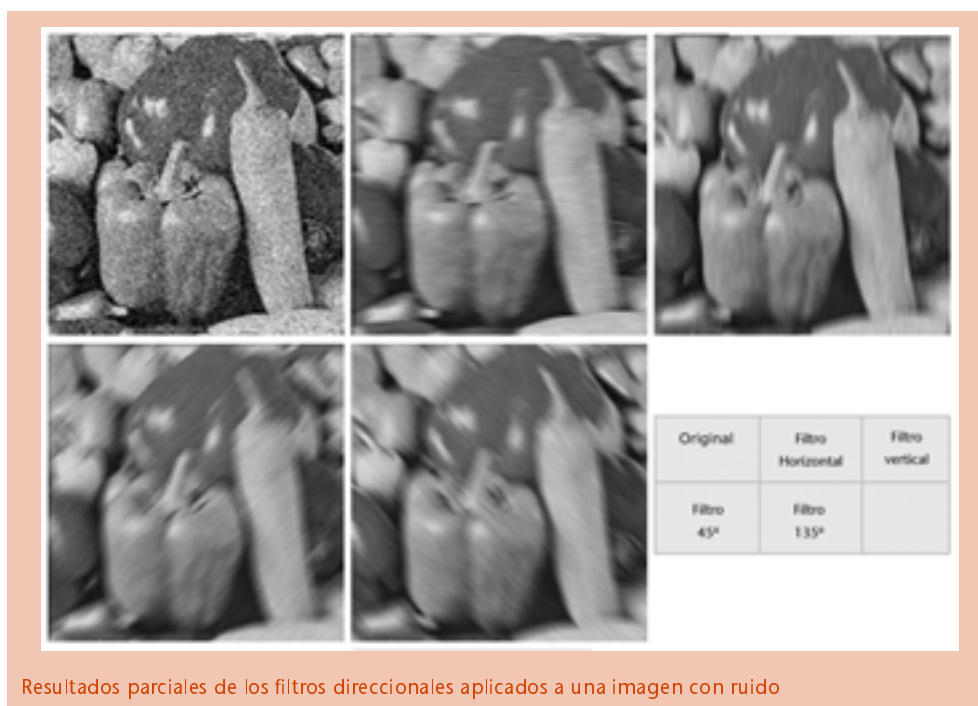
Resultados

Si comparamos en los ejemplos anteriores la imagen obtenida mediante el filtrado direccional con la obtenida con un filtro de promedio que utiliza un número de coeficientes parecido (filtro de 7×7 coeficientes), vemos que, desde un punto de vista visual, el filtrado direccional consigue una mayor atenuación del ruido –a pesar de que el número de coeficientes (40) es algo menor que con el promediado convencional (49). La justificación de este efecto es clara e intuitiva: con el filtrado direccional se promedian, por lo general, regiones con nivel de gris más o menos uniformes, haciendo que la percepción visual del ruido sea menor. La definición de los contornos y el filtrado direccional son excelentes aunque su principal problema es que presentan un cierto difuminado de zonas uniformes, próximas al contorno, que se encuentran dentro de una misma región. También podéis observar la aparición de algunos puntos con ruido de tipo impulsivo en la proximidad de los contornos debido a los errores en la selección de la dirección de filtrado adecuada.

En la siguiente figura se compara la imagen obtenida mediante el uso del filtrado direccional anterior con la resultante de haber aplicado una plantilla de promedio de 15×15 puntos. En este ejemplo, ambas imágenes tienen un nivel de ruido aparente similar.



No obstante, la pérdida de nitidez causada por el filtro de promedio es más que evidente. Para finalizar, mostraremos la imagen original con ruido y los distintos filtrados direccionales individuales que se han aplicado sobre ella. Apreciad de qué manera es fácil identificar el tipo de filtrado a partir de la dirección en que se producen las pérdidas de definición en la imagen.



Filtros dependientes de un parámetro: filtros gaussianos

Otra alternativa para obtener un mejor compromiso entre el nivel de reducción de ruido y la nitidez es usar plantillas cuyos coeficientes varían en función de las caracterís-

ticas de la imagen. En general se trata de estimar, en función de las propiedades locales de la imagen, un parámetro que controla el grado con el que se promedian los elementos de la imagen en cada región. Existe una gran cantidad de propuestas distintas cuyo uso y resultados dependen, en muchas ocasiones, de las propias imágenes y cuyo análisis detallado escapa de los propósitos de este texto. Nos disponemos a describir, en exclusiva, una de las plantillas más populares dependiente de un parámetro: la plantilla gaussiana.

En general, el uso de este tipo de filtros exige una fuerte carga computacional y un detallado ajuste de los parámetros, por lo que sólo se utilizan en aplicaciones en las que aparece un elevado nivel de ruido y el resto de los métodos no producen los resultados deseados.

Podemos expresar la plantilla gaussiana bidimensional de forma analítica mediante la siguiente ecuación:

$$P_{gauss}[i, j] = \begin{cases} C(\xi) \cdot \exp\left(-\frac{i^2+j^2}{\xi^2}\right) & \text{si } (i, j) \in I \\ 0 & \text{otros valores} \end{cases}$$

Es decir, se trata de un filtro cuyos coeficientes se expresan como una función gaussiana dentro del ámbito de definición de la plantilla. El tipo de ponderación de los datos de la imagen depende del valor del componente ξ que se puede asociar con la desviación típica de la función gaussiana. Hay que ajustar el valor de la constante $C(\xi)$ en función del parámetro ξ para que la ganancia de la plantilla sea la unidad; es decir, que cuando se aplique a una región con nivel uniforme, no se modifique el nivel de gris de la imagen. Así pues, podemos determinar el valor de la constante como:

$$C(\xi) = \frac{1}{\sum_{(i,j) \in I} \exp\left(-\frac{i^2+j^2}{\xi^2}\right)}$$

En caso de que el parámetro ξ aparezca determinado en función de las características locales de la imagen, será necesario calcular la constante de manera dinámica, lo cual supone una considerable carga computacional adicional.

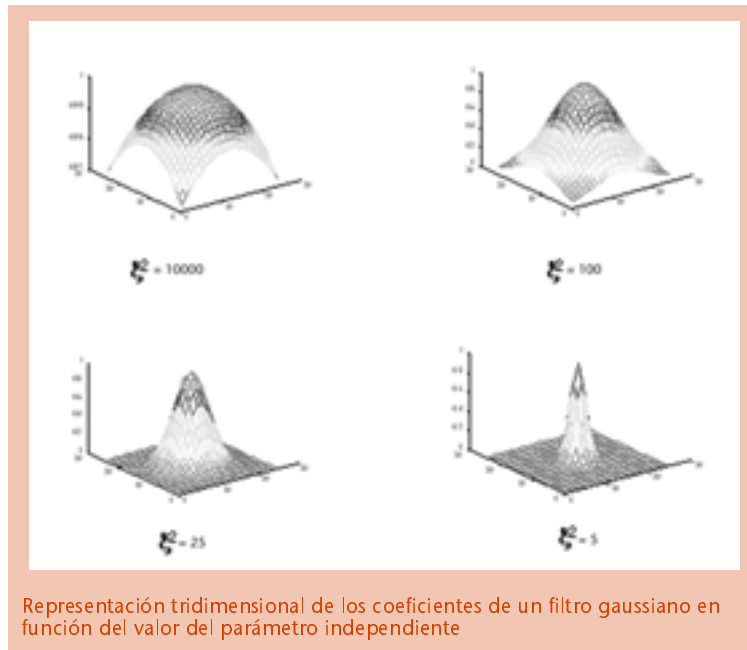


En la figura nos encontramos con la superficie bidimensional de una campana de Gauss para algunos valores del parámetro ξ , representándose para una plantilla de 25×25 píxeles. Cuando ξ tiene un valor elevado (p.ej. $\xi^2 = 10.000$), la función decrece lentamente con respecto a su valor unitario de origen. En este caso, se lleva a cabo un importante filtrado paso bajo o suavizado de la imagen ya que to-

dos los elementos de imagen están ponderados por coeficientes próximos a los que tomarían con un filtro de promedio convencional. Sin embargo, a medida que ξ disminuye, los coeficientes decrecen con una mayor rapidez conforme nos acercamos a los extremos de la plantilla, reduciéndose el número de píxeles que intervienen en el promediado espacial de la imagen.

Las gráficas indican que para valores de ξ^2 próximos a 5, prácticamente no se realiza ningún promedio, puesto que sólo el píxel

central y sus vecinos más inmediatos son significativos.



Mecanismos de control del parámetro independiente

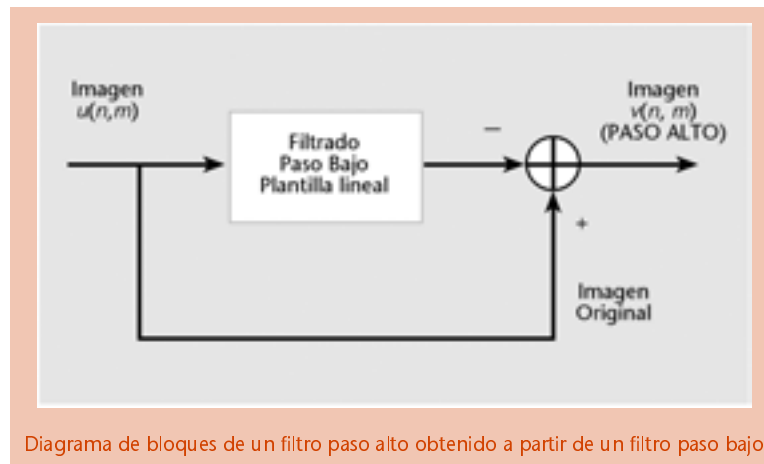
Tras haber entendido cómo afecta el parámetro de control a la forma y tipo de promediado que realiza la plantilla, parece evidente que tenemos que definir la filosofía general con la que procesaremos la imagen. En principio, deberíamos establecer algún procedimiento para determinar las características locales de la imagen que nos va a indicar cuándo estamos en una región prácticamente uniforme, en la que podemos aplicar un factor de promedio elevado para reducir el nivel de ruido sin perder nitidez en la imagen, y cuándo estamos en una región con elevados cambios, en la que es conveniente no realizar promedio alguno para no difuminar los contornos.

A menudo se utilizan funciones auxiliares para medir las características de la imagen y determinar el valor del parámetro adecuado. Una posible metodología consiste en determinar los contornos de la imagen mediante el uso de filtros paso alto y controlar el parámetro independiente en función de los valores obtenidos. En las regiones donde se detecta una presencia importante de contornos, aplicaremos un promediado de pocos píxeles (parámetro pequeño), mientras que en las regiones donde prácticamente no se han detectado contornos, podemos aplicar un promedia-

do en el que participen un gran número de píxeles.

La mejora de imágenes mediante filtros dependientes de un parámetro es bastante compleja, tanto desde el punto de vista computacional (cálculo de constantes de ganancia y aplicación de filtros de coeficientes variables), como en lo que respecta a ajustes de las funciones y parámetros. Por este motivo, su uso es relativamente poco frecuente y casi limitado a aplicaciones en las que se desea restaurar fotogramas afectados con un alto nivel de ruido, donde el resto de los métodos considerados en este apartado no generan resultados satisfactorios. La función gaussiana se puede sustituir por otras funciones como cosenos alzados, funciones de ponderación piramidal, etc. Con estas funciones alternativas, la intención es obtener una relación más lineal entre los coeficientes del filtro y el parámetro de control. En el caso de las funciones gaussianas, la forma de la función bidimensional es muy sensible al parámetro de control, de modo que, para determinados niveles, pequeños cambios del parámetro pueden colapsar con rapidez la función para que ésta actúe únicamente sobre el píxel central. En el resto de las funciones, esta relación es más lineal, así que el ajuste de parámetros es más simple e intuitivo.

Filtrado paso alto



El filtrado paso alto de una imagen consiste en mantener o enfatizar los cambios de nivel de gris mientras se suprimen las regiones con nivel de gris uniforme. Lo más usual a la hora de realizar estas operaciones es el uso de filtros lineales basados en plantillas tipo FIR. El resultado de un filtro paso alto puede ser interesante en sí mismo, por ejemplo para detectar de forma automática los contornos de un objeto, o para combinarlo con la imagen original y así enfatizar las variaciones de nivel de gris y mejorar la percepción de la imagen.

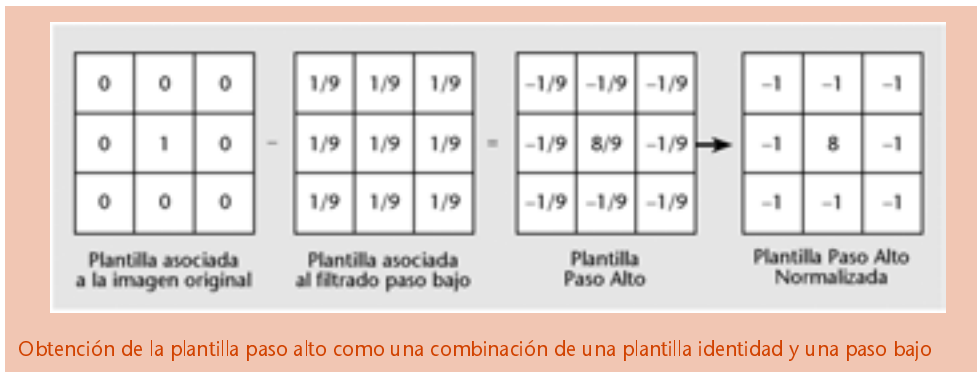
Las principales aplicaciones del filtrado paso alto son la mejora y la detección de contornos.

Para obtener una idea intuitiva de cómo deben ser los coeficientes de una plantilla paso alto podemos considerar que podemos obtener una señal paso alto restando de la imagen original una versión filtrada paso bajo de la misma (figura). En efecto, las regiones uniformes de la imagen se cancelan por la diferencia entre ambas señales, mientras que los componentes de alta frecuencia, que sólo están presentes en la imagen original, aparecerán en la salida.

Obtención de la plantilla del filtro paso alto

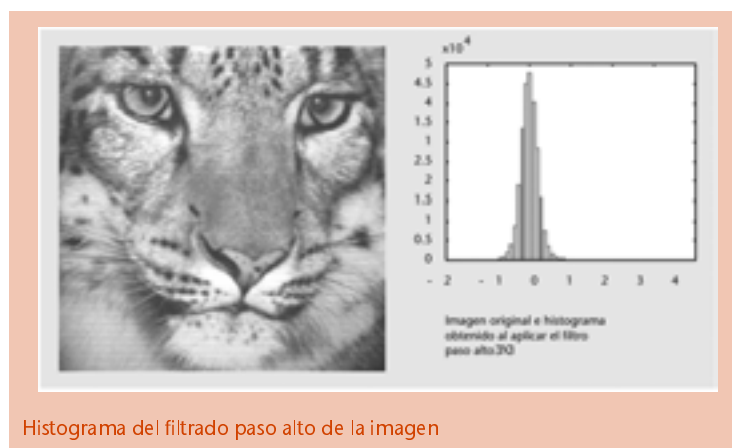
El sistema representado en la figura anterior es lineal ya que la salida se obtiene como una combinación directa de señales que, a su vez, son determinadas a partir de transformaciones lineales. Por lo tanto, podemos determinar la plantilla equivalente del sistema combinando las plantillas individuales utilizadas en cada rama del sistema.

En la siguiente figura representamos el resultado obtenido para el caso particular en que la plantilla del filtro paso bajo consista en un promedio en una ventana 3×3 . La plantilla resultante se puede escalar en el factor que el diseñador desee.



En el diseño de plantillas paso bajo, el factor de escala se encontraba condicionado por el hecho de que la ganancia del filtro en las regiones con nivel de gris uniforme fuera igual a la unidad, con objeto de que el nivel de gris medio fuera constante. No obstante, en el caso de filtros paso alto, no disponemos de ningún criterio predeterminado para calcular la ganancia del filtro. En el ejemplo que estamos considerando es más habitual operar con la plantilla normalizada porque las operaciones implicadas son simplemente desplazamientos de bits (multiplicar por 8) y restas. Una vez realizado el filtrado paso alto, es habitual la aplicación de transformaciones puntuales elementales para escalar el resultado y acondicionarlo para su representación gráfica.

Representación de la imagen resultante en un filtro paso alto



El filtrado paso alto de la imagen afecta directamente al margen dinámico, de modo que, para visualizar la imagen resultante, es necesario ajustar de nuevo los niveles. Además, los niveles obtenidos pueden ser positivos o negativos, por lo que es necesario definir qué criterio vamos a utilizar para su representación mediante niveles de gris exclusivamente positivos.

En la figura aparece el histograma que hemos obtenido tras aplicar a una imagen un filtro paso alto 3×3 como el considerado en el ejemplo anterior. Antes de aplicar el filtrado paso bajo, hemos normalizado la imagen para que los niveles de gris se correspondan con números reales entre 0 (negro) y 1 (blanco); el ejemplo supone que trabajamos con niveles de gris normalizados a la unidad.

El histograma resultante indica que el margen dinámico de la imagen filtrada toma valores situados entre $-1, 3$ y $3, 2$, de manera que, si queremos visualizarlos, será preciso que transformemos los niveles.

Nos encontramos con dos alternativas fundamentales a la hora de visualizar una imagen. La primera consiste en tomar el valor absoluto del resultado y, con posterioridad, escalarlo para que cubra el margen dinámico de interés (por norma general se ajusta para que el valor máximo coincida con el blanco). Esta representación destaca los componentes de alta frecuencia de la imagen sin tener en cuenta el signo y sentido de los cambios y suele ser útil y fácil de interpretar en la mayor parte de los problemas de reconocimiento. Asimismo, es habitual realizar una corrección sobre el resultado obtenido con el fin de facilitar la visualización de la información contenida en los niveles de gris próximos al negro.

Otra alternativa consiste en multiplicar la imagen resultante por una constante, de forma que el margen dinámico esté comprendido entre $-0,5$ y $0,5$ (-128 y 128 en caso de que mantengamos una representación con 8 bits) y, más tarde, añadirle al resultado un *offset* de $0,5$ que transforme el origen en el nivel de gris medio (el *offset* será de 128 para codificaciones con 8 bits). Esta forma de representar la imagen permite que nos hagamos una idea del signo resultante en la transformación, aunque la interpretación de la imagen no es tan directa como en el caso anterior. En algunas ocasiones es conveniente ajustar la primera constante multiplicativa para obtener imágenes con menor margen dinámico, lo que puede resultar visualmente más agradable.

En la siguiente figura indicamos los dos tipos de representación gráfica mencionados para la imagen considerada en este ejemplo. En el ejemplo de la izquierda, donde representamos el valor absoluto del filtrado paso alto, hemos escalado el resultado entre 0 y 255 , y también hemos aplicado una corrección gama de valor $2,5$. En el de la derecha, hemos utilizado una enfatización de contraste de los niveles de gris medios.



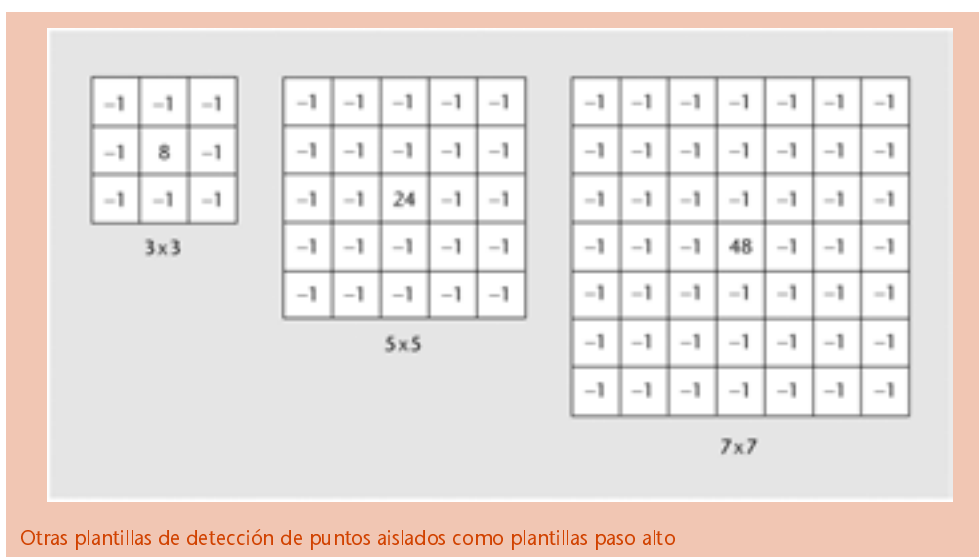
Otras plantillas para el filtrado paso alto

Utilizando el mismo proceso de sustracción entre el sistema identidad y las plantillas paso bajo podemos obtener otras plantillas alternativas para el filtrado paso alto. En la figura que vamos a analizar a continuación hallaremos las alternativas 3×3 , 5×5

y 7×7 obtenidas a partir de los filtros de promedio. Estas plantillas se conocen con el nombre de **detección de puntos aislados**, ya que producen un elevado nivel de salida en los elementos de imagen cuyo valor de gris es muy distinto del de los píxeles situados próximos a él.

Si se aplicaran estas plantillas a imágenes con ruido de tipo impulsional, podríamos detectar con facilidad los elementos de imagen que tienen este tipo de ruido, al estar enfatizando su nivel de gris (en positivo o en negativo). De entre estas plantillas, en especial se utiliza la de 3×3 por la gran simplicidad de las operaciones binarias involucradas (restar y desplazar bits).

Las plantillas de detección de puntos, aunque son un claro ejemplo de filtros paso alto por cuanto permiten el paso de puntos aislados y eliminan las regiones uniformes, no son las más utilizadas debido, fundamentalmente, a que la interpretación de sus resultados no es inmediata. En el apartado “Estimación del gradiente y estimación de contornos” siguiente examinaremos los filtros y estimadores de gradiente y derivadas de orden superior que son los más populares para la implementación de filtrados paso alto de imágenes.



Estimación del gradiente y detección de contornos



Una de las principales aplicaciones de los filtros paso alto consiste en cuantificar las variaciones de nivel gris que se producen en la superficie de la imagen. Por norma general, estas variaciones están directamente relacionadas con los contornos de los objetos o con las texturas y propiedades geométricas de los propios componentes de la imagen.

Podemos interpretar una imagen como una superficie bidimensional donde el nivel de gris nos indica el valor que toma la función en cada una de las posiciones del plano (x, y) . En la figura adjunta hemos representado las líneas con nivel de gris constante para una imagen que será fácilmente identificable para el lector (curvas isonivel).

Como en toda superficie bidimensional, es posible calcular el gradiente (que nos dará una medida de la magnitud), la dirección y el sentido en el que se producen los cambios de nivel de gris en la extensión de la superficie.

Para el caso de imágenes definidas en un espacio de variables continuas, el gradiente viene determinado por:

$$\nabla(u(x, y)) = \left[\frac{\partial u(x, y)}{\partial x}, \frac{\partial u(x, y)}{\partial y} \right]^T$$

Se trata de un vector que podemos calcular en cada punto de la imagen, que se dirige hacia la dirección de máximo crecimiento de la función y cuyo módulo ofrece una idea de la magnitud de los cambios de nivel de gris. En la siguiente figura hemos representado de forma esquemática los vectores gradiente de la imagen que hemos visto en el ejemplo anterior.



Estimación del módulo del gradiente a partir de ejes de coordenadas

El gradiente aporta información vectorial acerca de las variaciones de nivel de gris que se producen en la imagen. En muchas aplicaciones estaremos más interesados por la magnitud de las variaciones de nivel de gris que por las direcciones en las que éstas se producen. El módulo del gradiente viene determinado por:

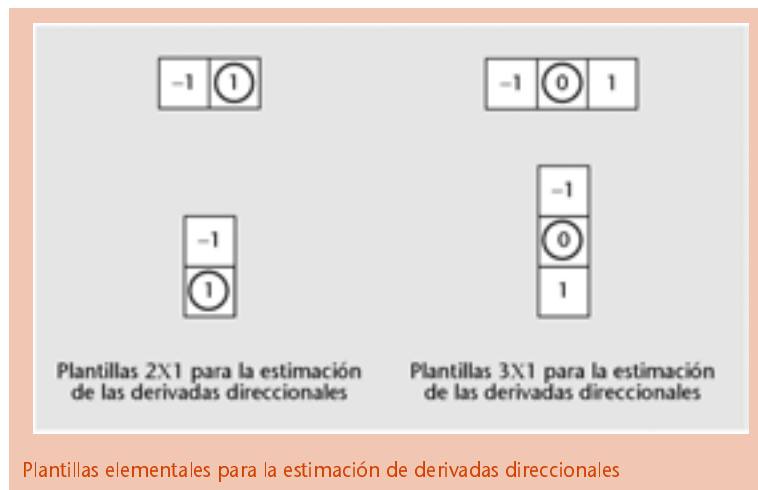
$$\|\nabla(u(x, y))\| = \left(\left(\frac{\partial u(x, y)}{\partial x} \right)^2 + \left(\frac{\partial u(x, y)}{\partial y} \right)^2 \right)^{1/2},$$

lo que supone que tendremos que estimar las derivadas parciales de la imagen con respecto a cada una de las coordenadas x y y , con posterioridad, determinar sus cuadrados, sumarlos y extraer la raíz cuadrada. Tendríamos que efectuar esta serie de operaciones para todos los píxeles de la imagen, por lo que la carga computacional asociada al cálculo del módulo del gradiente sería muy elevada. Para simplificar la estimación, el procedimiento es éste:

$$\|\nabla(u(x, y))\| = \left| \frac{\partial u(x, y)}{\partial x} \right| + \left| \frac{\partial u(x, y)}{\partial y} \right|,$$

donde en lugar de determinar el cuadrado de cada componente, tomamos su valor absoluto y suprimimos el cálculo de la raíz cuadrada. Los resultados obtenidos mediante esta aproximación suelen ser satisfactorios en la mayor parte de las aplicaciones y su implementación es directa y simple.

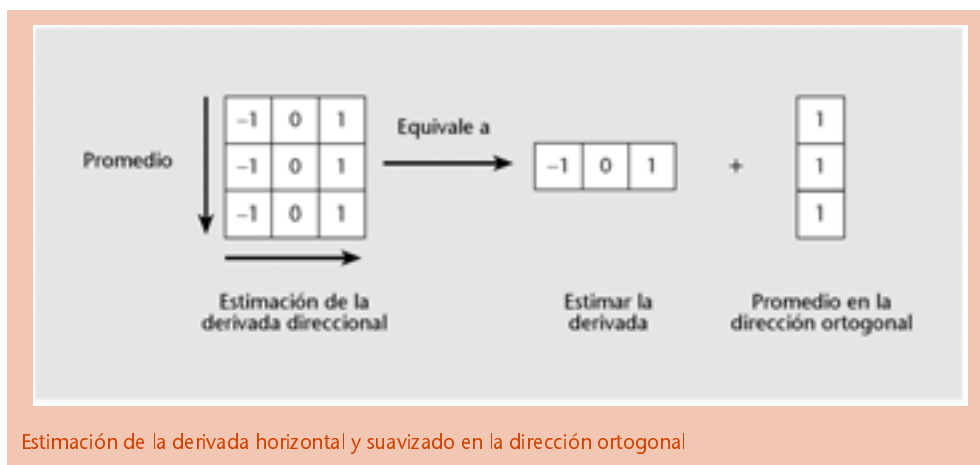
Plantillas para estimar los componentes del gradiente



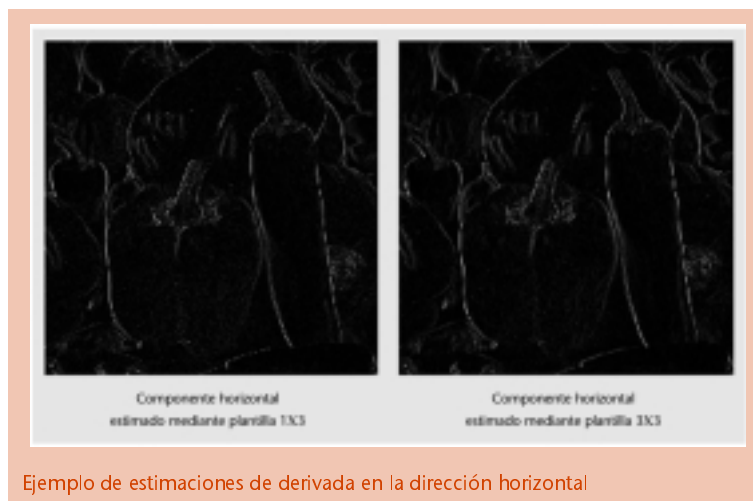
A la hora de estimar los componentes del gradiente haremos uso de una serie de plantillas que estiman la derivada en cada dirección en tanto que diferencias entre los niveles de gris de píxeles adyacentes. A continuación podemos ver una imagen donde aparecen representados dos modelos posibles para este tipo de plantillas. En el modelo de la izquierda, la derivada se estima como la diferencia entre el nivel del gris del píxel de referencia y el píxel anterior y , en el modelo de la derecha, se utiliza una pequeña variante en la que la derivada es aproximada por la diferencia entre los dos elementos de imagen que se hallen más cerca del píxel de referencia.

Las plantillas diferenciales son muy sensibles al ruido que pueda tener la imagen original, de manera que suele ser recomendable aplicar algún tipo de filtrado paso bajo a la imagen resultante. No obstante, es evidente que hay que realizar el filtrado paso bajo en la dirección ortogonal a la que estamos estimando la derivada puesto que, en caso contrario, la medida no sería correcta.

En la siguiente figura representamos una plantilla 3×3 que utiliza un filtrado paso bajo en la dirección ortogonal a la que se estima la derivada. Tal y como indicamos en la gráfica, esta plantilla se puede descomponer en dos plantillas elementales que actúan en direcciones ortogonales. La primera realiza la estimación de la derivada, mientras que la segunda promedia los resultados obtenidos en píxeles adyacentes. El lector puede comprobar que las dos alternativas son equivalentes calculando el resultado que obtenemos al aplicar la plantilla 3×3 a una imagen con todos los píxeles iguales a cero, a excepción de un único píxel central con nivel de gris unitario (la imagen resultante coincide con la propia plantilla), y verificando que coincide con el que obtenemos si aplicamos de forma secuencial las dos plantillas unidimensionales. Es evidente que el orden con el que se manejan las dos plantillas elementales puede ser intercambiado.



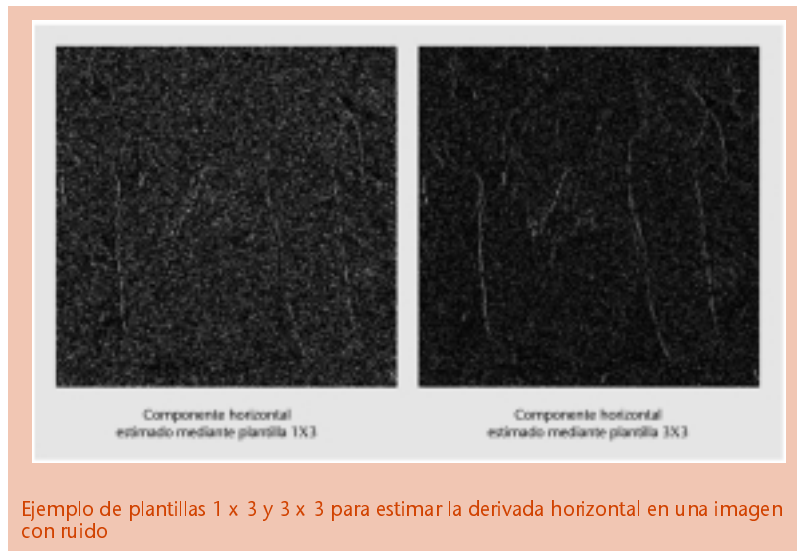
Ejemplos de estimación de gradiente: comparativa entre plantillas 1×3 y 3×3



En la siguiente figura nos disponemos a representar la estimación del componente horizontal del gradiente obtenido mediante el uso de una plantilla diferencial 1×3 y el obtenido mediante una plantilla donde se realiza promedio en la dirección vertical. Los resultados que producen ambos métodos son comparables cuando el nivel de rui-

do que presenta la imagen original es prácticamente nulo.

En la figura siguiente se presentan los mismos resultados para una imagen que contiene un cierto nivel de ruido blanco. En este segundo caso, es evidente que el promedio en la dirección ortogonal proporciona una ganancia neta con respecto al ruido.



Plantillas comunes para estimar derivadas direccionales

Componente horizontal	$\begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & 0 & 1 \\ -\sqrt{2} & 0 & \sqrt{2} \\ -1 & 0 & 1 \end{bmatrix}$	
	Componente vertical	$\begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$	$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$	$\begin{bmatrix} -1 & -\sqrt{2} & -1 \\ 0 & 0 & 0 \\ 1 & \sqrt{2} & 1 \end{bmatrix}$
		Roberts	Prewitt	Sobel	Frei-Chen

Plantillas más comunes para la estimación de los componentes del gradiente

Como resulta obvio, las posibles alternativas para conjugar la estimación de la derivada con el promedio de muestras en la dirección ortogonal es muy variado. En la figura encontramos varias de las propuestas de mayor uso expresadas como plantillas 3×3 , donde el píxel central es considerado el elemento de referencia. Entre éstas, las que se utilizan con mayor frecuencia son, probablemente, las de Roberts, Sobel y Prewitt.

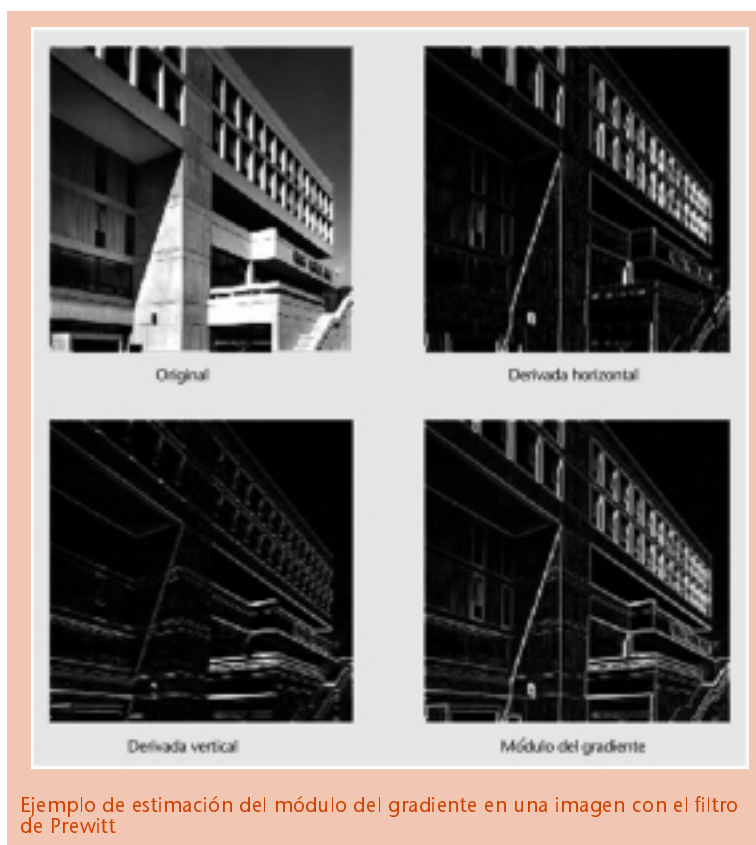
En el primer caso se trata de una variante de las plantillas de dos elementos (diferencia directa entre píxeles) que ya hemos considerado con anterioridad. Al intervenir sólo dos elementos de imagen se trata de una estimación bastante sensible al ruido, a pesar de que las direcciones en las que se estima el gradiente no coinciden con los ejes principales de la imagen. Se trata de direcciones ortogonales y, en consecuencia,

el módulo del gradiente puede seguir estimándose como la suma de los valores absolutos de cada plantilla (o como la raíz cuadrada de los cuadrados de cada componente, en caso de que queramos ser más precisos).

Las alternativas de Prewitt, Sobel y Frei-Chen se pueden identificar directamente como variantes, teniendo en cuenta la filosofía de estimar la derivada direccional en un sentido y promediar en la dirección ortogonal. En el caso de Sobel y Frei-Chen, el promediado paso bajo da una mayor importancia a los elementos centrales que a los extremos. Entre ellas, la variante de Frei-Chen supone una mayor complejidad de cálculo al requerir la realización de productos reales entre las muestras y los coeficientes, mientras que Sobel sólo precisa desplazamientos de bits. Los resultados obtenidos con estas tres variantes son comparables, pero los más utilizados son los de Prewitt y Sobel.



Ejemplos de estimación del módulo del gradiente



En la figura aparecen los resultados obtenidos para las estimaciones de gradiente en las direcciones horizontal, vertical y la combinación de ambas (suma de valores absolutos) utilizando el filtro de Prewitt. Notad, en cualquier caso, que la imagen final contiene la información de contornos de los elementos principales de la imagen.

Las imágenes de la siguiente figura comparan los resultados finales obtenidos mediante los filtros de Roberts, Prewitt, Sobel y Frei-Chen. En este ejemplo hemos representado los negativos de las imágenes con objeto de poder visualizar mejor las diferencias entre las distintas alternativas. En lo sucesivo usaremos, con frecuencia, este tipo de representación de las imágenes asociadas a los contornos.



Otras plantillas alternativas

-1	-1	-1	0	1	1	1	-1	-1	-1	0	1	1	1
-1	-1	-1	0	1	1	1	-1	-2	-2	0	2	2	1
-1	-1	-1	0	1	1	1	-1	-2	-3	0	3	2	1
-1	-1	-1	0	1	1	1	-1	-2	-3	0	3	2	1
-1	-1	-1	0	1	1	1	-1	-2	-3	0	3	2	1
-1	-1	-1	0	1	1	1	-1	-2	-2	0	2	2	1
-1	-1	-1	0	1	1	1	-1	-1	-1	0	1	1	1

Plantilla derivada horizontal
BOXCAR 7X7

Plantilla derivada horizontal
Abdou (Pirámide truncada)

Ejemplos de plantillas de 7×7 para la estimación de la derivada direccional

En esta figura presentamos algunas alternativas que involucran una mayor cantidad de elementos de imagen en la estimación. La primera de ellas es una variante del filtro de Prewitt extendida a un dominio de 7×7 elementos de imagen. Estos operadores se pueden extender a cualquier tamaño de plantilla y se conocen con el nombre de operadores boxcar.

La segunda alternativa se debe a Abdou y tiene la forma de una pirámide truncada en la que

los pesos decrecen de manera lineal con la distancia (a medida que se alejan del centro). En la gráfica hallamos la representación del estimador de derivada horizontal para una ventana de 7×7 (aunque, como es evidente, la plantilla se puede generalizar con facilidad para cualquier tamaño y dirección). En el siguiente ejemplo podemos ver el resultado de haber aplicado una plantilla de Abdou 7×7 a la imagen con ruido que hemos usado antes para comparar las plantillas 1×3 y 3×3 .



Derivada de la plantilla gaussiana

Una plantilla que suele involucrar un número considerable de elementos de imagen es la denominada DroG (*Derivative of a Gaussian*).

En este caso, obtenemos la plantilla derivando una función de gauss bidimensional con respecto a las variables en las que queremos estimar la derivada direccional. El problema se plantea inicialmente con variables de naturaleza continua y, después, se particulariza para variables discretas. Con el fin de obtener la plantilla para estimar la derivada vertical tendremos que derivar la gaussiana bidimensional respecto de la variable x . Es decir:

$$H(x, y) = -\frac{\partial}{\partial x}[g_1(x) \cdot g_2(y)],$$

donde:

$$g_1(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{1}{2} \cdot \left(\frac{x}{\sigma_1}\right)^2};$$

$g_2(y)$ se definiría de forma análoga. La derivada de esta función con respecto a la variable x es:

$$\frac{\partial}{\partial x}g_1(x) = -\frac{1}{\sqrt{2\pi\sigma^2}} \cdot \frac{x}{\sigma^2} \cdot e^{-\frac{1}{2} \cdot \left(\frac{x}{\sigma_1}\right)^2}$$

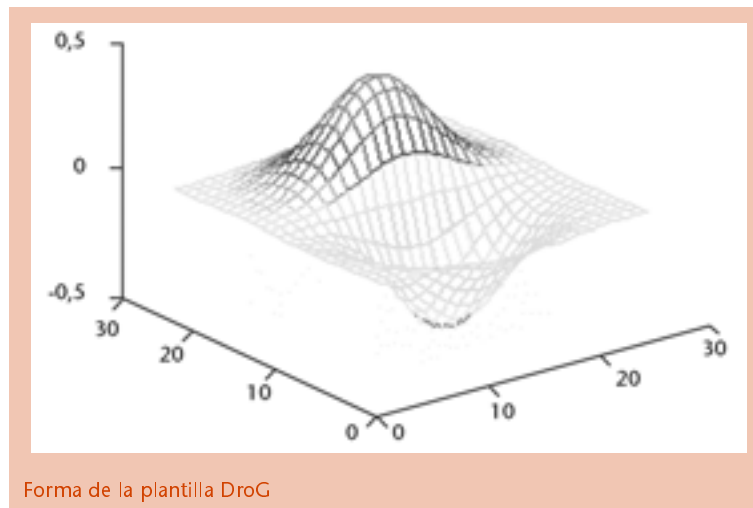
De forma que la expresión de la plantilla se puede reducir a:

$$H(x, y) = \frac{x}{\sigma} \cdot g_1(x) \cdot g_2(y)$$

Para finalizar, basta con particularizar las variables continuas en valores discretos:

$$P_{DroG}[n, m] = \frac{n}{\sigma} \cdot g_1[n] \cdot g_2[m]$$

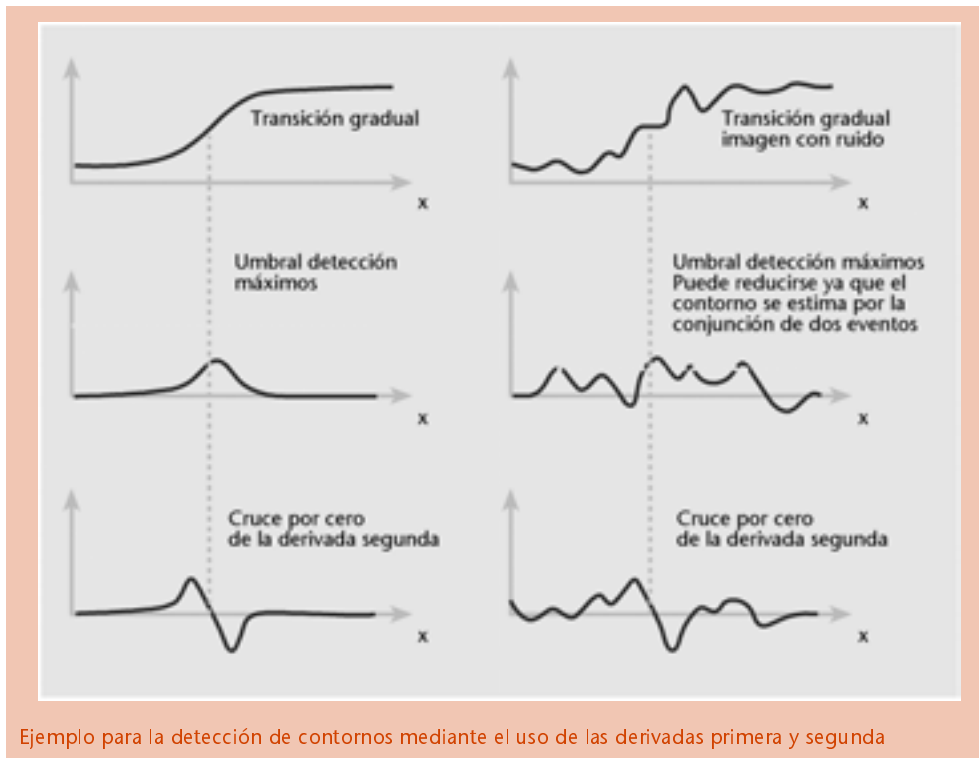
La forma de esta plantilla, que podemos ver en la figura que se sigue, tiene un aspecto parecido al del resto de las plantillas derivadas de la función gaussiana. Insistiremos en el hecho de que para la mayor parte de las aplicaciones suele ser suficiente con el uso de plantillas 3×3 o 2×2 , quedando relegadas las plantillas de tamaño superior para aquellas aplicaciones en las que el nivel de ruido es relativamente elevado.



Operadores de Laplace y cruces por cero para la estimación de contornos

La estimación del módulo o la dirección del gradiente genera resultados satisfactorios en la detección de contornos siempre y cuando los cambios de nivel de gris entre objetos o regiones sean abruptos. En el caso de transiciones suaves, el módulo de las derivadas direccionales no suele tener la energía suficiente como para activar la detección del contorno, aunque posteriormente se aplique una binarización de la imagen.

Una posible solución a este problema consiste en estimar la presencia de contornos mediante el uso de derivadas de orden superior y la detección de pasos por cero. La idea general de esta estrategia se ilustra en la siguiente figura donde, por simplicidad, sólo consideramos funciones de una variable.



Ejemplo para la detección de contornos mediante el uso de las derivadas primera y segunda

En la gráfica anterior encontramos la representación de una transición lenta entre dos regiones suponiendo que la señal tiene un nivel de ruido prácticamente inapreciable. La derivada primera de la función se encuentra inmediatamente debajo del nivel de gris y presenta, como consecuencia de la lentitud de las transiciones, amplitudes reducidas. Si el umbral de detección de contornos está situado por encima de estos niveles, no será posible detectar la transición. En cambio, en la derivada segunda de la función, hallamos un cambio de signo justo en el instante en que se produce la transición, de modo que, al menos en principio, parece posible detectar la presencia del contorno examinando las posiciones en las que se producen los cruces por cero de la segunda derivada.

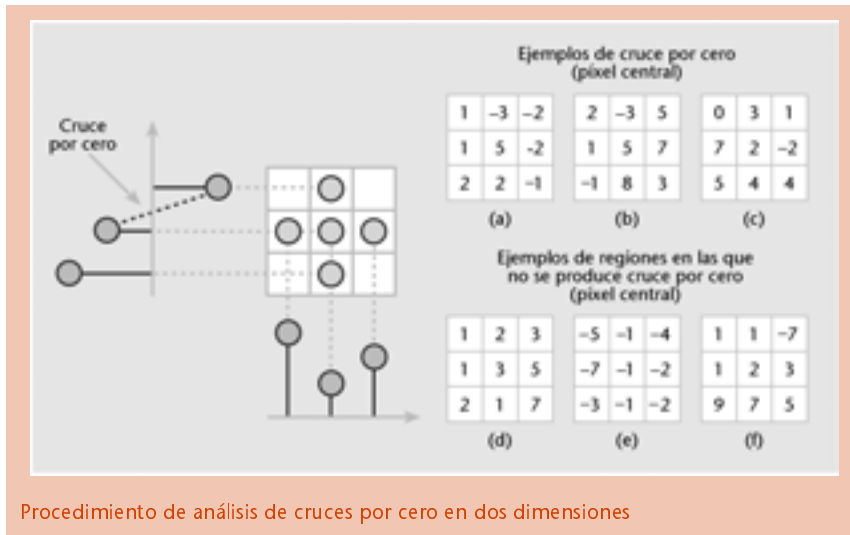
No obstante, aunque la idea general parece excelente, tendremos que refinarla debido a que los cruces por cero de la segunda derivada son muy sensibles al ruido. En la gráfica de la derecha podemos ver los resultados que obtendríamos en el supuesto de que la señal original contuviera cierto nivel de ruido. Ahora, los cruces por cero de la segunda derivada se producen en posiciones que parecen completamente aleatorias y que no tienen una relación directa con los contornos de la imagen. Podemos solucionar este problema mediante la combinación de los resultados de la primera y segunda derivada. En efecto, podemos seguir detectando la presencia del contorno mediante los cruces por cero de la segunda derivada siempre y cuando sólo se consideren aquellos cruces por cero en los que la derivada primera se mantiene por encima de un umbral prefijado. Tal y como ilustra la figura, podemos situar el umbral que delimita los valores de la primera derivada por debajo de lo habitual, ya que el contorno se caracteriza no sólo por un rebosamiento de este nivel, sino también por la aparición simultánea de un cruce por cero de la segunda derivada. La conjunción de los dos parámetros proporciona una mayor protección del sistema frente al ruido.

Cálculo de los cruces por cero en funciones de dos dimensiones

Para generalizar este procedimiento en funciones de dos variables, es preciso que tengamos en cuenta cómo se extienden los conceptos de cruces por cero y derivadas segundas para este tipo de funciones.

La forma más simple de determinar si podemos considerar un píxel como un cruce por cero de una función bidimensional consiste en examinar las transiciones que se producen

en dos direcciones ortogonales, que, en general, se toman como los mismos ejes donde representamos la imagen. Si en alguno de estos dos ejes se produce una transición de positivo a negativo o viceversa, el elemento de imagen será considerado un cruce por cero de la función. Podemos ver este procedimiento en la siguiente figura, donde mostramos algunos ejemplos que serían considerados cruces por cero y otros que no lo serían.



Procedimiento de análisis de cruces por cero en dos dimensiones

Resulta evidente que podríamos extender esta definición a un mayor número de direcciones en las que se examina el posible cambio de signo de la función, con lo que algunas regiones, como la (f) en la figura, se podrían

considerar cruces por cero. No obstante, esto no suele ser necesario puesto que, en cualquier caso, este cambio de signo ya será considerado cuando examinemos otro elemento de imagen vecino.

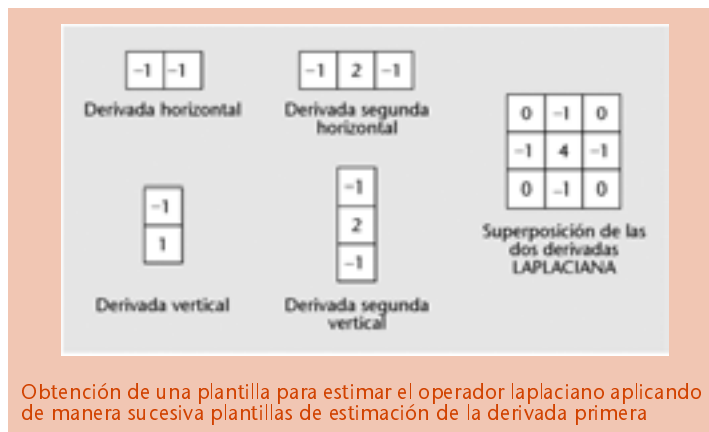
Derivada segunda en funciones de dos dimensiones

Podemos llevar a cabo la extensión de la derivada segunda a funciones bidimensionales usando diferentes alternativas que combinen las derivadas parciales de segundo orden. No obstante, la opción más empleada es, de una forma casi invariable, la laplaciana. Esta función se define como:

$$\nabla^2(u(x, y)) = \frac{\partial^2 u(x, y)}{\partial x^2} + \frac{\partial^2 u(x, y)}{\partial y^2}$$

El problema, ahora, es obtener estimadores de esta transformación en términos de plantillas. Las opciones son, como es de esperar, muy variadas y su uso dependerá, como en el resto de los operadores ya considerados, de la aplicación y las preferencias del diseñador.

En la figura adjunta mostramos un procedimiento para la obtención de la plantilla asociada a un operador del tipo laplaciano a partir de plantillas elementales para la estimación de la derivada direccional.



Obtención de una plantilla para estimar el operador laplaciano aplicando de manera sucesiva plantillas de estimación de la derivada primera

En la parte superior izquierda podemos ver el resultado de haber aplicado la plantilla para la derivada horizontal a una imagen con todos los píxeles iguales a cero menos el central (la imagen, que no se representa en la figura, corresponde a una función delta de Dirac bidimensional). Si sobre este resultado volvemos a aplicar la plantilla para la derivada horizontal, obtenemos la plantilla asociada a la derivada segunda en la dirección horizon-

tal. Se representan los mismos resultados para derivadas en el sentido vertical en la parte inferior izquierda de la gráfica.

Para finalizar, en la parte derecha, representamos la superposición de las dos derivadas de segundo orden en una imagen bidimensional. Podemos identificar la imagen resultante directamente con una plantilla bidimensional para la estimación de la laplaciana.

Ejemplo de estimación de la derivada segunda

En la siguiente figura encontramos representado el resultado obtenido al aplicar esta plantilla a una imagen. En esta representación hemos desplazado el valor medio de la imagen al nivel de gris medio, por lo que los valores positivos quedan representados entre el nivel de

gris 128 y 255, mientras que los negativos, entre 0 y 127. Observad que las regiones que se corresponden con fuertes transiciones de la imagen original se sitúan en zonas donde el nivel de gris pasa de blanco a negro o viceversa, indicando la presencia de un cruce por cero.

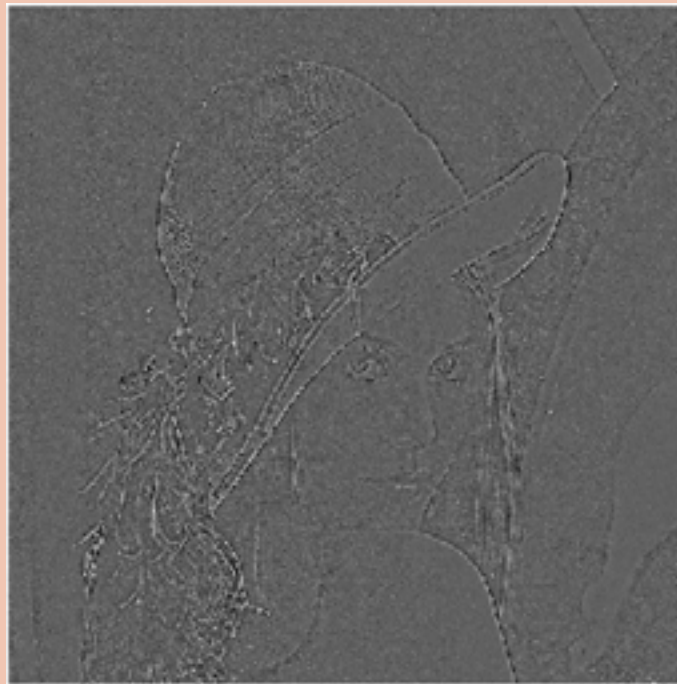


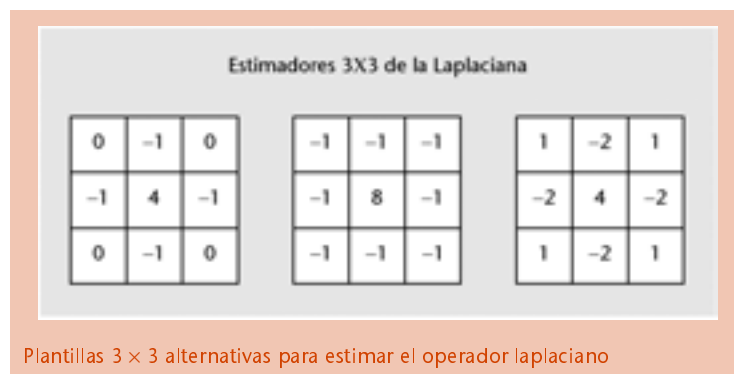
Imagen obtenida al aplicar el operador laplaciano



Otras plantillas de uso habitual

Como es de esperar existen otras plantillas alternativas para estimar la laplaciana. En la si-

guiente figura podemos ver otras dos opciones 3×3 que se utilizan con cierta frecuencia.





Laplaciana de una gaussiana

También es posible definir una plantilla para aproximar el operador laplaciano que toma como base la función de gauss bidimensional. Esta plantilla se conoce con el nombre de LoG

$$L(x, y) = \frac{\partial^2}{\partial x^2}(g_1(x) \cdot g_2(y)) + \frac{\partial^2}{\partial y^2}(g_1(x) \cdot g_2(y)) = g_2(y) \cdot \left(\frac{\partial^2}{\partial x^2} \cdot g_1(x)\right) + g_1(x) \cdot \left(\frac{\partial^2}{\partial y^2} \cdot g_2(y)\right),$$

donde, como ya hemos visto antes, $g_1(x)$ se define como:

$$g_1(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot e^{-\frac{x^2}{2\sigma^2}},$$

y $g_2(y)$ se define de modo análogo.

Determinando la derivada segunda con respecto a x de $g_1(x)$, obtenemos:

$$\frac{\partial^2}{\partial x^2}(g_1(x)) = \frac{1}{\sqrt{2\pi\sigma^2}} \cdot \frac{1}{\sigma^2} \cdot \left(\frac{x^2}{\sigma^2} - 1\right) \cdot e^{-\frac{x^2}{2\sigma^2}}$$

Si sustituimos este resultado y su equivalente para $g_2(y)$, el resultado es:

$$L(x, y) = C \cdot \left(2 - \frac{x^2 + y^2}{\sigma^2}\right) \cdot e^{-\frac{x^2 + y^2}{2\sigma^2}},$$

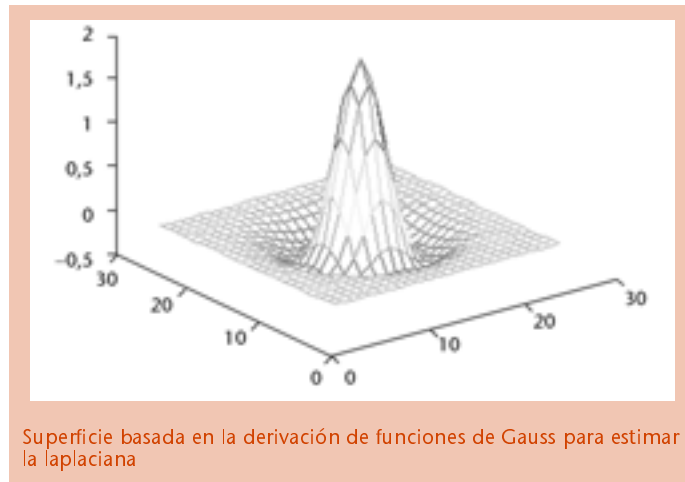
donde hemos supuesto que las desviaciones típicas en las dos variables son idénticas y hemos dejado una constante de ganancia genérica C .

(*Laplacian of a Gaussian*). Con este objetivo, calcularemos la laplaciana de esta función y, con posterioridad, sustituiremos las variables continuas por discretas. Así pues:

En caso de que particularicemos este resultado para variables discretas, obtenemos la expresión general de una plantilla para la estimación de la laplaciana que se basa en funciones gaussianas:

$$P_{Lap}[n, m] = C \cdot \left(2 - \frac{n^2 + m^2}{\sigma^2}\right) \cdot e^{-\frac{n^2 + m^2}{2\sigma^2}},$$

donde tendremos que determinar C deberá para normalizar que la suma de todos los coeficientes que intervienen en la plantilla sea igual a la unidad. Podemos ver la forma de la esta plantilla en la figura adjunta, donde hemos tomado, para facilitar la representación tridimensional, una ventana de 25 x 25 elementos de imagen y una desviación típica de valor 3. La superficie pondera los píxeles situados en la proximidad del centro con valores positivos que decrecen rápidamente a medida que nos alejamos, hasta convertirse en negativos a una cierta distancia. Esta superficie recibe el nombre de sombrero mejicano por la forma especial en que pondera los píxeles de la imagen.

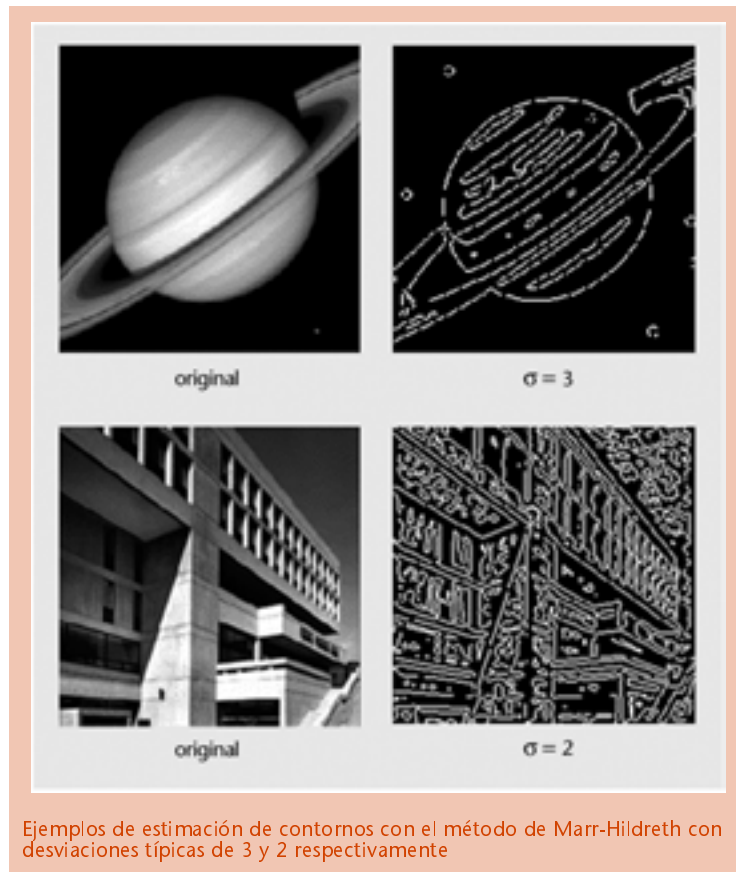


Ejemplos de detección de contornos sobre la base de la plantilla LoG y los cruces por cero

Marr-Hildreth fue quien propuso la detección de contornos basada en el uso de una plantilla LoG, así como el análisis de los cruces por cero. Las estimaciones obtenidas dependen de un único parámetro que es la desviación típica de la función gaussiana. Cuando la desviación típica es grande, intervienen un elevado número de píxeles en la

estimación de la derivada segunda, con lo que el ruido tiene poco efecto y no es necesario considerar los valores que toma la derivada primera.

En las figuras podemos ver algunos ejemplos de estimaciones de contornos que utilizan este procedimiento.



Filtros paso alto para el realce de contornos

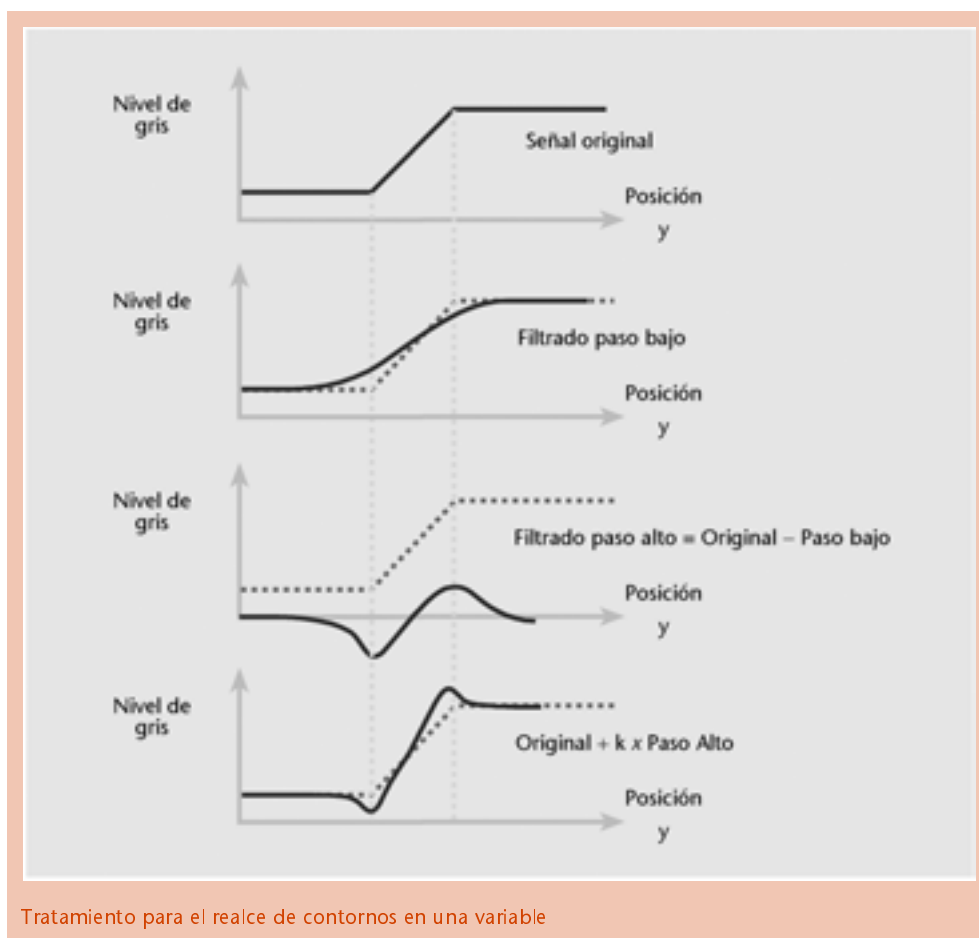
El resultado de un filtrado paso alto se puede combinar con la imagen original para obtener una mejora en la nitidez aparente gracias a un realce de los contornos. Una posible forma de implementar este realce consiste en añadir a la imagen original una versión filtrada paso alto de la misma:

$$w[n, m] = u[n, m] + \lambda \cdot u_{HP}[n, m],$$

donde $u[n, m]$ representa la imagen original, $u_{HP}[n, m]$, la versión paso bajo y $w[n, m]$, la imagen resultante; el parámetro λ controla el factor en el que se pondera la señal paso alto con respecto a la original.

En general, los filtros empleados para realizar esta operación son variantes de los filtros paso alto definidos como la diferencia entre la imagen original y un filtro paso bajo. También se utilizan, a menudo, algunas aproximaciones a la laplaciana.

Para comprobar que la superposición de una señal paso alto con la original produce transiciones más bruscas en los contornos de la imagen, con la consiguiente mejora de su percepción, es útil considerar la siguiente gráfica en la que ilustramos esta combinación con señales unidimensionales.



La señal original podría representar el nivel de gris de la imagen a lo largo de una línea. La transición entre los dos niveles es suave, así que la delimitación del contorno no será clara. Justo debajo de la señal original se representa la versión paso bajo, en la cual las transiciones entre los dos niveles de gris son todavía más lentas. La diferencia entre los dos componentes coincide con la señal original filtrada paso alto.

Es evidente que, tal y como vemos en el gráfico, si le añadimos este componente a la señal original obtenemos una transición más rápida entre los dos niveles. Además, notad que el paso de un nivel de gris a otro también se ve aumentado, ya que antes de la transición se produce una pequeña inversión en la dinámica de la señal. Todos

estos factores ayudan a que, desde un punto de vista subjetivo, la imagen aparente una mayor definición con motivo de que el sistema visual humano es capaz de localizar las transiciones con una mayor precisión.



Ejemplo de realce de contornos



En la figura podemos apreciar el resultado de haber aplicado un filtrado para el realce de contornos. En este ejemplo, la imagen original está enfocada de forma correcta, por lo que el realce de los contornos simplemente enfatiza los cambios de nivel de gris, produ-

ciendo, de esta manera, la sensación de una mayor definición y nitidez. No obstante, si el realce de los contornos se exagera, las imágenes pueden llegar a parecer extrañas y con falta de realismo.

Etapa 2: Transformaciones espaciales no lineales y procesamiento morfológico

Introducción

Los filtros no lineales proporcionan herramientas adicionales para la mejora de imagen que, por norma general, facilitan la supresión de algunos tipos particulares de ruido. Son útiles en especial cuando el ruido está distribuido de forma muy local en algunas zonas de la imagen. Un ejemplo de este tipo de ruido son las manchas de polvo o los arañazos y rasguños que aparecen en fotografías antiguas; también resultan útiles para combatir ruidos de tipo impulsivo o impulsional. El ruido impulsional aparece en una imagen digital como consecuencia de errores en la transmisión o almacenamiento de la misma. El efecto de estos errores es poco notorio si sólo afecta a los bits menos significativos, pero puede ser muy molesto si afecta a los bits más significativos. En este último caso, el resultado es que algunos píxeles aislados aparecen con niveles de gris completamente distintos a los de los elementos situados en su proximidad. El ruido impulsional también puede aparecer en imágenes analógicas como consecuencia de interferencias electromagnéticas debidas a la conmutación de máquinas.

Los filtros no lineales resultan útiles para eliminar manchas y ruidos locales en la imagen.

Del mismo modo, los filtros no lineales se utilizan como procesamiento previo en operaciones de segmentación de la imagen; las operaciones más elementales son la dilatación y la erosión.

Los filtros no lineales que estudiaremos en este módulo están basados en operaciones de ordenación de los píxeles en el interior de una determinada región. Las operaciones de ordenación generalmente consisten en sustituir el valor del píxel sobre el que se aplica el proceso por el valor máximo, mínimo o la mediana de los elementos de imagen que se encuentran dentro del entorno de análisis.

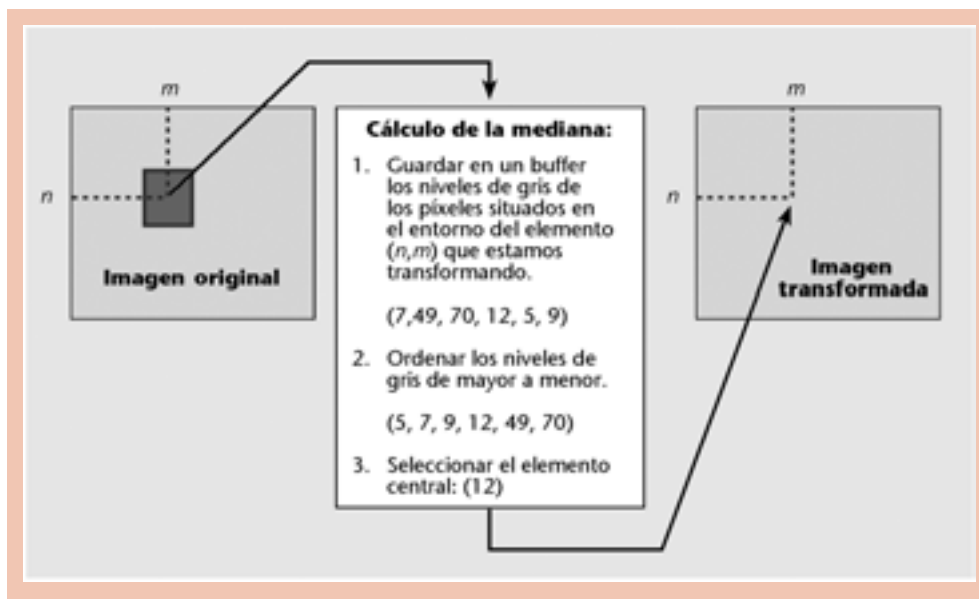
Filtros de mediana

Los filtros de mediana son, probablemente, los filtros no lineales utilizados con mayor profusión en el tratamiento de señales unidimensionales e imágenes. Se trata de

filtros con una implementación relativamente sencilla que consiguen eliminar ruidos de tipo impulsional con gran eficiencia y sin afectar de manera excesiva a la calidad de la señal.

En la figura podemos ver el principio de funcionamiento básico de este tipo de filtros en el contexto de procesado de imagen. El valor del píxel con ordenadas (n, m) de la imagen transformada se obtiene aplicándole una rutina de ordenación a los píxeles situados en un entorno del elemento (n, m) de la imagen original. Tal y como aparece indicado en el gráfico, los elementos situados en el interior de la plantilla de la imagen original se ordenan de mayor a menor y se selecciona el elemento cuyo nivel de gris está situado en la posición central. Esta posición central corresponde a la mediana estadística; es decir, aquel nivel de gris tal que existen tantos elementos con un nivel de gris superior como inferior a él.

Es importante no confundir los filtros de mediana con los de valor medio. En el primer caso, sustituimos el elemento central por el valor que queda en el centro tras haber ordenado todos los píxeles de forma correcta, mientras que en el segundo, lo sustituimos por el valor promediado entre todos los píxeles que entran en la plantilla. El resultado final puede ser muy distinto en función de las condiciones de la imagen.



De manera analítica:

$$v[n, m] = \text{mediana}\{u[i, j]; (i, j) \in I\}$$

La mediana estadística es una operación muy adecuada para extraer ruido de tipo impulsional en una imagen, sin que la definición de la misma se vea prácticamente afectada.



Ejemplo de filtro de mediana con ruido impulsional



Ejemplo de un filtro de mediana aplicado a una imagen con ruido impulsional

En la figura anterior aparece una imagen afectada por ruido impulsional y el resultado obtenido al aplicar un filtro de mediana con una plantilla de 3×3 a la imagen. La mejora conseguida es más que evidente, puesto que, como podemos ver, ha desaparecido prácticamente la totalidad del ruido. La justificación intuitiva está, también, clara: los elementos con ruido tienen un nivel de gris muy distinto del de sus vecinos, por lo que, al ordenarlos, éstos siempre se sitúan en los extremos de la lista (los más pequeños o los mayores) de manera que en la imagen resultante siempre se eliminan y sustituyen por el nivel de gris de un píxel vecino cuyo

valor es más probable, desde el punto de vista estadístico, y menos molesto, desde el punto de vista visual. El filtro de mediana es útil para extraer líneas o píxeles aislados manteniendo una buena resolución espacial. No obstante, cuando el número de elementos con ruido es comparable o superior a la mitad del número de píxeles de la ventana, los resultados suelen ser poco beneficiosos. Asimismo, si se aplican a una imagen con ruido blanco, obtenemos unos resultados bastante pobres sin que se produzcan, por ello, reducciones significativas del nivel de ruido.



Coste computacional

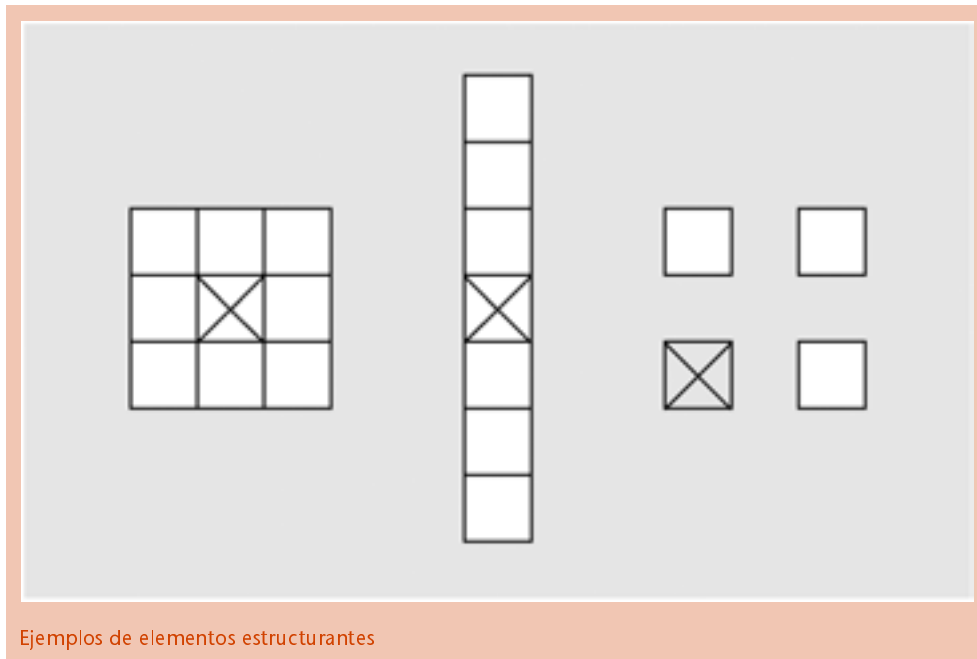
El coste computacional del filtro de mediana está directamente relacionado con la eficiencia del algoritmo de ordenación. Los más eficaces pueden ordenar un conjunto de M números realizando un número de comparaciones situado en torno a $(1/2) M \log M$. Es decir, que para ordenar una ventana de 5×5 (25 elementos de imagen), sólo

es necesario que realicemos unas 18 comparaciones. Pero, todavía podemos mejorar más estos algoritmos si tenemos en cuenta que al desplazar la ventana sólo se introducen algunos elementos nuevos, mientras que otros desaparecen, de modo que los que se mantienen ya han sido ordenados de forma previa.

Elemento estructurante

El entorno de píxeles procesados para determinar la mediana es, en la mayor parte de los casos, una matriz cuadrada de 3×3 o 5×5 píxeles. No obstante, el entorno de píxeles que se tienen en cuenta en la ordenación puede tener cualquier otra forma, aunque, en general, la forma de este entorno se define mediante una plantilla que llamamos elemento estructurante.

En la figura hallamos una definición gráfica de varios elementos estructurantes de uso habitual. La cruz indica el elemento central sobre el que se aplica la plantilla y los elementos en gris, los píxeles que formarán parte del entorno. El elemento estructurante de la izquierda corresponde a un entorno de 3×3 píxeles que se aplica sobre el elemento central y coincide con el que hemos utilizado en el ejemplo anterior. El elemento estructurante representado en el centro de la figura sólo tiene en cuenta a los vecinos ubicados en la línea vertical del elemento central. Es un elemento estructurante que, como veremos, resulta útil para la extracción de ruido o manchas horizontales en una imagen. El elemento estructurante de la izquierda se utiliza poco en la práctica y únicamente se representa para insistir en el carácter general que tiene la definición del entorno del proceso. En este ejemplo sólo vamos a tener en cuenta a tres vecinos que, además, se encuentran alejados del píxel central (el propio píxel central no se tiene en cuenta).



Ejemplo con elemento estructurante vertical



Resultado de haber aplicado un filtro de mediana con elemento estructurante vertical para eliminar manchas horizontales

En el ejemplo que se sigue podemos ver el resultado de haber aplicado un elemento estructurante vertical a una imagen para suprimir una mancha horizontal presente en la fotografía original. El elemento estructurante aplicado es una plantilla de 11×1 píxeles que aplicamos sobre el elemento central. Observad cómo el filtrado ha suprimido la mancha horizontal y la definición de la imagen se ha visto algo reducida.

Para reducir este tipo de ruido en fotografías antiguas (rayas y manchas) podemos utilizar filtros de mediana con elementos estructurantes verticales y horizontales. La estrategia de proceso más utilizada consiste en aplicar, en primer lugar, uno de los filtros y sobre el resultado obtenido, el otro filtro. Mediante este proceso, primero eliminamos las manchas en una dirección y después, en la otra. También es habitual utilizar filtros de mediana con elementos estructurantes en forma de cruz.

Filtros de valor máximo y de valor mínimo



Una variante de los filtros de mediana consiste en seleccionar el valor máximo o el mínimo después de la ordenación. Estas variantes se utilizan, sobre todo, para eliminar algunos componentes de la imagen cuyo nivel de gris está situado por encima o por debajo del entorno global y constituyen la base de las operaciones morfológicas. En la figura adjunta observamos el resultado de aplicar un filtro de valor máximo a una de las imágenes con ruido impulsional.

La complejidad computacional asociada a los filtros de valor máximo y mínimo es considerablemente inferior a la de los filtros de mediana. En efecto, para encontrar el máximo o mínimo en la región de análisis no es necesario llevar a cabo la ordenación de todos los píxeles, puesto que, con el sólo hecho de inspeccionar los valores de todos los elementos y seleccionar aquel que toma el valor máximo o mínimo será suficiente.

El efecto del filtro de valor máximo parece más que evidente: cada vez que en el interior del elemento estructurante nos encontramos con un píxel blanco, el resultado obtenido es un píxel blanco, lo cual querrá decir que el píxel blanco se expande en toda la región de interés delimitada por el elemento estructurante. Esta

operación se conoce, de manera alternativa, con el nombre de dilatación y resulta útil en especial para la mejora de imágenes binarias y en aplicaciones de segmentación. Así pues, las operaciones de dilatación y erosión constituyen la base de los métodos de procesado morfológico.

Procesado morfológico

El procesado morfológico utiliza herramientas de álgebra no lineal y opera con conjuntos de puntos teniendo en cuenta su conectividad y su forma. Las operaciones morfológicas se utilizan principalmente para la simplificación de imágenes y para la cuantificación de las formas características de los objetos. La matemática involucrada en esta familia de métodos es sustancialmente diferente de la utilizada con los métodos lineales, la convolución y el cálculo.

Las principales aplicaciones del procesado morfológico son:

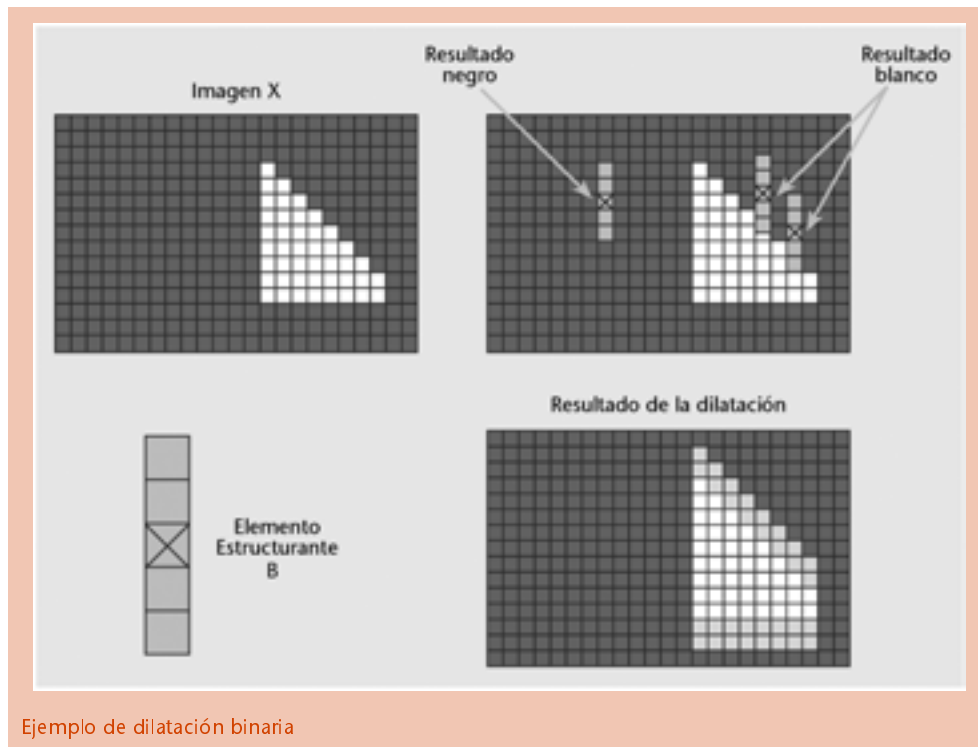
- filtrado de ruidos;
- simplificación de formas;
- análisis de estructuras de objetos: esqueletos, adelgazamiento, marcaje de objetos, etc.;
- segmentación de objetos con respecto al fondo;
- descripción cuantitativa de los objetos: área, perímetro, proyecciones, etc.

Una descripción detallada de los principales elementos del tratamiento morfológico de imágenes resulta compleja y escapa de los propósitos de este texto. En consecuencia, únicamente presentaremos algunos de los procedimientos más elementales y sugeriremos algunas de las posibles aplicaciones.

Dilatación y erosión en imágenes binarias

Las operaciones de dilatación y erosión son equivalentes a los filtros de valor máximo y valor mínimo, respectivamente. Cuando las imágenes son de naturaleza binaria, la interpretación de las operaciones de dilatación y erosión es bastante intuitiva.

Dilatación binaria



Ejemplo de dilatación binaria

Consideremos como ejemplo el proceso de dilatación. En la figura se muestra una imagen binaria X y un elemento estructurante B . Para determinar el resultado de la dilatación es necesario desplazar el punto de aplicación del elemento estructurante mediante toda la imagen binaria. El resultado será igual a la unidad (blanco), siempre y cuando algún píxel del elemento estructurante entre en contacto con un elemento de la imagen binaria que tenga un valor igual a la unidad (blanco). En caso contrario, el resultado es negro. En efecto, tened en cuenta que la dilatación consiste en tomar el valor máximo de todos los píxeles que están incluidos dentro de la estructura de la imagen B . Cuando la estructura no está en contacto con la zona blanca de la imagen, todos los píxeles interiores son negros, por lo que el máximo sigue siendo negro. En cambio, cuando algún elemento de la estructura está en contacto con la región blanca de la imagen, el valor máximo también será blanco.

Desde un punto de vista intuitivo, podemos decir que el resultado de la dilatación será cualquier conjunto de puntos en el que exista algún tipo de contacto (intersección distinta de cero) entre el elemento estructurante y la imagen binaria. En la figura del ejemplo, los píxeles que se le han añadido a la imagen original como resultado de la dilatación aparecen representados en gris claro (en realidad son píxeles blancos). Observad cómo la imagen resultante se “dilata” en la misma dirección y tamaño que el elemento estructurante, de manera que, como vemos, el elemento estructurante permite controlar el factor de dilatación y la dirección en la que ésta se produce.

Desde un punto de vista más formal, una descripción analítica de la dilatación suele ser ésta:

$$X \oplus B = \max\{x[i, j], \quad \forall (i, j) \in B_{nm}\}$$

donde B_{nm} representa el elemento estructurante desplazado hasta la posición del elemento de imagen de la fila n y columna m .



Ejemplos de dilatación

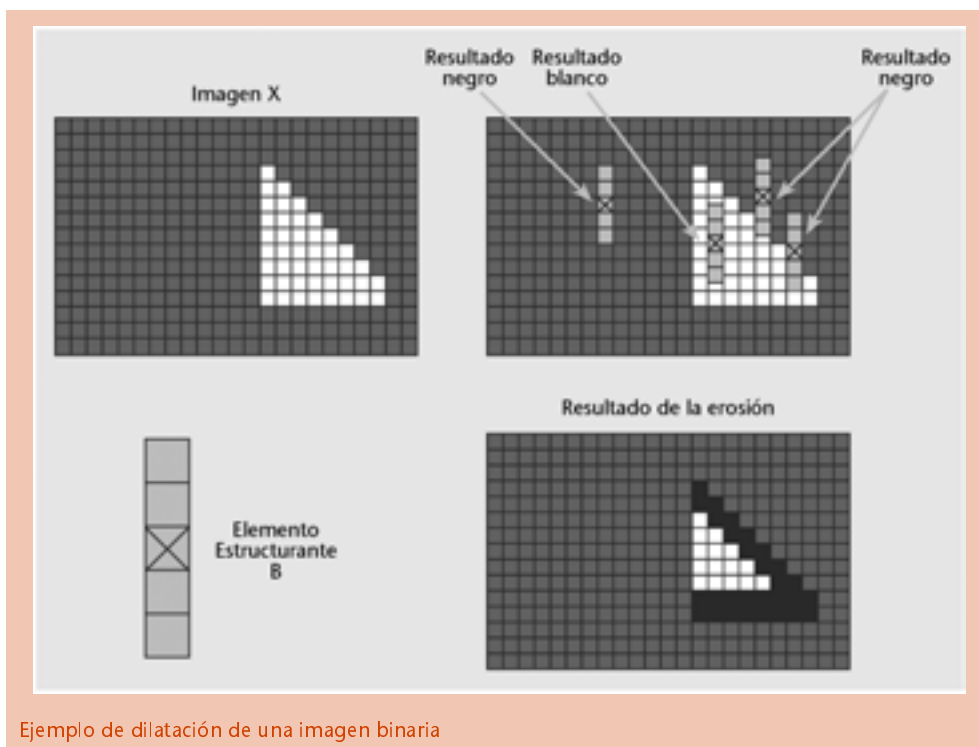


Ejemplo de dilatación de una imagen binaria

En la figura podemos ver el resultado de una dilatación aplicada a una imagen binaria. El elemento estructurante utilizado es una matriz cuadrada de 13×13 . Observad que, como consecuencia de una incorrecta binarización de la imagen original, algunos dígitos de la

imagen binaria aparecen cortados o con ruido en su interior. La dilatación produce una expansión sistemática de los dígitos de forma que, finalmente, se conectan las regiones disjuntas y desaparece el ruido del interior de los dígitos.

Erosión binaria



Ejemplo de dilatación de una imagen binaria

Podemos interpretar la erosión de un modo parecido a la dilatación mediante la superposición de las regiones correspondientes a la imagen binaria y el elemento estructurante. En este caso, el resultado de la erosión es un píxel blanco, pero sólo si el elemento estructurante se superpone por completo a la región blanca de la imagen. En efecto, sólo en este caso, todos los píxeles considerados por el elemento estructurante son blancos y, en consecuencia, el mínimo de todos ellos también es blanco. En cualquier otra situación en la que el elemento estructurante no se superponga totalmente a la zona blanca de la imagen, el valor mínimo será un píxel negro. En la figura siguiente hallamos la representación del resultado de una erosión (hemos utilizado la misma imagen y elemento estructurante que en un ejemplo anterior).

Ejemplo de erosión binaria

Podemos formular la erosión de manera analítica del mismo modo que lo hacemos con la dilatación, sin más que sustituir la operación de buscar el máximo por la de buscar el mínimo.

$$X \ominus B = \min\{x[i, j], \forall (i, j) \in B_{mm}\}$$

Las operaciones de dilatación y erosión con imágenes binarias tienen una complejidad computacional inferior a las operaciones de máximo y mínimo (dilatación y erosión con imágenes de nivel de gris). Esto es así debido a que, en la mayoría de los casos, no es necesario inspeccionar los valores de todos los píxeles en el interior del elemento estructurante. Cuando estamos realizando una operación de dilatación podemos obtener la búsqueda siempre que encontremos un píxel que toma el valor unidad (seguro que es el máximo). Por tanto, sólo en caso de que todos los píxeles sean iguales a cero es necesario completar la búsqueda de valores mediante toda la extensión del elemento estructurante. De manera análoga, en operaciones de erosión podemos determinar el análisis siempre que se encuentre un píxel con valor igual a cero (seguro que es el mínimo).



Ejemplo con erosión



En este ejemplo consideramos que la imagen original también se ha binarizado de forma incorrecta, pero ahora con un umbral demasiado reducido. El problema principal es la aparición de abundante ruido en la binariza-

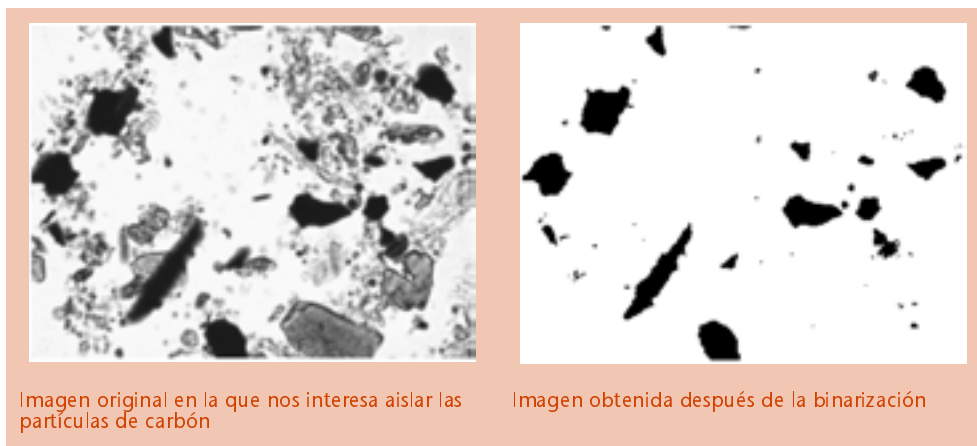
ción (presencia de píxeles blancos en regiones que deberían ser negras). Como podemos comprobar, la erosión de la imagen elimina este tipo de ruidos.

Combinando las operaciones de dilatación y erosión

La erosión es una operación dual a la de dilatación, lo cual significa que las dos operaciones son totalmente equivalentes si intercambiamos el papel de los ceros y los unos. En efecto, podemos interpretar una dilatación de esta manera: los píxeles blancos se expanden de acuerdo con la geometría del elemento estructurante. Esta dilatación de los niveles más claros se puede interpretar como una “erosión” de los más oscuros y viceversa.

La combinación de las operaciones de erosión y dilatación es particularmente interesante para reducir la presencia de ruido en imágenes de naturaleza binaria. Para ilustrar cómo podemos combinar ambas operaciones, vamos a considerar el ejemplo de la figura siguiente.

La imagen original corresponde a una muestra de sedimentos extraídos del fondo de un lago, observados mediante un microscopio. El objetivo es separar las partículas de carbón (las más oscuras y con un cierto tamaño mínimo) del resto de los elementos. La opción más evidente parece la de pasar la imagen a blanco y negro y, con posterioridad, binarizar el resultado utilizando un umbral adecuado que permita separar las partículas de carbón del resto de los elementos.



El resultado que obtenemos después de la binarización es bastante correcto. No obstante, aparecen partículas negras de pequeño tamaño que desearíamos eliminar ya que no se corresponden con las partículas de carbón. Una posible opción consiste en aplicar una dilatación de la imagen, con lo que estas partículas serán absorbidas por el entorno blanco del fondo. Sin embargo, también aplicaremos el proceso de dilatación a las partículas de carbón, de manera que reduciremos su tamaño. Este efecto tiene su

importancia, puesto que el resultado final que deseamos obtener es medir el área de las partículas de carbón.

La solución al problema consiste en aplicar, en primer lugar, una dilatación y posteriormente una erosión, mediante el uso, en ambas operaciones, del mismo elemento estructurante. Con la primera operación eliminamos las partículas de ruido y aunque reducimos el tamaño de las partículas de carbón reales, éstas recuperarán su tamaño original (al menos de forma aproximada) mediante la posterior operación de erosión. Observad que las partículas de tamaño reducido desaparecen por completo con la primera operación de dilatación, con lo que ya no volverán a aparecer.



Operaciones de *opening* y *closing*

Las operaciones de erosión y dilatación no son inversas una con respecto a la otra y su concatenación puede producir resultados distintos en función del orden en el que se aplican. La operación conjunta de dilatar y, más tarde, erosionar una imagen utilizando el mismo elemento estructurante ha sido analizada en el subapartado “Combinando las operaciones de dilatación y erosión”. Esta operación recibe el nombre de *closing*. Si intercambiamos el orden de las operaciones, es decir, primero erosionamos y después dilatamos, obtenemos una operación que recibe el nombre de *opening*.

Las operaciones de *opening* y *closing* son duales si intercambiamos los píxeles blancos por los negros y viceversa.

En una imagen donde los objetos son blancos con fondo negro, la operación de *closing* (primero dilatar y luego erosionar) tiene los siguientes efectos:

- Desaparece el ruido (puntos negros) que pudiera existir en el interior de los objetos.
- Conecta los objetos que están muy próximos.
- Suaviza la forma de los contornos de los objetos.
- Mantiene los objetos pequeños (blancos) sobre el fondo (negro).

La operación de *opening* (primero erosionar y después dilatar) tiene los siguientes efectos:

- Elimina los objetos pequeños (blancos) sobre el fondo (negro).
- Mantiene el ruido (puntos negros) que aparece en el interior de los objetos.
- Mantiene, sin conectar, los objetos que están muy próximos.
- Suaviza la forma de los objetos.

Teniendo en cuenta la dualidad entre estas dos operaciones, los efectos de una u otra operación se intercambian cuando los objetos de interés están formados por píxeles negros en un fondo blanco. Así, si recordáis el caso del subapartado “Combinando las operaciones de dilatación y erosión”, se trataba de una operación de *closing*, aunque los efectos obtenidos son los que hemos descrito para la operación de *opening*. En aquel caso, los objetos de interés estaban formados por objetos oscuros sobre un fondo claro.

Gradiente morfológico

Otra operación de interés que podemos obtener a partir de la combinación de los resultados de una dilatación y una erosión es el gradiente morfológico. Una forma de definir el gradiente morfológico es mediante la diferencia entre el resultado de una operación de dilatación y una de erosión:

$$G[n, m] = (X \oplus B) - (X \ominus B)$$

Consideremos como ejemplo la imagen de la figura siguiente. Si realizamos una operación de dilatación, todos los objetos blancos se expanden en el fondo negro. Cuando aplicamos la operación de erosión, es el fondo el que se expande hacia los objetos. La diferencia entre ambas operaciones marcará la región que se ha expandido y contraído en las dos operaciones y que se corresponde con el contorno de los objetos.



Ejemplo de gradiente morfológico en imágenes binarias

Gradiente morfológico en imágenes de nivel de gris



Los resultados obtenidos en el apartado “Gradiente morfológico” también pueden generalizarse para imágenes multinivel. En este caso, la dilatación y la erosión se corresponden con los filtros de valor máximo y mínimo que habíamos descrito en apartados anteriores. Las imágenes muestran algunos resultados de gradientes morfológicos obtenidos mediante las diferencias entre las operaciones de dilatación y erosión en imágenes con varios niveles de gris.

Etapa 3: Transformaciones geométricas

Introducción

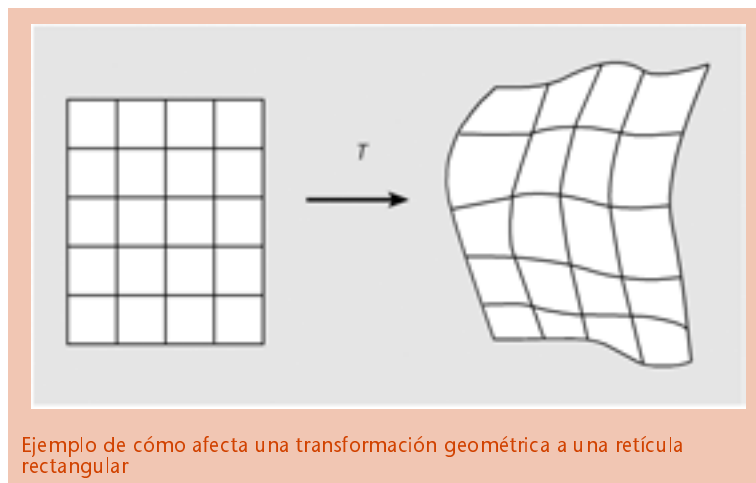
Las transformaciones geométricas en dos dimensiones se utilizan para modificar la estructura reticular de la imagen. Son de uso común en el de tratamiento de gráficos por ordenador y en el desarrollo de efectos especiales en imágenes.

Las aplicaciones más comunes en las se requiere el uso de transformaciones geométricas son:

- cambio de tamaño de la imagen: interpolación, diezmado y *zoom*;
- corrección de distorsiones del sistema de captura;
- corrección de distorsiones ópticas;
- corrección de no linealidades en los espejos del escáner;
- corrección de desalineaciones de página en OCR;
- efectos de distorsión de la imagen (*goo*, deformaciones puntuales y esféricas);
- efectos de metamorfosis entre imágenes (*morphing*).

Una transformación geométrica es una aplicación $T()$ que asigna una nueva posición (x', y') a cada elemento (x, y) de la imagen original.

El efecto de la transformación geométrica es un cambio en la posición de los elementos de imagen. En la figura siguiente, podemos ver a modo de esquema cómo puede afectar una transformación de este tipo a una retícula rectangular.



La transformación geométrica se define como una función bidimensional de dos variables reales:

$$(x'_k, y'_k) = T(x_k, y_k)$$

Una forma directa de realizar la transformación geométrica consiste en explorar toda la imagen original, es decir, todas las posiciones de los elementos de imagen (x_k, y_k) y determinar las posiciones (x', y') a las que se tiene que desplazar cada elemento de imagen.

Esta estrategia básica tiene algunos problemas que consideraremos con detalle en los siguientes apartados, pero define la esencia fundamental de una transformación geométrica. El tipo de aplicación que utilicemos determinará las características finales de la transformación y el cambio geométrico que se aplica.



Ejemplo: corrección de distorsiones ópticas



Corrección de distorsiones ópticas mediante transformaciones geométricas

En la figura hallamos una posible deformación de una imagen debida al uso de un objetivo con distancia focal corta (gran angular). Observad cómo los dos edificios tienden hacia el centro de la fotografía a medida que aumenta la altura. Mediante el uso de transformaciones geométricas podemos corregir este efecto, distorsionando la retícula de la imagen original para que los dos edificios aparezcan paralelos.

La transformación geométrica empleada en la imagen anterior se denomina corrección de perspectiva y podemos definirla de forma analítica mediante las siguientes ecuaciones:

$$x' = x_0 + (x - x_0)(1 + \alpha y)$$

$$y' = y$$

En estas ecuaciones hemos supuesto que la imagen se referencia mediante unos ejes x (horizontal) e y (vertical) cuyo origen se encuentra en el extremo inferior izquierdo de la imagen (ejes de coordenadas habituales).

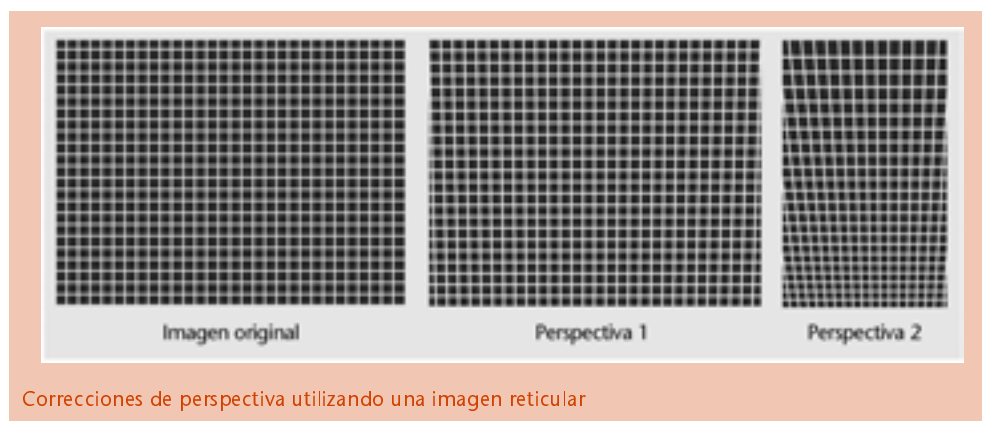
Esta ecuación nos indica que la coordenada y se mantiene, debido a que sólo queremos realizar la corrección de perspectiva en el sentido horizontal; la corrección horizontal depende del valor de la coordenada y . Para puntos que están situados en la parte inferior de la imagen, el valor de y es muy pequeño y la coordenada x' será prácticamente igual a la coordenada x . En cambio, a medida que el valor de y aumenta, αy será significativo y las coordenadas x' se expanden con respecto al centro de la imagen (coordenada x_0). El efecto de la corrección de perspectiva será tanto más exagerado cuanto mayor sea el valor de α .

La corrección de perspectiva modifica el área o región donde se define la imagen ya que, en nuestro ejemplo, la parte superior de la imagen será más ancha que la parte inferior. Para volver a disponer de una región rectangular es necesario eliminar parte de la información de la imagen por medio del uso de un nuevo muestreo mediante una ventana rectangular.

Representación gráfica de las transformaciones geométricas

Una forma habitual para ver el efecto de las transformaciones geométricas en una imagen consiste en utilizar una imagen formada por una retícula rectangular como la que vemos en la siguiente figura. Podemos interpretar el efecto de la transformación de forma intuitiva viendo el resultado que obtenemos con este tipo de imágenes.

En este ejemplo se ha realizado la misma corrección de perspectiva que hemos aplicado en el ejemplo anterior y mostramos los resultados obtenidos con dos valores del parámetro a a distintos. Observad que en el ejemplo de la derecha, que se corresponde con el valor de a más elevado, ha sido necesario recortar una parte considerable de la imagen en el sentido horizontal para poder obtener una zona útil rectangular.



Metodología de la transformación

En el apartado introductorio hemos visto que una transformación del tipo

$$(x', y') = T[(x, y)]$$

define de forma natural el proceso mediante el que se realiza la transformación geométrica. En esencia, podemos descomponer este procedimiento en forma algorítmica en las siguientes etapas:

- 1) Explorar todos los elementos (x, y) de la imagen original.
- 2) Determinar las nuevas coordenadas del destino (x', y') de cada píxel.
- 3) Introducir el valor (brillo o color) del píxel original en la nueva posición.

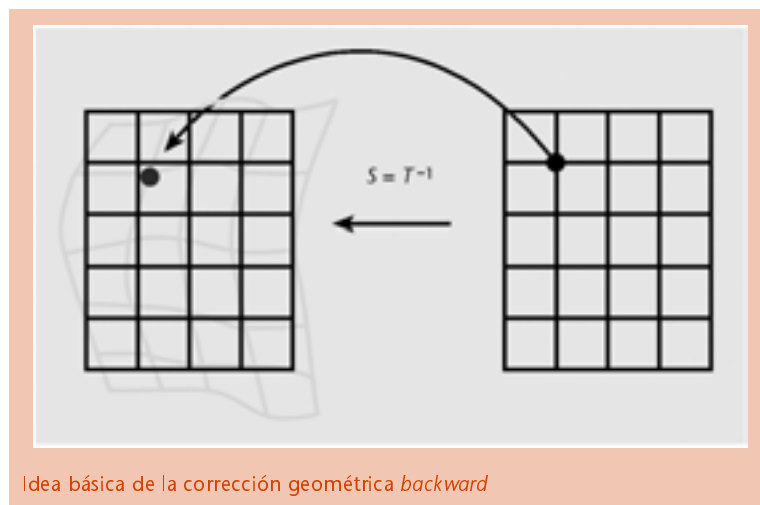
Esta metodología se conoce como corrección geométrica hacia adelante (*forward*) y aunque es simple e intuitiva adolece de un problema de cierta importancia.

En efecto, aunque exploremos todos los píxeles de la imagen original no tenemos ninguna garantía de que también se explorarán todos los píxeles de la imagen resultante. Así pues, el resultado final depende de las características de la transformación $T()$, pero es evidente que, en algunos casos, es posible que exista un conjunto de puntos (x'_k, y'_k) a los que no se les haya asignado ningún valor.

Por norma general este problema se soluciona realizando algún tipo de interpolación o extrapolación de los valores de los píxeles adyacentes a los que sí se les han asignado los niveles de gris o color a partir de la imagen original. El resultado obtenido suele resultar satisfactorio siempre que el número de píxeles de los que no disponemos de ninguna información sea reducido y esté disperso en el resto de los elementos de la imagen. En aquellos casos en los que aparecen áreas o regiones de tamaño considerable formadas en exclusiva por este tipo de elementos, la interpolación o extrapolación de valores suele resultar ineficaz.

Otro problema posible, aunque de menor importancia, es que algunas coordenadas (x', y') de la imagen destino pueden estar asociadas a más de un punto de la imagen original, lo cual significa que la transformación $T()$ proporciona el mismo par de coordenadas para puntos distintos. La solución satisfactoria a este problema acostumbra a consistir en sustituir el nivel de gris por el valor medio de todos los elementos de imagen que se han colapsado en un mismo punto.

Corrección hacia atrás (*backward*)



Una alternativa a la corrección hacia adelante es replantear el problema de la transformación geométrica de forma que sea la imagen destino la que gobierna el procedimiento de la transformación. Esta filosofía se ilustra en la figura adjunta y,

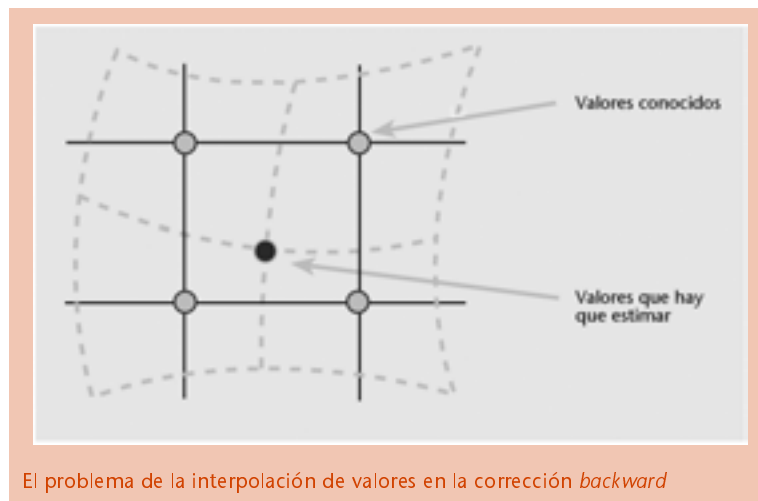
básicamente, consiste en plantear la transformación matemática de forma inversa a la que en un principio parece natural. Ahora disponemos de una imagen destino que está formada por una retícula rectangular de elementos de imagen con coordenadas (x', y') . Exploraremos todos estos píxeles de forma secuencial y la transformación nos proporcionará las coordenadas (x, y) del elemento de imagen correspondiente en la imagen original. En tal caso podemos expresar la transformación como:

$$(x, y) = S[(x', y)']$$

En general, las transformaciones $S()$ y $T()$ se encuentran relacionadas, siendo una la recíproca de la otra.

Mediante la corrección hacia atrás podemos garantizar que todos los píxeles de la imagen destino son explorados y se les asigna algún valor de la imagen original. Empleamos la corrección *backward* más que la *forward* en la mayor parte de las aplicaciones.

Interpolación de valores



La transformación $S(x', y')$ suele proporcionar valores reales que no coinciden exactamente con una coordenada (píxel) de la imagen original. El problema se ha ilustrado en la figura del apartado “Metodología de la transformación”, donde mostramos que el valor proporcionado por la transformación no concuerda con la retícula rectangular de la imagen original. Existen distintas alternativas para determinar el nivel de gris que asignaremos a la imagen destino. El cálculo se basa en utilizar algoritmos de interpolación conocidos que utilizan los niveles de gris de los píxeles que encontramos en la imagen original.

Los algoritmos más utilizados son:

- interpolación de orden cero (*nearest neighbour*),
- interpolación bilineal,
- interpolación bicúbica,
- interpolación mediante *splines*.

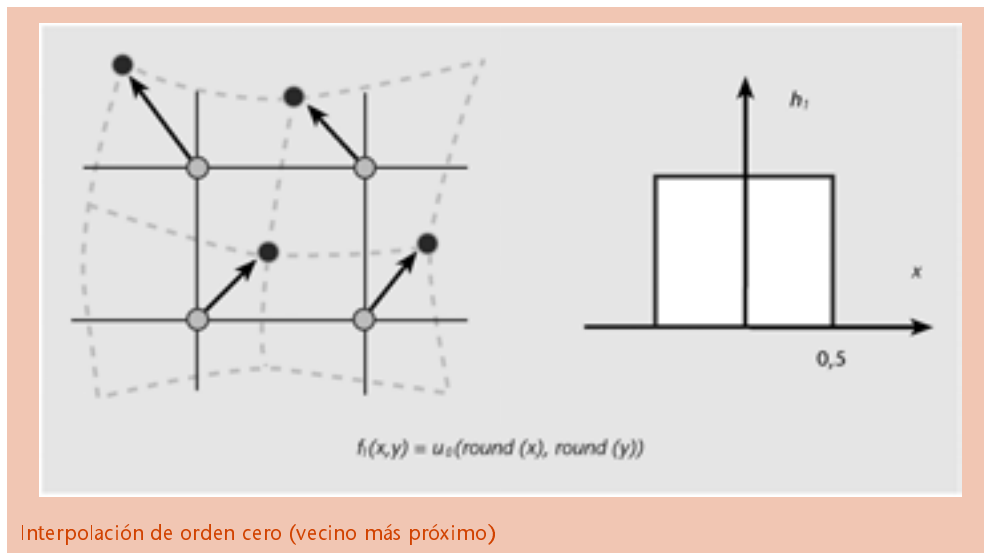
En los apartados siguientes nos disponemos a analizar las características de los tres primeros métodos que más se utilizan en la práctica.

De forma analítica, todos los algoritmos de interpolación considerados se pueden adaptar a la siguiente formulación general:

$$f_n(x, y) = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} g_s(l\Delta x, k\Delta y) \cdot h_n(x - l\Delta x, y - k\Delta y)$$

donde $f_n(x, y)$ es el valor interpolado, g_s representa los elementos de la imagen original y h_n los coeficientes de la interpolación. Esta expresión es muy parecida a la que hemos utilizado para definir las plantillas de los filtros espaciales. De hecho, en algunos casos es posible interpretar todos estos algoritmos de interpolación como filtros espaciales que actúan en la imagen original. En general, sólo se utilizan los elementos más próximos de la imagen original, por lo que la mayoría de los coeficientes de interpolación (h_n) son nulos.

Interpolación de orden cero



Este procedimiento consiste en sustituir el valor interpolado por el valor más cercano del píxel de la imagen original. En este caso, todos los coeficientes del filtro interpolador son nulos, a excepción del correspondiente al vecino más próximo.

La figura ilustra la esencia de este método cuya complejidad computacional es muy reducida. La interpolación por el vecino más próximo suele producir excelentes con-

tornos horizontales y verticales, pero tiene problemas con los contornos diagonales, en los que aparece un efecto de escalera.

Interpolación bilineal

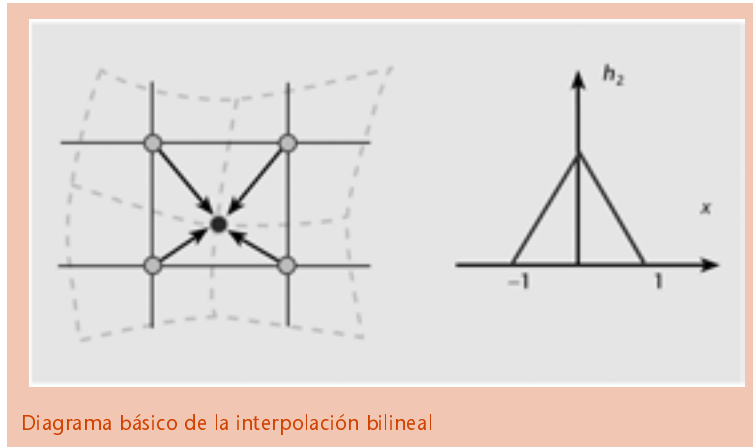


Diagrama básico de la interpolación bilineal

En esta interpolación intervienen los cuatro vecinos más próximos de forma proporcional a la distancia entre el punto que vamos a estimar y los puntos de los que disponemos. En la figura ilustramos la ponderación, y la podemos representar analíticamente mediante la ecuación:

$$f_2(x, y) = (1 - a)(1 - b) \cdot u_0(l, k) + a(1 - b) \cdot u_0(l + 1, k) + b(1 - a) \cdot u_0(l, k + 1) + ab \cdot u_0(l + 1, k + 1)$$

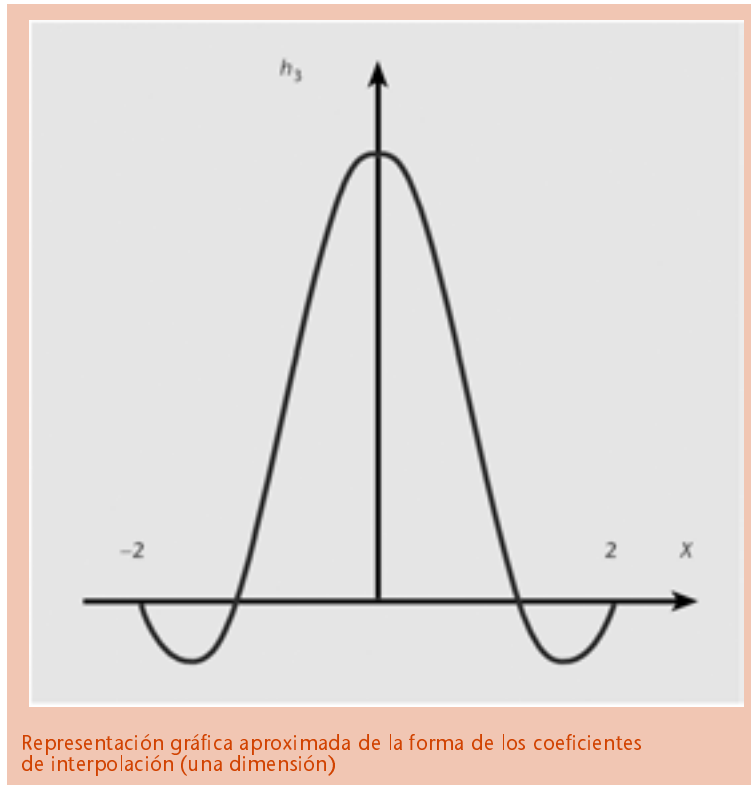
$$l = \text{round}(x), \quad a = x - l \\ k = \text{round}(y), \quad b = y - k$$

La interpolación bilineal introduce una pequeña pérdida en resolución y un ligero filtrado paso bajo; por otra parte, no introduce problemas de aproximaciones en escalera en los contornos diagonales.

Interpolación bicúbica

En este caso, se ponderan los 16 píxeles más próximos al punto que deseamos estimar. La función de ponderación (que determina los coeficientes h_n) suele ser una función del tipo sombrero mejicano (*mexican hat*) que podéis ver en la figura. Este procedimiento de interpolación produce excelentes resultados aunque resulta complejo desde el punto de vista computacional, y se utiliza a menudo en escáneres comerciales para aumentar la resolución de forma digital. Los coeficientes del interpolador vienen dados por la siguiente ecuación:

$$h_3 = \begin{cases} 1 - 2|x|^2 + |x|^3 & \text{si } |x| < 1 \\ 4 - 8|x| + 5|x|^2 - |x|^3 & \text{si } 1 \leq |x| < 2 \\ 0 & \text{resto valores} \end{cases}$$



Transformaciones más habituales

	No linealidad en la exploración de línea. Distancia entre el objeto y el escáner no uniforme.
	Distorsión panorámica. Aparece en escáneres o cámaras de línea en las que existen errores en el motor de desplazamiento.
	Distorsión de skew. Desplazamiento relativo entre objeto y sistema de captura (es habitual en satélites que no están en órbita geoestacionaria).
	Distorsión de escala. La distancia entre los objetos y la cámara varía en tomas sucesivas.
	Distorsión de perspectiva debida a la proyección de 2D de objetos 3D.

Correcciones de distorsión más habituales

Las transformaciones geométricas se utilizan para corregir distintos tipos de distorsión o bien para producir distorsiones dirigidas a efectos especiales. En la figura hallamos un resumen de algunas de las correcciones más utilizadas.

Desde un punto de vista analítico, la mayoría de las transformaciones se definen como polinomios de las variables independientes. Las más empleadas son las que veremos en los subapartados siguientes.

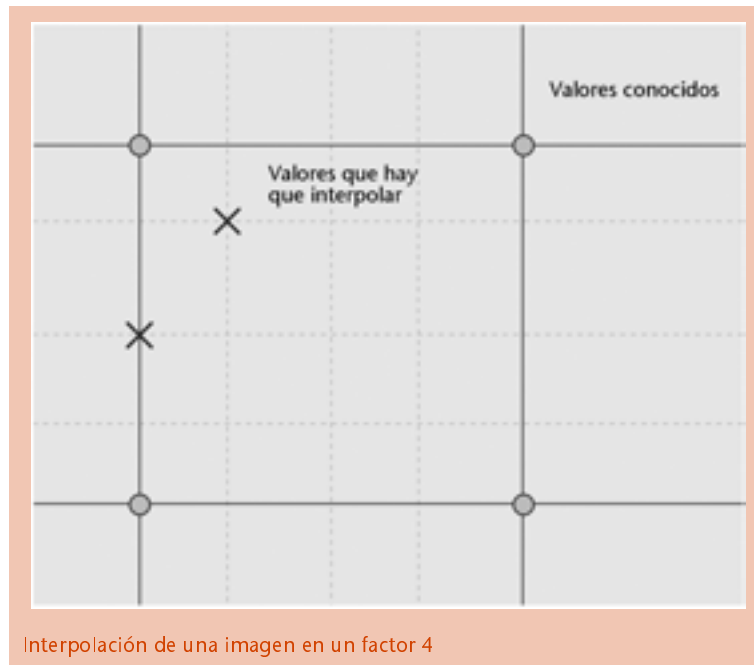
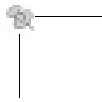
Cambio de escala

La transformación de cambio de escala se utiliza, principalmente, para cambiar el tamaño de las imágenes (interpolación o diezmar). Las ecuaciones de la transformación geométrica vienen dadas por:

$$x = ax'$$

$$y = by'$$

En caso de que $a < 1$ y $b < 1$ se trata de una interpolación de la imagen; cuando, por otra parte, a y b son mayores que la unidad estaremos reduciendo el número de píxeles de la imagen.



Consideremos como ejemplo una interpolación (aumento del tamaño de la retícula de muestreo) en un factor 4 en ambas direcciones. Este problema se corresponde con unos valores de a y b iguales a 0,25. Según la ecuación de la transformación geométrica, para

unos valores de x' e y' enteros podemos tener valores de x e y en incrementos de 0,25. Los posibles valores que tenemos que interpolar en las coordenadas del espacio de la imagen original se representan de forma esquemática en la siguiente figura.

Rotación

La transformación asociada a una rotación queda definida mediante la siguiente ecuación:

$$x = x' \cos \theta + y' \sin \theta$$

$$y = -x' \sin \theta + y' \cos \theta$$

Sesgo (*skew*)

Para la corrección de sesgo utilizaremos ecuaciones del tipo:

$$x = x' + y' \tan \theta$$

$$y = y'$$

Transformación afín

Una transformación genérica que puede englobar la rotación, el cambio de escala y el sesgo es:

$$x = a_0 + a_1 \cdot x' + a_2 \cdot y'$$

$$y = b_0 + b_1 \cdot x' + b_2 \cdot y'$$

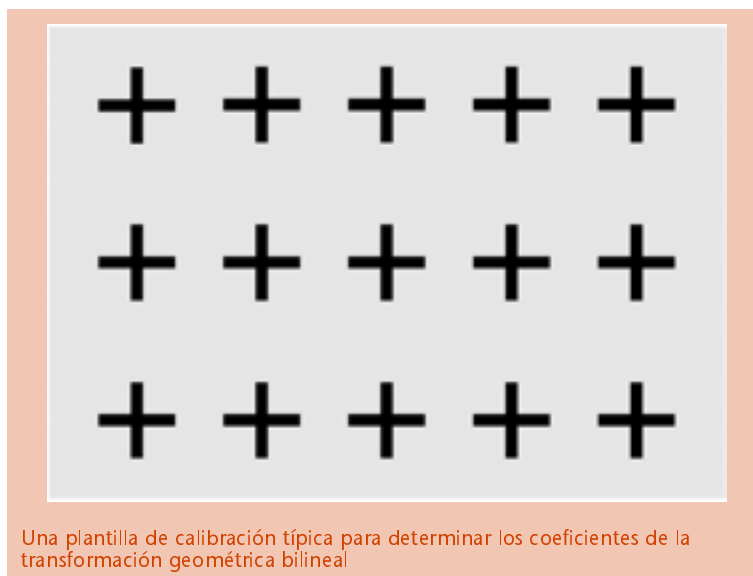
Transformación bilineal

Esta transformación es muy genérica y permite englobar distintos tipos de distorsiones complejas, pero es importante que no la confundamos con la interpolación bilineal. La relación entre las coordenadas de la imagen original y la imagen destino se puede expresar a modo de combinaciones polinómicas de primer orden. La ecuación asociada a la transformación será ésta:

$$x = a_0 + a_1 \cdot x' + a_2 \cdot y' + a_3 \cdot x' \cdot y'$$

$$y = b_0 + b_1 \cdot x' + b_2 \cdot y' + b_3 \cdot x' \cdot y'$$

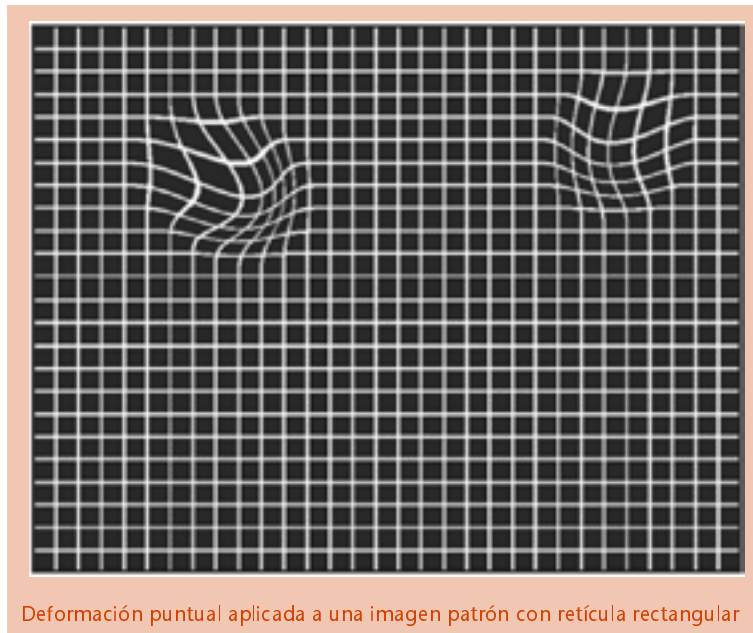
Hay que calcular los valores de los parámetros a_k y b_k para cada caso. Con frecuencia, utilizamos plantillas de calibración (podéis ver la figura) en las que se conoce con exactitud las posiciones de distintos puntos característicos de la imagen. Sobre la imagen obtenida, después de escanear la plantilla de calibración (imagen que contiene los errores y posibles distorsiones geométricas), aplicamos métodos de reconocimiento de formas para determinar las posiciones de los puntos críticos y ajustamos los coeficientes a_k y b_k con objeto de que los errores finales sean mínimos.



Deformaciones puntuales

Las deformaciones puntuales son unas de las transformaciones geométricas más utilizadas por aplicaciones de efectos especiales (*goo*, *morpher*, etc) y consisten en desplazar un punto de la imagen origen mediante un vector dirigido por el ratón del usuario. La posición del punto aparece en la imagen destino en la misma posición que ha especificado el usuario; los puntos que están en la proximidad del punto sobre el que se ha ejercido la acción también se desplazan, aunque el vector de desplazamiento es cada vez menor a medida que aumenta la distancia. Por norma general, estos programas permiten controlar el radio de acción sobre el que actúa el desplazamiento.

En la figura que vamos a ver a continuación representamos el resultado de una deformación puntual aplicada a una imagen patrón que está formada por una retícula rectangular. Asimismo, ilustramos el efecto de distorsión aplicado a una imagen natural.



En los programas comerciales de retoque fotográfico existen varios tipos de transformaciones geométricas que permiten realizar distorsiones de la imagen, la mayor parte de las cuales son modelos de movimiento o variantes de las deformaciones puntuales. Las distorsiones más utilizadas son las esféricas (la transformación geométrica acerca o aleja los puntos de un centro), las de tipo torbellino y en zig-zag, entre otras.

Etapa 4: Otras aplicaciones

Introducción

En este módulo examinaremos algunas aplicaciones adicionales del tratamiento digital de la imagen. Todas estas aplicaciones resultan especialmente complejas y combinan distintas metodologías para procesar las imágenes. La complejidad y la multitud de variantes en algunos de estos sistemas impiden que realicemos una descripción detallada de todos los posibles métodos, por lo que describiremos, en exclusiva, sus principios generales.

Metamorfosis de imágenes

En general, la metamorfosis de imágenes se conoce con el nombre de *morphing* y consiste en obtener transiciones graduales entre dos imágenes. El objetivo principal de esta metamorfosis es formar una secuencia de vídeo que, al reproducirla, produzca la sensación de que la metamorfosis se está llevando a cabo con total naturalidad.

Los efectos de la metamorfosis han sido ampliamente utilizados en el cine antes de la llegada de los efectos especiales. La técnica más usada era el corte inteligente (*clever-cut*) en la que, mediante cortes, se iban superponiendo distintos estados de la metamorfosis (hombre lobo, Doctor Jeckill y Mr. Hide). Se solía aprovechar el movimiento del sujeto, de la cámara o el paso del protagonista entre árboles u obstáculos para facilitar la edición de la escena.

Los primeros efectos del *morphing* digital fueron creados por la Industrial Light and Magic para la película *Willow* (1988). Más tarde han sido utilizados en una gran cantidad de películas de ciencia ficción y hoy día el cine utiliza procedimientos realmente sofisticados en los que muchos de los parámetros de la metamorfosis se pueden calcular de forma automática con la ayuda de un ordenador.

La idea básica de una metamorfosis entre dos imágenes es la de realizar una disolución paulatina de una imagen en la otra (*cross-dissolve*). La disolución consiste en combinar las dos imágenes en distintas proporciones en función de su posición en la secuencia temporal. Así pues, en las primeras imágenes de la secuencia predomina el primer sujeto, mientras que en las últimas, predomina el resultado final de la metamorfosis. La imagen central de la secuencia es especialmente importante y su calidad determina, en buena medida, la calidad final de la secuencia. En una disolución simple, calcularemos la imagen intermedia combinando a partes iguales la primera y última imagen de la secuencia.

Podemos formular la disolución directa de una manera muy sencilla:

$$w[n, m] = \frac{T-t}{T} \cdot u[n, m] + \frac{t}{T} \cdot v[n, m],$$

donde $u[n, m]$ representan los niveles de gris de la imagen inicial, $v[n, m]$, los de la final, $w[n, m]$, la disolución entre las dos imágenes y T , el tiempo de duración total de la secuencia. Observad que el instante inicial $w[n, m]$ se corresponde exactamente con la imagen inicial, y el instante final, con la final. Obtendremos el resto de los fotogramas de la secuencia ponderando de forma proporcional los niveles de gris de ambas imágenes.

Uno de los principales problemas de la disolución directa es que las imágenes iniciales y finales deben estar adaptadas a la perfección para que el resultado tenga credibilidad. Y, por adaptación entendemos que los puntos críticos de cada una de las imágenes (posición de los ojos, boca, nariz y orejas) tengan que situarse en las mismas posiciones en ambas imágenes, puesto que, en caso contrario, al disolver las dos imágenes apreciaremos efectos indeseados, como la aparición de dos bocas, cuatro ojos, etc. Resulta evidente que esta particularidad limita considerablemente la aplicación de una disolución directa en los efectos del *morphing*.

En la figura que vemos a continuación nos encontramos con un ejemplo de dos imágenes (inicial y final) y el resultado intermedio que obtenemos al aplicar una disolución directa. Aunque las dos imágenes tienen posiciones espaciales parecidas (no exactas), el resultado es por completo insatisfactorio.



Resultado de una metamorfosis utilizando la disolución directa

Morphing utilizando deformaciones puntuales

Uno de los primeros métodos que se utilizaron para generar digitalmente la metamorfosis fueron las deformaciones puntuales, en las que el usuario tiene que esta-

blecer puntos de correspondencia entre la imagen inicial y la final. Los puntos de correspondencia pueden ser el centro de la nariz de cada una de las imágenes, los extremos de los ojos, los lóbulos de las orejas, etc.

La filosofía general del método es parecida a la disolución directa, aunque antes de realizar la disolución entre las imágenes aplicaremos una deformación puntual a cada una de ellas. La deformación puntual es tal que los puntos que han sido marcados en la imagen original se deforman de manera paulatina hacia el destino que ha marcado el usuario en la imagen final. La deformación depende del tiempo, de forma que, en la imagen central, los puntos sólo han llegado a la mitad del recorrido marcado por el usuario. Por su parte, a la imagen final le aplicaremos un procedimiento parecido pero inverso; es decir, para obtener la imagen central sólo habrá que desplazarlos hacia la mitad de su recorrido. El objetivo es que los puntos críticos de las imágenes deformadas coincidan en cualquier instante de tiempo para que la disolución resulte natural.

En el ejemplo siguiente hallaremos el resultado de un *morphing* basado en esta técnica. Las dos primeras imágenes indican los puntos de correspondencia que teníamos prefijados. El resultado de la imagen central, como podemos ver, es mucho más real que el obtenido mediante una disolución directa.



El método de las deformaciones puntuales, aunque produce resultados aceptables, suele tender a producir deformaciones en los sujetos que puedan parecer poco naturales, hecho que se hace más notorio a medida que difieren más las posiciones de los puntos críticos en ambas imágenes. Una variante que produce resultados bastante mejores es la deformación lineal. En esta variante, el usuario tiene que especificar los dos puntos de una recta y sus correspondencias en la imagen de destino. El procedimiento determina de forma automática las transformaciones geométricas que hay que aplicar a las imágenes para que las rectas especificadas coincidan en todo momento durante la disolución de imágenes. En algunos procedimientos avanzados existen

técnicas para localizar automáticamente los puntos o rectas críticos, simplificando la tarea de especificación de la metamorfosis.

Fusión de imágenes

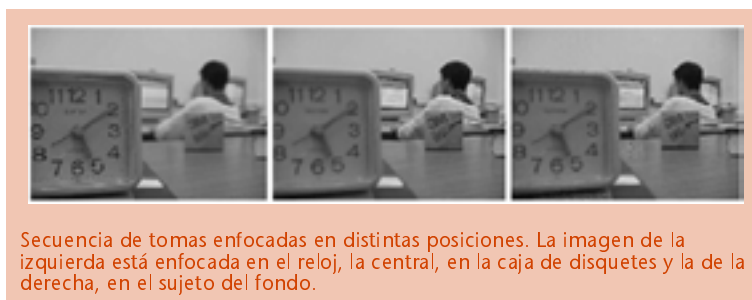
La fusión de imágenes consiste en combinar varias imágenes de un mismo objeto, tomadas desde el mismo punto de vista, aunque con unas distancias de enfoque distintas. El objetivo es producir una imagen global donde todos los objetos, con independencia de su distancia con respecto a la cámara, permanezcan enfocados a la perfección. Las aplicaciones de la fusión son muy variadas y abarcan desde la fotografía digital a la microscópica de objetos 3D (donde la profundidad de campo es muy reducida y sólo es posible observar una sección enfocada de manera correcta).

Los métodos utilizados para realizar la fusión de las imágenes pueden combinar técnicas muy diferentes y producir resultados de distinta naturaleza. La estrategia más simple consiste en determinar los gradientes (derivadas) de cada una de las imágenes que constituyen la toma y combinar los resultados en función de los gradientes obtenidos en cada una de las zonas de la imagen. El fotograma final se construye a partir de la combinación de las distintas tomas de manera que para cada zona u objeto utilicemos la parte de la fotografía que ha quedado mejor enfocada. Por su parte, el valor del gradiente proporciona una idea cuantitativa del enfoque de cada región y se utiliza como criterio para determinar cuál es la toma que contribuirá en cada zona de la fotografía. Recordad que cuanto más enfocados están los objetos, mayor es el contenido en alta frecuencia (contornos bien definidos, con presencia de cambios abruptos y, por tanto, elevados valores del gradiente).

Para mejorar los resultados es conveniente corregir previamente las imágenes respecto de la distancia de enfoque. En efecto, al modificar la distancia de enfoque se producen algunos cambios minúsculos en cuanto a la zona visible de la imagen (ligero cambio en el ángulo de visión), variaciones que podemos corregir mediante transformaciones geométricas. El objetivo final, en cualquier caso, será obtener un conjunto de transformaciones que conviertan todas las tomas con la misma zona de visión.

Otros posibles algoritmos para combinar los resultados de la secuencia de tomas se basan en la realización de una serie de transformaciones de la imagen, combinando los resultados de estas transformaciones para producir la imagen final. Las transformaciones más utilizadas son las de Fourier, coseno y wavelet.

En las estas figuras podemos ver un resultado típico de fusión de imágenes.





La fusión de las tres imágenes se ha obtenido utilizando un algoritmo basado en la transformada wavelet. Los distintos objetos que forman la imagen aparecen enfocados de manera correcta.

Fotografía panorámica

En el contexto del tratamiento digital de imagen, entendemos por fotografía panorámica el conjunto de técnicas que se utilizan para combinar fotogramas parciales de una misma escena en un único fotograma final (panorama).

Como en las aplicaciones anteriores, los métodos utilizados para generar el panorama a partir de las tomas parciales pueden hacer uso de estrategias de tratamiento de imagen muy distintas y dependen de las condiciones en las que se han tomado las fotografías. Es habitual distinguir entre las tomas que se han realizado pivotando el ángulo de una cámara que se encuentra situada en una posición fija (en general sobre un trípode) de las tomas en las que interviene un movimiento lateral de la cámara (por ejemplo, fotografiar un póster con desplazamientos de la cámara).

Es habitual que el proceso de construcción del panorama se descomponga en dos partes. La primera consiste en aplicar transformaciones geométricas para corregir los posibles problemas de perspectiva que distorsionan los objetos y que pueden dificultar la formación del panorama final. En general, resulta útil que exista entre un 30% y un 50% de solapamiento entre los fotogramas sucesivos para, así, facilitar las tareas de recomposición automática del panorama.

Si hemos recogido las tomas parciales con un gran angular, los objetos tendrán algunas deformaciones esféricas que pueden resultar difíciles de compensar. Además, estas distorsiones acostumbran a tener direcciones distintas en cada uno de los fotogramas. En efecto, el objeto que encontramos en la parte derecha de un fotograma aparecerá en la parte izquierda del siguiente fotograma, con lo que sus desviaciones se producirán en sentidos contrarios. Así que, como vemos, una efectiva corrección de las distorsiones ópticas facilita notablemente la recomposición del panorama.

En algunas cámaras digitales podemos utilizar la información obtenida durante la toma de la fotografía (distancia focal) y los modelos de la propia cámara para compensar y corregir estas distorsiones. No obstante, este hecho no resulta trivial, puesto que la deformación de los objetos también depende de su distancia con respecto al objetivo.

Una vez realizada la corrección de perspectiva, hay que combinar las distintas tomas en un único panorama. Los métodos más utilizados se basan en la correlación de las imágenes, que, en esencia, buscan zonas donde exista una gran similitud entre los píxeles de las dos tomas que se tienen que solapar. Lo más normal es que la zona común se sustituya por el valor medio de los píxeles de las dos fotografías.

En el ejemplo siguiente podremos ver la composición de un fotograma a partir de dos fotografías. Observad el solapamiento entre las dos tomas y la aparición de algunos errores en la región en la que se ha realizado la fusión de las dos fotografías. Notad, también, la corrección de perspectiva que ha distorsionado los fotogramas a formas no rectangulares antes de combinarlos en uno solo.





Panorama final obtenido mediante la combinación de los tres fotogramas

