

Procesado de Imágenes



ANTONIO TABERNERO GALÁN, 2015

FOTOGRAFÍA COMPUTACIONAL, ETSII, UPM

Procesado de Imágenes

La charla de hoy la dedicaremos a exponer las técnicas básicas para modificar, mejorar o extraer información de una imagen digital.

Procesado Digital de Imágenes clásico:

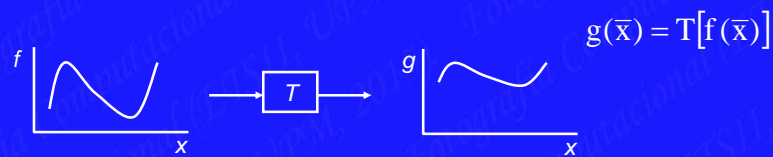
- Sólo es una parte de Fotografía Computacional.
- Suele ser un primer paso de muchos procesos.

ANTONIO TABERNERO GALÁN, 2015

FOTOGRAFÍA COMPUTACIONAL, ETSII, UPM

Transformaciones de rango (hoy)

El valor del píxel cambia pero sigue estando asociado a la misma posición (x,y) original.



Original $im(\bar{x})$

$im'(\bar{x}) = T[im(\bar{x})]$

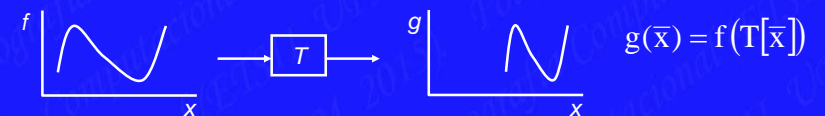


ANTONIO TABERNERO GALÁN, 2015

FOTOGRAFÍA COMPUTACIONAL, ETSII, UPM

Transformaciones de dominio (next)

Transformaciones de dominio: el valor del píxel no cambia pero su posición cambia de (x,y) a una nueva (x',y').



Original $im(\bar{x})$

$im'(\bar{x}) = im(T[\bar{x}])$



ANTONIO TABERNERO GALÁN, 2015

FOTOGRAFÍA COMPUTACIONAL, ETSII, UPM

Transformaciones de Rango

El valor del píxel cambia pero sigue estando asociado a la misma posición espacial (x,y) original.



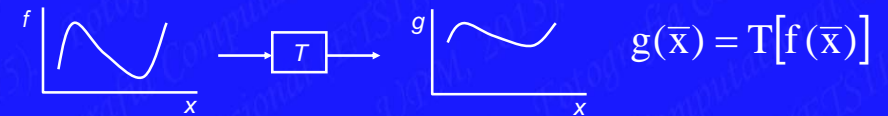
- 1) Transformaciones punto a punto: el nuevo valor solo depende del valor del píxel original.

$$g(\bar{x}) = T[f(\bar{x})]$$

- 2) Transformaciones de vecindad: el nuevo valor depende de los valores de la vecindad.

$$g(\bar{x}) = T[f(\text{vecindad}(\bar{x}))]$$

Transformaciones punto a punto

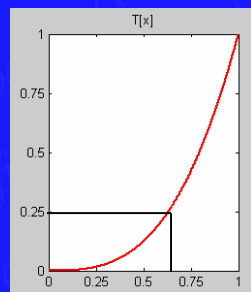


La transformación queda definida dando la aplicación $y=T(x)$ que convierte un valor de entrada (x) en otro de salida (y).

Usualmente $T()$ se define sobre el intervalo de entrada $[0,1]$ y su salida cubre el intervalo $[0,1]$ (0 = negro, 1 = blanco).

La $T()$ usada suele ser monótona creciente: asegura que la dirección oscuro-claro se mantenga en la imagen de salida.

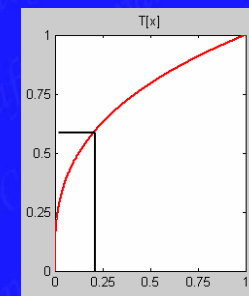
Ejemplos de aplicaciones $out=T(in)$



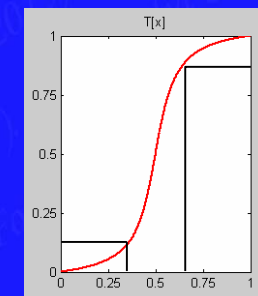
Comprime oscuros Expande claros



Ejemplos de aplicaciones $out=T(in)$



Expande oscuros Comprime claros

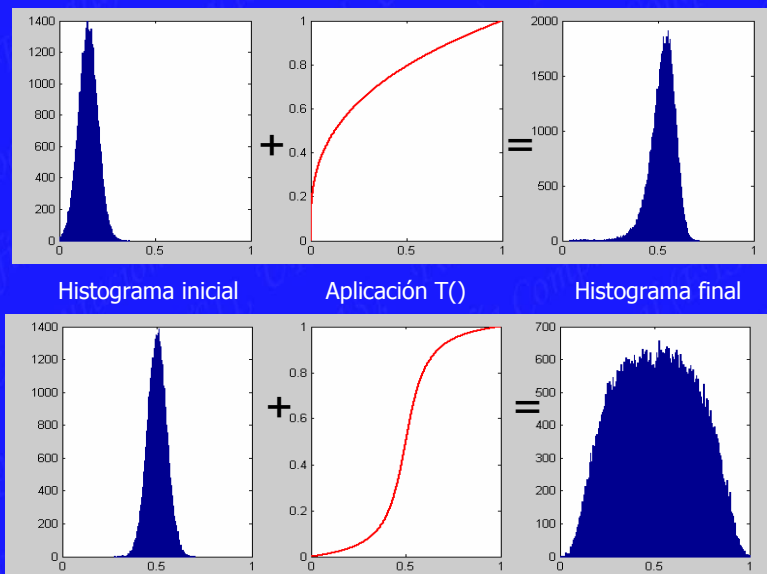


Comprime oscuros Expande medios Comprime claros

Zonas de crecimiento lento de T ($T' \approx 0$) = compresión.

Zonas de rápido crecimiento de T ($T' \gg 1$) en T = expansión.

Efecto sobre histogramas



ANTONIO TABERNERO GALÁN, 2015

FOTOGRAFÍA COMPUTACIONAL, ETSII, UPM

Crear $T[]$ ajustadas al histograma

Histograma puede usarse para automatizar la elección de $T()$.

Posible objetivo: crear un histograma cubriendo todo el rango y más o menos plano (un histograma ecualizado).

- Si hay alta concentración de valores (histograma alto) debemos expandir (derivada de T alta) y viceversa.

$$T'(s) \propto h(s) \Rightarrow T(x) \approx \int_0^x h(s) \cdot ds$$

Otro objetivo: conseguir $T(x)$ tal que aplicada a una imagen su histograma resultante se parezca al de otra.

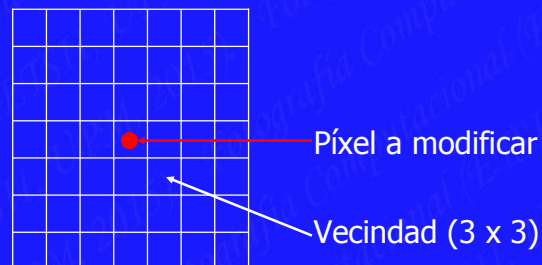
ANTONIO TABERNERO GALÁN, 2015

FOTOGRAFÍA COMPUTACIONAL, ETSII, UPM

FILTRADO de IMÁGENES

Transformaciones de vecindad: el nuevo valor depende de los valores de sus alrededores: $g(\bar{x}) = T[f(\text{vecindad}(\bar{x}))]$

La vecindad se define como la ventana (3x3, 5x5, etc.) de píxeles de alrededor usados en el cálculo del valor de salida.

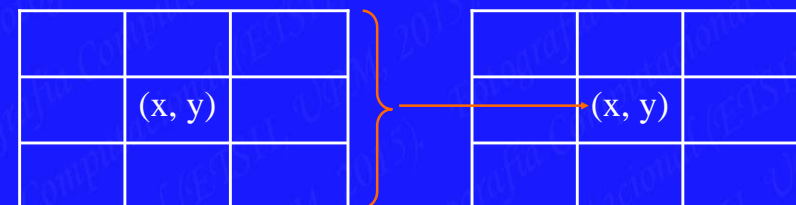


ANTONIO TABERNERO GALÁN, 2015

FOTOGRAFÍA COMPUTACIONAL, ETSII, UPM

Filtrado de Imágenes

Para cada píxel (x,y) de la imagen original se calcula una función de los valores de los píxeles vecinos.



A partir de esos 9 valores se obtiene un resultado que será el valor del píxel (x,y) de la imagen de salida.

ANTONIO TABERNERO GALÁN, 2015

FOTOGRAFÍA COMPUTACIONAL, ETSII, UPM

Numerosas aplicaciones

- Realce de imágenes, incrementar contraste, suavizar o resaltar bordes.
- Cambios de tamaño: suavizado previo a reducir una imagen, implementación de interpolaciones lineales.
- Eliminación/reducción de ruido.
- Extraer información sobre la imagen: bordes, gradientes
- Identificar texturas en una imagen, detectar patrones o puntos característicos.

Filtros lineales

- El valor de salida es una combinación lineal de los valores de los píxeles vecinos.
- El filtro queda definido por los coeficientes que se aplican a los valores vecinos antes de sumarse para obtener el resultado final o valor de salida.
- Los coeficientes se dan como una máscara o filtro: su posición indica a que vecino se aplican.

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

0	1/4	0
1/4	0	1/4
0	1/4	0

0	1/4	0
1/4	-1	1/4
0	1/4	0

Ejemplo: filtro de media

$$h[,] = \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

- Reemplaza cada pixel con la media de los vecinos.
- La suma de sus coeficientes es la unidad: mantiene el valor medio de la imagen.
- Suaviza imagen, elimina cambios bruscos, detalle.
- Modifica poco las zonas de cambios suaves o graduales y no cambia zonas de valor constante.

Efecto filtro promedio sobre imagen



Imagen clásica disponible en MATLAB:
`imread('cameraman.tif');`

Aplicamos filtro promedio 3x3, 5x5, 7x7



Detalles más pequeños que el tamaño del filtro se pierden.
Las zonas con poco detalle no se ven afectadas.

Aplicación filtro promedio $h[,] = \frac{1}{9}$

1	1	1
1	1	1
1	1	1

im[.,.]

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

im'[.,.]

0									

Slide Credit: S. Seitz

ANTONIO TABERNEIRO GALÁN, 2015

FOTOGRAFÍA COMPUTACIONAL, ETSII, UPM

Aplicación filtro promedio $h[,] = \frac{1}{9}$

1	1	1
1	1	1
1	1	1

im[.,.]

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

im'[.,.]

0	10								

Slide Credit: S. Seitz

ANTONIO TABERNEIRO GALÁN, 2015

FOTOGRAFÍA COMPUTACIONAL, ETSII, UPM

Aplicación filtro promedio $h[,] = \frac{1}{9}$

1	1	1
1	1	1
1	1	1

im[.,.]

0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	0	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	0	90	90	90	90	90	0	0
0	0	90	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0



im'[.,.]

?	?	?	?	?	?	?	?	?	?
?	0	10	20	30	30	30	20	10	?
?	0	20	40	60	60	60	40	20	?
?	0	30	60	80	80	80	60	30	?
?	0	30	50	80	80	80	60	30	?
?	0	30	50	80	80	80	60	30	?
?	0	20	30	50	50	60	40	20	?
?	10	20	30	30	30	30	20	10	?
?	10	10	10	0	0	0	0	0	?
?	?	?	?	?	?	?	?	?	?

Slide Credit: S. Seitz

ANTONIO TABERNEIRO GALÁN, 2015

FOTOGRAFÍA COMPUTACIONAL, ETSII, UPM

Expresión matemática de filtros lineales

Filtrado: $im2 = \text{filter2}(h, im);$ o $im2 = \text{imfilter}(im, h)$

$$im2[m, n] = \sum_{k, l} h[k, l] \cdot im[m + k, n + l]$$

Convolución: $im2 = \text{conv2}(h, im);$

$$im2[m, n] = \sum_{k, l} h[k, l] \cdot f[m - k, n - l]$$

La convolución invierte la máscara antes de aplicarla.

Son equivalentes si la máscara $h[,]$ es simétrica.

ANTONIO TABERNEIRO GALÁN, 2015

FOTOGRAFÍA COMPUTACIONAL, ETSII, UPM

Filtro identidad ($\Sigma=1$)



Entrada

0	0	0
0	1	0
0	0	0

Filtro $h[,]$



Salida

Respetar el píxel central (1) y no lo combina con el resto (0's)

El resultado es la imagen original.

Filtro promedio ($\Sigma=1$)



Entrada

1	1	1
1	1	1
1	1	1

Filtro $h[,]$



Salida

Elimina los detalles al nivel del tamaño del filtro (1-3 píxeles)
No altera las zonas uniformes (sin detalle).

El resultado es una imagen "suavizada".

Filtro laplaciano ($\Sigma=0$)



Entrada

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} - \frac{1}{9} \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} = \frac{1}{9} \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Filtro $h[,]$



Salida

SALIDA = Imagen original – Imagen suavizada
Imagen con el detalle que se eliminó en la imagen "suavizada".

El resultado tiene media 0 (valores positivos y negativos)

Realce bordes "sharpening" ($\Sigma=1$)



Entrada

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} + \alpha \cdot \frac{1}{9} \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

Filtro $h[,]$



Salida

SALIDA = Imagen original + "extra" de detalle
Aumentamos el detalle de la imagen.

El resultado es una imagen más nitida.
¡ Puede dar problemas si exageramos!

Extracción de bordes ($\Sigma=0$)



Entrada

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

Filtro $h[\quad , \quad]$



Salida

El filtro calcula diferencias entre píxeles vecinos arriba/abajo
Si los píxeles son similares, salida baja, si son distintos, alta.

SALIDA = detecta bordes horizontales de la imagen.

Extracción de bordes ($\Sigma=0$)



Entrada

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

Filtro $h[\quad , \quad]$



Salida

El filtro calcula diferencias entre píxeles vecinos izda/derecha
Si los píxeles son similares, salida baja, si son distintos, alta.

SALIDA = detecta bordes verticales de la imagen.

Extracción de bordes ($\Sigma=0$)



Entrada

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

Filtros $h[\quad , \quad]$



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



Salida

Combinamos la salida de los dos filtros anteriores (la suma de sus valores absolutos).

SALIDA = detecta bordes en cualquier dirección en la escala dada por el tamaño de los filtros usados.

¡ Pero el filtrado ya no es lineal (por el valor absoluto) !

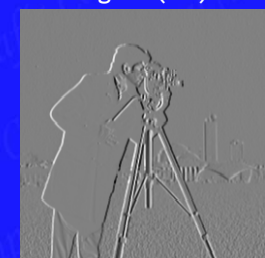
Visualización de imágenes "negativas"

- Cuando la salida de un filtrado (bordes de antes) tiene valores positivos y negativos podemos visualizar su valor absoluto o sumar un gris medio al resultado
- Con MATLAB podemos usar `imagesc()` -> escala la imagen tal que valor mínimo=negro, máximo=blanco.

`imagesc(abs(im))`



`imagesc(im)`



Aplicación a imágenes en color

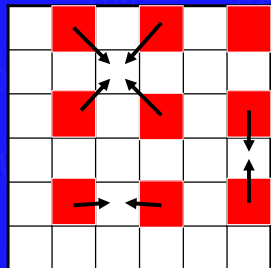


En imágenes color tratamos los planos por separado. Podemos filtrarlos todos o sólo algunos de ellos. A veces es mejor cambiar a otro espacio de color antes de aplicar el filtrado.

Consideraciones generales

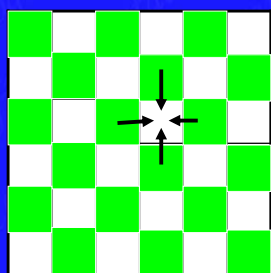
- Filtros $\Sigma = 1$ (paso-bajo)
- Algún tipo de promedio de la vecindad.
 - Salida con valores siempre positivos, similares a los valores originales.
 - Salida de aspecto "similar" a la entrada.
- Filtros $\Sigma = 0$ (paso-alto)
- Calculan diferencias entre píxeles vecinos, capturando el detalle.
 - Valores $+$ y $-$ en la salida.
 - Respuesta baja (≈ 0) en zonas "constantes"
 - Respuesta alta ($+/-$) si la imagen tiene algún patrón que se parezca a la máscara usada.

Interpolación mosaico Bayer



¿Recordais la interpolación lineal usada en el "demosaiing" del LAB3?

En todos los casos (R,G,B) se puede Implementar como un filtrado lineal.



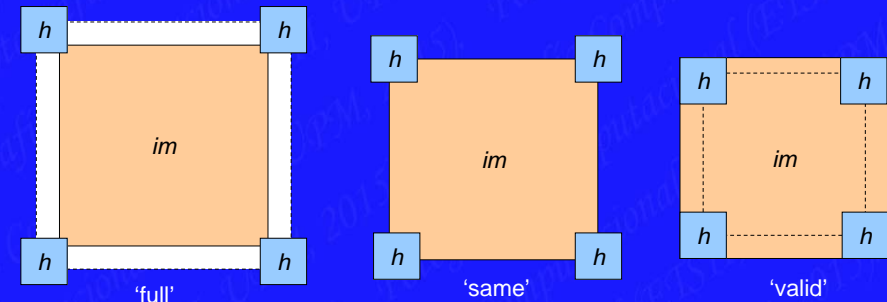
Se parte de los datos conocidos, 0's en las casillas vacías, y se aplica un filtro.

¿Qué filtro (3 x 3) habría que usar para los canales R y B? ¿Y para el G?

Aspectos prácticos al filtrar imágenes

Función de MATLAB: `imfilter(im, h, shape)`

- `shape = 'full'`: calcula la salida desde el primer "contacto" entre imagen y máscara.
- `shape = 'same'`: aplica máscara sólo en aquellos puntos pertenecientes a la imagen.
- `shape = 'valid'`: aplica máscara solo cuando toda la máscara cae dentro de la imagen original.



Rellenar con ceros (u otro valor)



ANTONIO TABERNO GALÁN, 2015

FOTOGRAFÍA COMPUTACIONAL, ETSII, UPM

Wrap around ('circular')



ANTONIO TABERNO GALÁN, 2015

FOTOGRAFÍA COMPUTACIONAL, ETSII, UPM

Propagar filas/columnas ('replicate')



ANTONIO TABERNO GALÁN, 2015

FOTOGRAFÍA COMPUTACIONAL, ETSII, UPM

Reflejar bordes ('symmetry')



ANTONIO TABERNO GALÁN, 2015

FOTOGRAFÍA COMPUTACIONAL, ETSII, UPM

Funciones de MATLAB

Funciones de MATLAB en Image Processing Toolbox:

```
im2=imfilter(im, h, 0); % Amplia con un valor dado (=0)
im2=imfilter(im, h,'circular'); % wrap around.
im2=imfilter(im, g, 'replicate') % Repetir datos del borde
im2=imfilter(im, g, 'symmetric') % reflexión del borde.
```

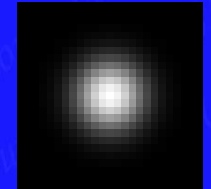
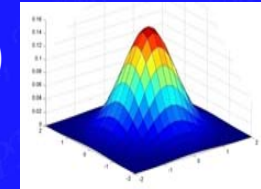
Función MATLAB suministrada (para uso en LAB donde Image Processing no está disponible):

```
fc_imfilter(im, h, 0), fc_imfilter(im, h, 'circular')
fc_imfilter(im, h, 'replicate'), fc_imfilter(im, h, 'symmetric')
```

Filtro gaussiano

Filtro promedio dando más importancia a los píxeles centrales.

$$G_{\sigma}(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$



Al crear una máscara finita hay que "cortar" en algún punto.

Regla: lado máscara = σ o 2σ

$\sigma=1 \rightarrow \text{lado}=2, N=2 \times \text{lado}+1 = 5$

0.003	0.013	0.022	0.013	0.003
0.013	0.059	0.097	0.059	0.013
0.022	0.097	0.159	0.097	0.022
0.013	0.059	0.097	0.059	0.013
0.003	0.013	0.022	0.013	0.003

$h[,]$ de tamaño 5×5 con $\sigma = 1$

Propiedades de filtros gaussianos

Comportamiento similar al filtro promedio: elimina el detalle de la imagen por debajo de su tamaño.

La aplicación repetida de un filtro de Gauss de ancho σ es lo mismo que aplicar uno de ancho mayor ($\sigma\sqrt{2}$) una sola vez.

La función de Gauss es separable y puede factorizarse en el producto de 2 gaussianas 1D

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

Crear máscara gaussiana en MATLAB

```
s=1; x = [-2:2] % sigma = 1, soporte = -2:2 (N=5)
h = exp(-0.5*(x/s).^2)/(sqrt(2*pi)*s) % Expresion gauss 1D
h = h/sum(h); % Normaliza para que suma=1
h2 = h' * h % Crea mascara 2D
```

Alternativamente: `h2=fspecial('gauss',5,1);`

Máscara tamaño 5×5 con $\sigma = 1$

0.003	0.013	0.022	0.013	0.003
0.013	0.059	0.097	0.059	0.013
0.022	0.097	0.159	0.097	0.022
0.013	0.059	0.097	0.059	0.013
0.003	0.013	0.022	0.013	0.003

Promediado con filtro gaussiano



ANTONIO TABERNEIRO GALÁN, 2015

FOTOGRAFÍA COMPUTACIONAL, ETSII, UPM

Promediado filtro de media (box filter)



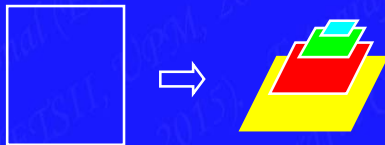
ANTONIO TABERNEIRO GALÁN, 2015

FOTOGRAFÍA COMPUTACIONAL, ETSII, UPM

Pirámides

La escala es un aspecto fundamental en imágenes.
Un mismo patrón puede presentarse a diversas escalas.

Interesa tener representaciones multi-escala de las imágenes



- Invertibles? ¿Puedo volver a la imagen original?
- Redundantes? ¿ Más coeficientes que la imagen original ?
- En orientación? ¿ Separan también orientaciones ?

ANTONIO TABERNEIRO GALÁN, 2015

FOTOGRAFÍA COMPUTACIONAL, ETSII, UPM

Ejemplos Pirámides

Piramide Gaussiana:

Una colección de sucesivas versiones de la imagen suavizadas (con un filtro gaussiano) y progresivamente reducidas de tamaño.

Piramide Laplaciana:

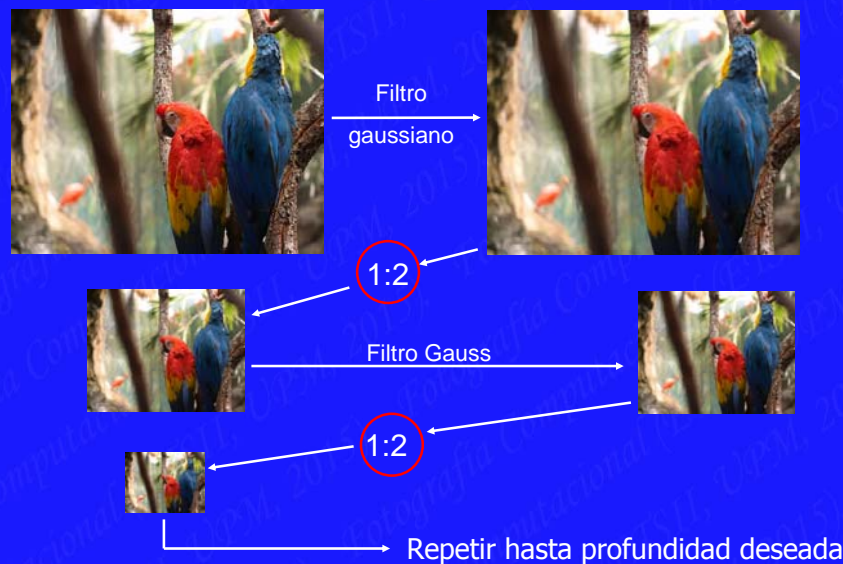
En cada nivel de escala guardamos el "detalle" asociado con esa escala.

Obtenida diferenciando una piramide gaussiana.

ANTONIO TABERNEIRO GALÁN, 2015

FOTOGRAFÍA COMPUTACIONAL, ETSII, UPM

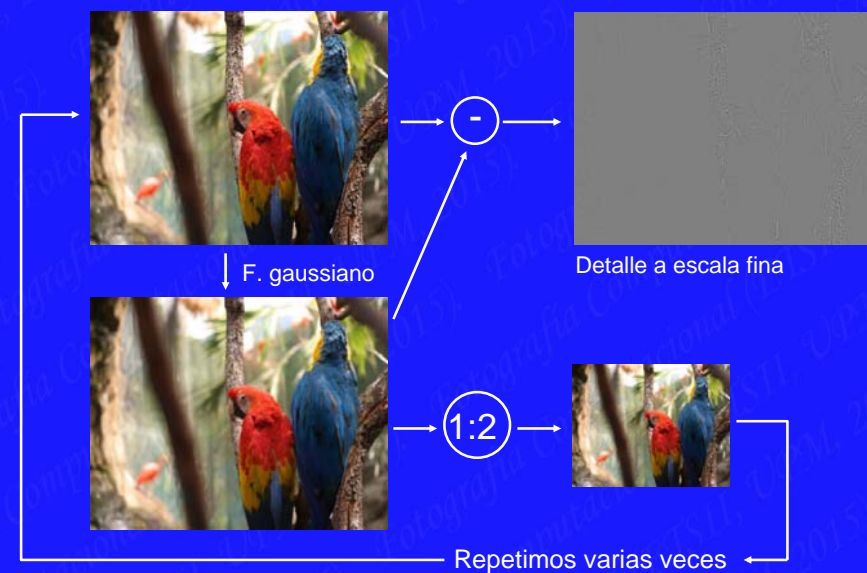
Ejemplo de piramide Gaussiana



ANTONIO TABERNERO GALÁN, 2015

FOTOGRAFÍA COMPUTACIONAL, ETSII, UPM

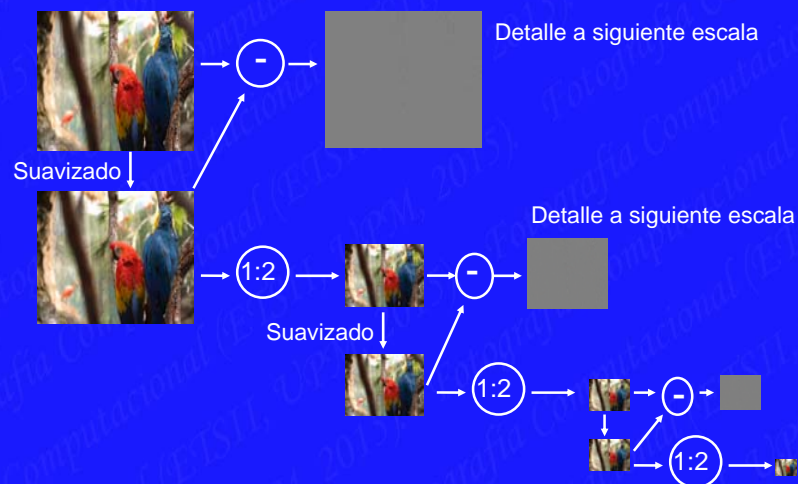
Ejemplo de piramide Laplaciana



ANTONIO TABERNERO GALÁN, 2015

FOTOGRAFÍA COMPUTACIONAL, ETSII, UPM

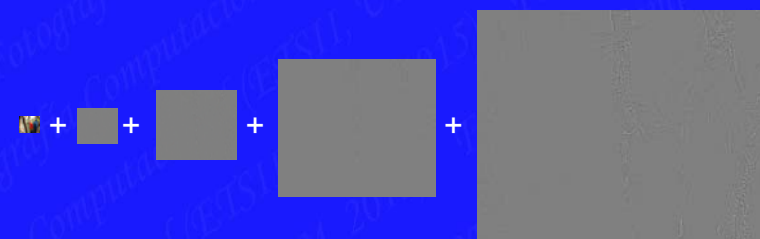
Ejemplo de piramide Laplaciana



ANTONIO TABERNERO GALÁN, 2015

FOTOGRAFÍA COMPUTACIONAL, ETSII, UPM

Piramide final



Una versión tamaño "sello" de la imagen original + sucesivas copias con el detalle correspondiente a cada escala.

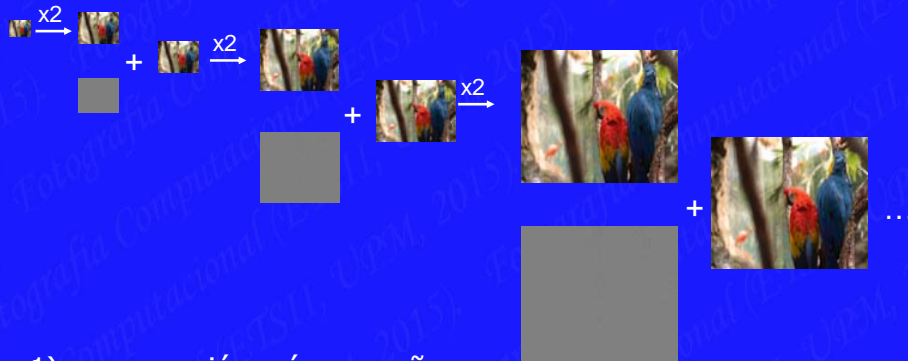
Su tamaño final es: $N^2 + \frac{N^2}{4} + \frac{N^2}{16} + \dots \approx \frac{4}{3} N^2$

un tercio mayor que la imagen original.

ANTONIO TABERNERO GALÁN, 2015

FOTOGRAFÍA COMPUTACIONAL, ETSII, UPM

Reconstrucción de la imagen



- 1) rec = versión más pequeña.
- 2) Ampliar rec por un factor 2
- 3) Sumar detalle correspondiente en esa escala.

Repetir 2) y 3) hasta llegar al tamaño original.

Filtrados no lineales

La operación con los vecinos puede ser una función no lineal.

Filtro de mediana: en vez de usar la media de los valores vecinos el valor de salida es la mediana.

Supongamos vecindad de píxeles 3x3:

94	99	92
96	90	90
92	91	0

Media = 82.8 Mediana = 92

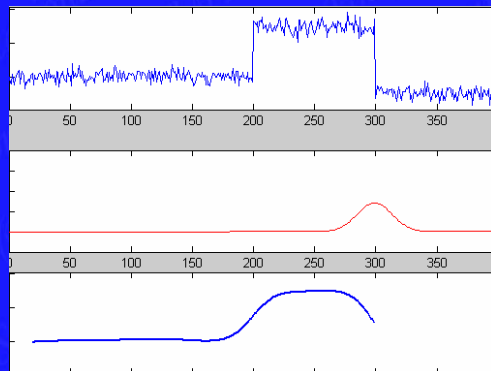
La media baja por la presencia del 0.

La mediana da un valor más representativo.

99
96
94
92
92
91
90
90
0

Filtro Bilateral

Un filtro gaussiano puede usarse para eliminar ruido. El problema es que altera profundamente los bordes.

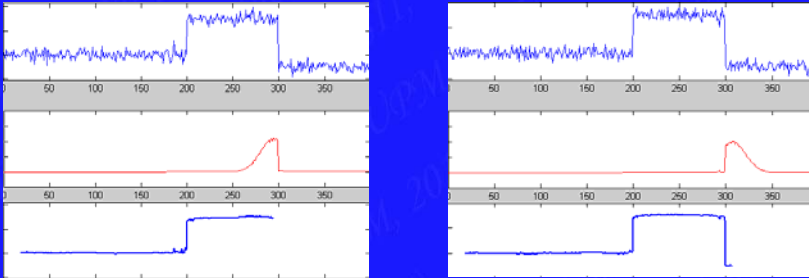


Deberíamos evitar promediar A TRAVÉS de los bordes.

Filtro Bilateral

- En un filtro gaussiano el factor de ponderación sólo depende de la distancia del píxel considerado a la posición actual.
- Todos los valores en un cierto radio se mezclan, aunque esos valores estén en el lado "equivocado" de un borde.
- Un filtro bilateral añade un segundo criterio: el factor de ponderación no sólo depende de la distancia a la posición actual sino también de la diferencia con el valor actual.
- Valores muy distintos del valor donde tengo mi filtro no contribuyen a la media, aunque estén muy cerca.
- El filtro aplica 2 gaussianas para calcular la ponderación:
 - En el dominio de la imagen (distancia entre píxeles)
 - En el rango de la imagen (valores de los píxeles).

Filtro Bilateral



EL FB evita "cruzar" bordes al promediar.
FB no es lineal: su implementación es más complicada y lenta.

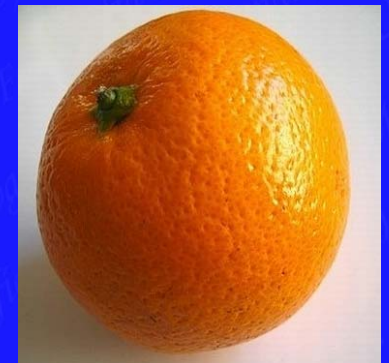
¡PERO ES MUY INTUITIVO Y TIENE MUCHAS APLICACIONES !

Referencias: Aurich 95, Smith 97, Tomasi 98]

Fusión imágenes

Queremos mezclar o "fusionar" varias imágenes

¿Cómo plantear el problema?



Fusión imágenes con una máscara

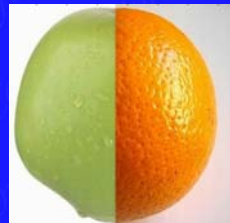
Una máscara (0/1) define qué imagen aparece en resultado.



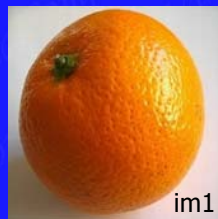
$$\text{res} = (1 - \text{mask}) * \text{im0} + \text{mask} * \text{im1}$$



mask

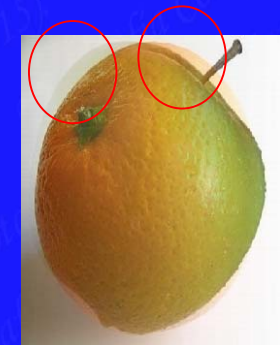
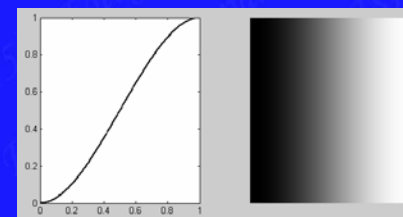


res



¿ Hacer transición más suave ?

Máscara más suave



Ahora aparecen copias "fantasmas".

¿Usar máscaras distintas según nivel de detalle?

- Abruptas para los detalles finos (escala fina)
- Suaves para los elementos que cambian lentamente.

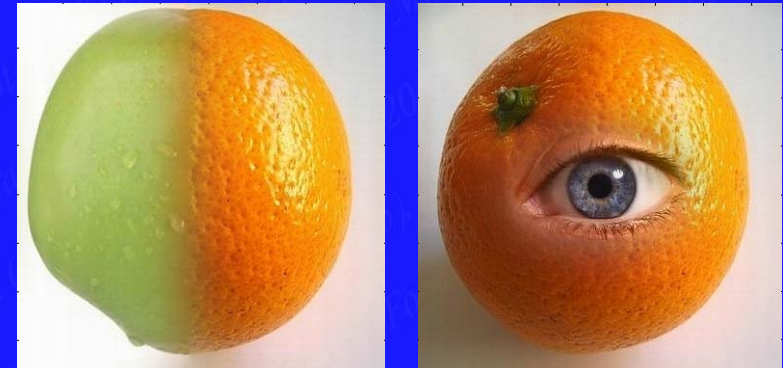
Fusión imágenes con pirámide Laplaciana

¿HAY TENER EN CUENTA LA ESCALA? ¡ USAR PIRÁMIDES !

PROCESO:

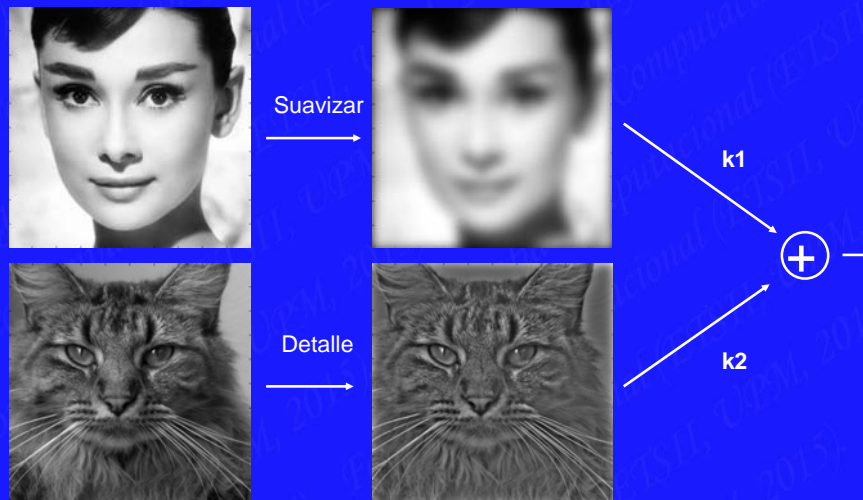
- 1) Partir de im_0 , im_1 del mismo tamaño.
- 2) Calcular pirámide laplaciana de ambas imágenes
- 3) Definir máscara mezcla del tamaño imágenes.
- 4) Calcular pirámide gaussiana de la máscara.
- 5) Mezclar las pirámides de las imágenes nivel por nivel con el correspondiente nivel de la pirámide máscara.
- 6) Colapsar la pirámide mezcla y obtener la imagen.

Resultado de fusión con pirámides



Imágenes híbridas

Oliva, Torralba, Schyns, "Hybrid Images", SIGGRAPH 2006



Imágenes híbridas



Se perciben de forma diferente según la distancia de observación:

- desde lejos (no se ve el detalle)
- desde cerca (predomina la percepción visual del detalle).

Para que queden bien hay que alinear previamente las imágenes.

Tema de la siguiente charla:

TRANSFORMACIONES
GEOMÉTRICAS.