



Introducción a visión por Computadoras con GPUs

Procesamiento de Imágenes. Filtros.

Jorge A. Sánchez
FaMAF, UNC, 2015

Semana 16-Mar-2015

1. Procesos puntuales
2. Ecualización de histogramas
3. Filtros lineales
4. Imágenes integrales
5. Representaciones multi-resolución
6. Transformaciones geométricas
7. Warping

Procesos puntuales

Operador puntual: $g(x) = h(f(x))$ or $g(x) = h(f_0(x), \dots, f_n(x)),$

Brillo-contraste: $g(x) = af(x) + b.$

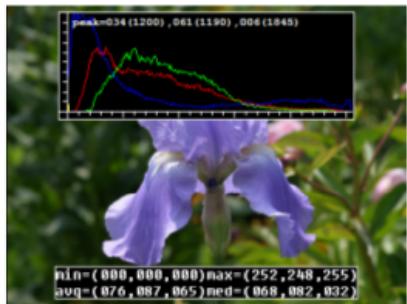
$$g(x) = a(x)f(x) + b(x),$$

blend lineal: $g(x) = (1 - \alpha)f_0(x) + \alpha f_1(x).$ $(0 \leq \alpha \leq 1)$

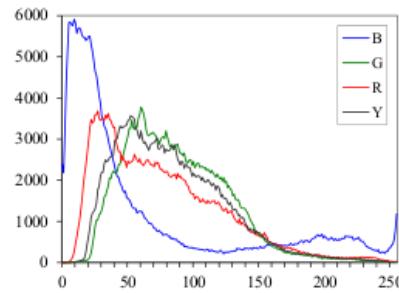
Matting y composición: $C = (1 - \alpha)B + \alpha F.$ $(\text{RGBA} \rightarrow \text{alpha matting})$

$$\begin{matrix} & \times & (1 - \alpha) & + & \alpha F \\ \begin{matrix} B \\ (a) \end{matrix} & & \begin{matrix} \alpha \\ (b) \end{matrix} & & \begin{matrix} \alpha F \\ (c) \end{matrix} & = & \begin{matrix} C \\ (d) \end{matrix} \end{matrix}$$

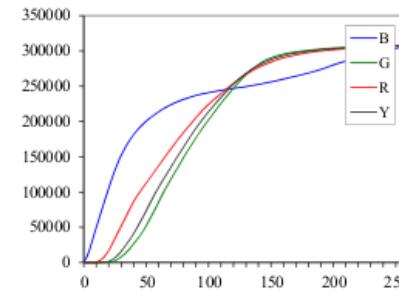
Ecuación de histogramas



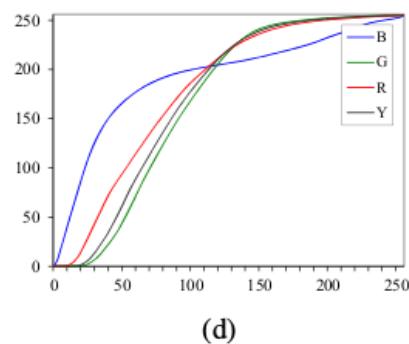
(a)



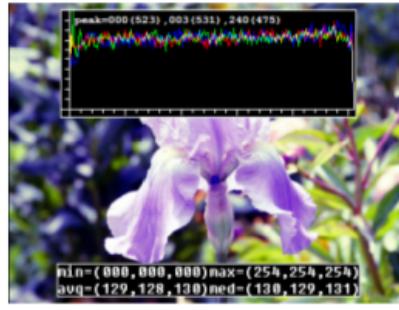
(b)



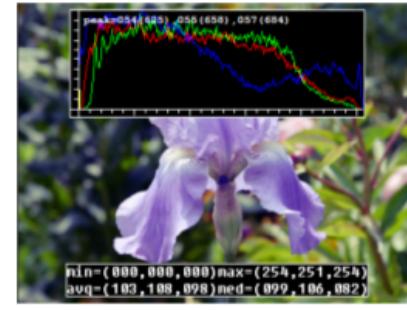
(c)



(d)



(e)



(f)

Objetivo: encontrar un mapeo f tal que la función resultante tenga un histograma plano

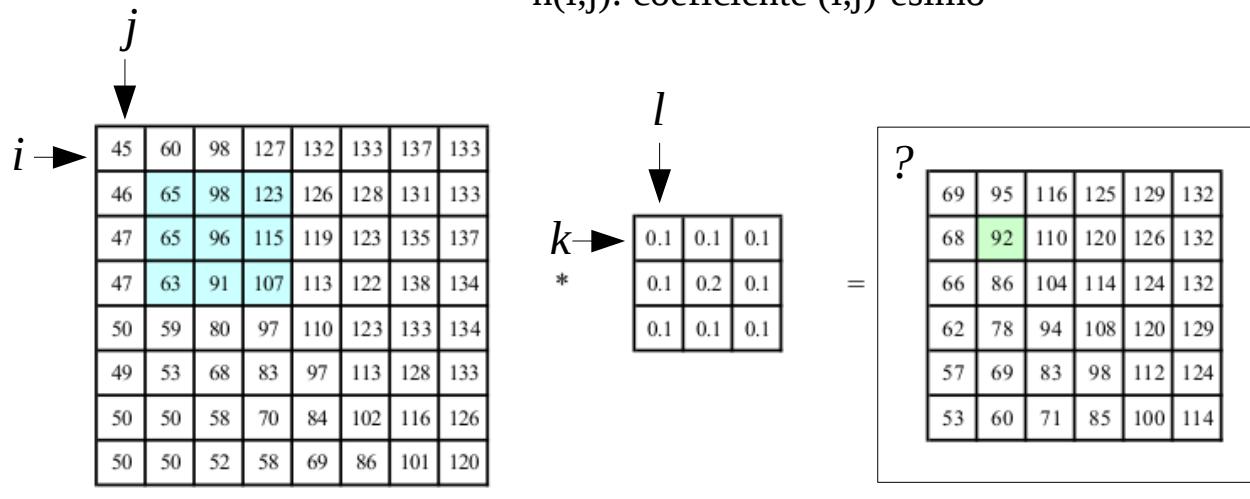
$$T(k) = \text{floor}((L - 1) \sum_{n=0}^k p_n).$$

p_n : proporción de píxeles con intensidad n
 L : número de valores posibles de f ($L=256$)

Filtros lineales

- Procesos puntuales vs. Operadores de vecindad
- Filtro lineal:
$$g(i, j) = \sum_{k,l} f(i + k, j + l)h(k, l).$$

h: *kernel* o máscara
h(i,j): coeficiente (i,j)-ésimo



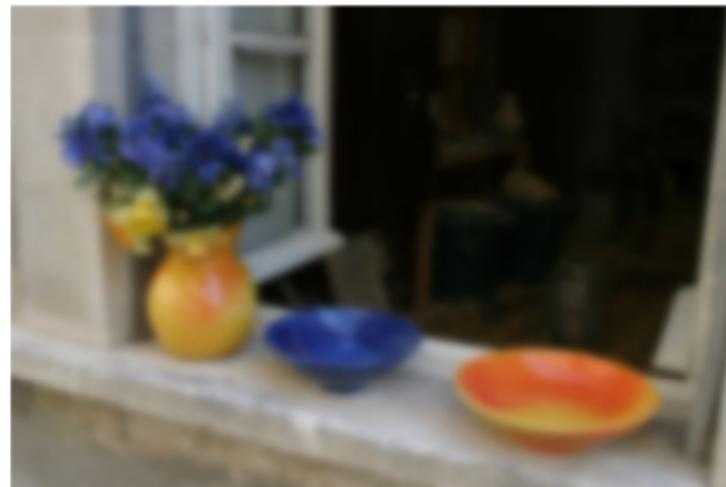
$f(x,y)$

$h(x,y)$

$g(x,y)$



(a)



(b)



(c)



(d)

(a) original, (b) blurred, (c) sharpened, (d) smoothed (edge preserving)

- Correlación:
$$g(i, j) = \sum_{k,l} f(i+k, j+l)h(k, l).$$

$$g = f \otimes h. \quad (\text{h: kernel o máscara})$$
- Convolución:
$$g(i, j) = \sum_{k,l} f(i-k, j-l)h(k, l) = \sum_{k,l} f(k, l)h(i-k, j-l),$$

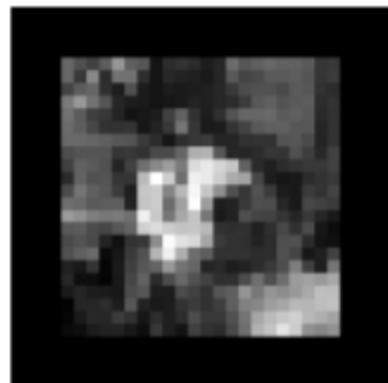
$$g = f * h, \quad (\text{h: función de respuesta al impulso})$$

Son operadores LSI (*linear shift-invariant*):

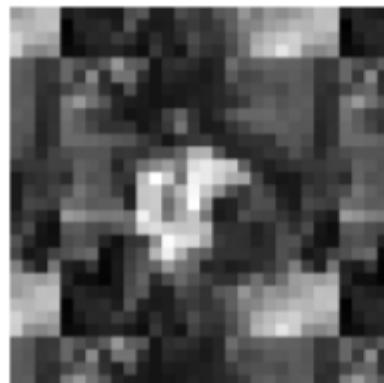
- Principio de superposición:
$$h \circ (f_0 + f_1) = h \circ f_0 + h \circ f_1,$$
- Homogeneidad:
$$g(i, j) = f(i+k, j+l) \Leftrightarrow (h \circ g)(i, j) = (h \circ f)(i+k, j+l),$$

Efectos de borde: *padding*

- Zero
- Valor constante
- *Clamp*: repetir los pixels de borde
- *Wrap*: repetición en forma cíclica (imagen=toroide)
- Espejado (mirror)
- ...



zero



wrap



clamp



mirror

Filtros lineales separables

- Correlación / convolución como multiplicación matriz-vector:

$$g * h \Leftrightarrow \mathbf{Hf}$$

$$\begin{bmatrix} 72 & 88 & 62 & 52 & 37 \end{bmatrix} * \begin{bmatrix} 1/4 & 1/2 & 1/4 \end{bmatrix} \Leftrightarrow \frac{1}{4} \begin{bmatrix} 2 & 1 & . & . & . \\ 1 & 2 & 1 & . & . \\ . & 1 & 2 & 1 & . \\ . & . & 1 & 2 & 1 \\ . & . & . & 1 & 2 \end{bmatrix} \begin{bmatrix} 72 \\ 88 \\ 62 \\ 52 \\ 37 \end{bmatrix}$$

- Convolución 2D: K^2 operaciones (mult.+suma) por píxel
- Separabilidad: $K^2 \rightarrow 2K$ (2 convoluciones 1D)
- Como saber si un *kernel* 2D es separable?
 - Inspección
 - Singular Value Decomposition (SVD): $\mathbf{K} = \sum_i \sigma_i \mathbf{u}_i \mathbf{v}_i^T$

Filtros lineales. Ejemplos

$$\frac{1}{K^2} \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & 1 & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix}$$

$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}$$

$$\frac{1}{256} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \\ 4 & 16 & 24 & 16 & 4 \\ 6 & 24 & 36 & 24 & 6 \\ 4 & 16 & 24 & 16 & 4 \\ 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

$$\frac{1}{8} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$\frac{1}{4} \begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

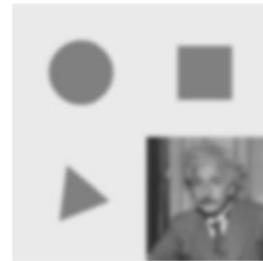
$$\frac{1}{K} \begin{bmatrix} 1 & 1 & \cdots & 1 \end{bmatrix}$$

$$\frac{1}{4} \begin{bmatrix} 1 & 2 & 1 \end{bmatrix}$$

$$\frac{1}{16} \begin{bmatrix} 1 & 4 & 6 & 4 & 1 \end{bmatrix}$$

$$\frac{1}{2} \begin{bmatrix} -1 & 0 & 1 \end{bmatrix}$$

$$\frac{1}{2} \begin{bmatrix} 1 & -2 & 1 \end{bmatrix}$$



(a) box, $K = 5$

(b) bilinear

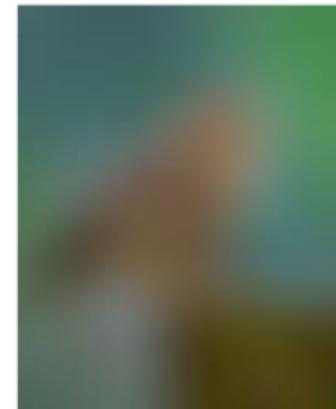
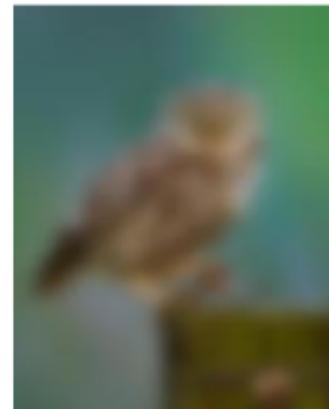
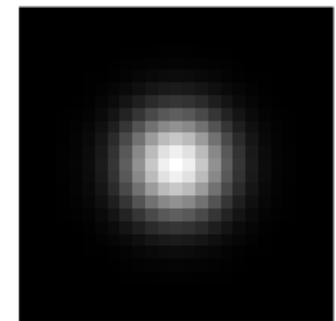
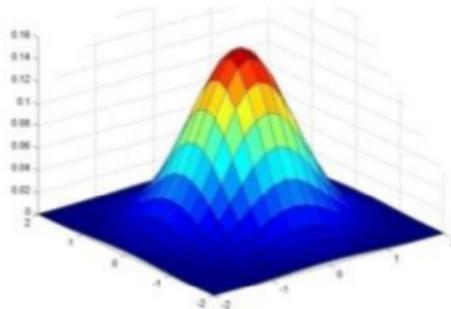
(c) “Gaussian”

(d) Sobel

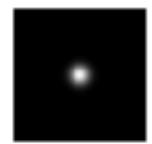
(e) corner

Filtro Gaussiano

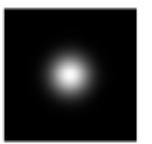
$$G_\sigma = \frac{1}{2\pi\sigma^2} e^{-\frac{(x^2+y^2)}{2\sigma^2}}$$



$\sigma = 1$ pixel



$\sigma = 5$ pixels



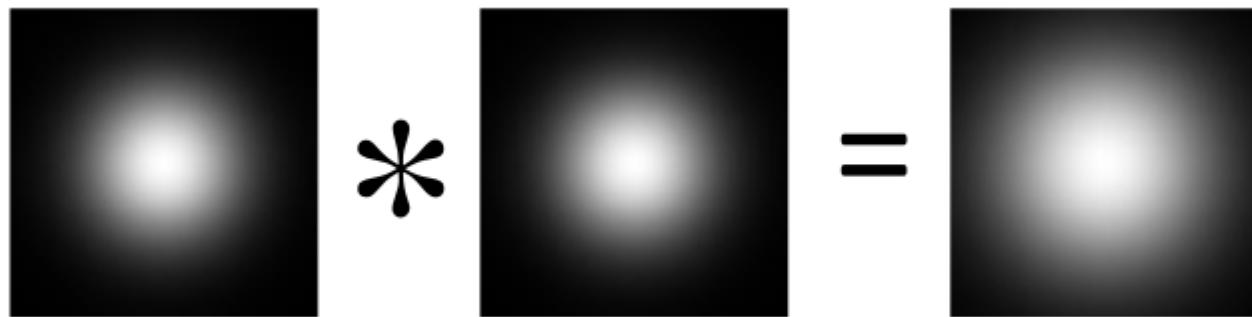
$\sigma = 10$ pixels



$\sigma = 30$ pixels

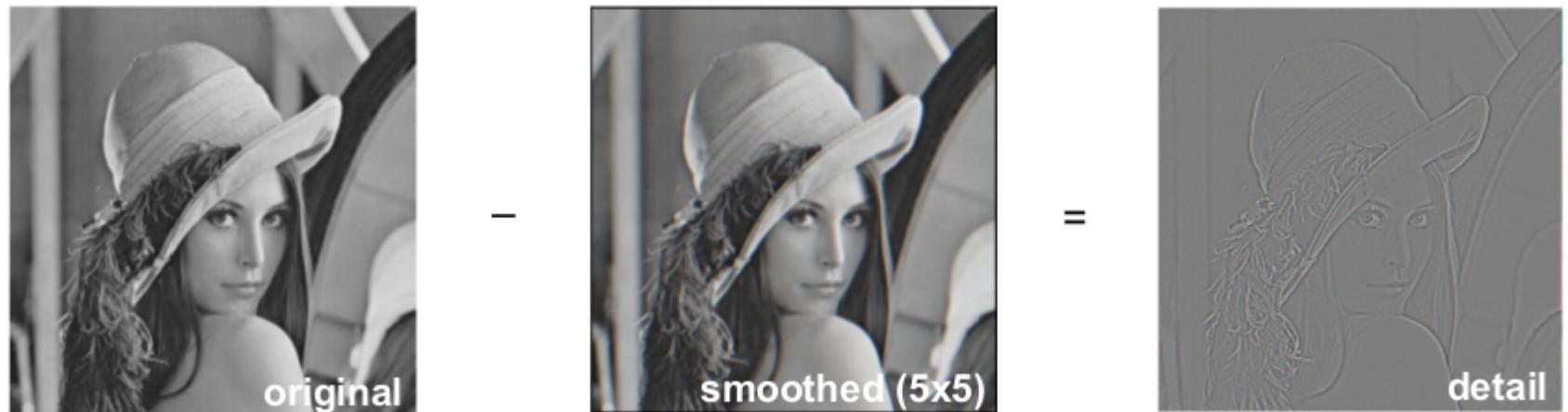
Filtro Gaussiano

- Separable
- Remueve las componentes de alta frecuencia
- Convolución de 2 kernels Gaussianos es otro kernel Gaussiano



→ convolucionar dos veces con un kernel de parámetro σ es lo mismo que una convolución con un kernel de $\sqrt{2}\sigma$

El suavizado elimina frecuencias altas



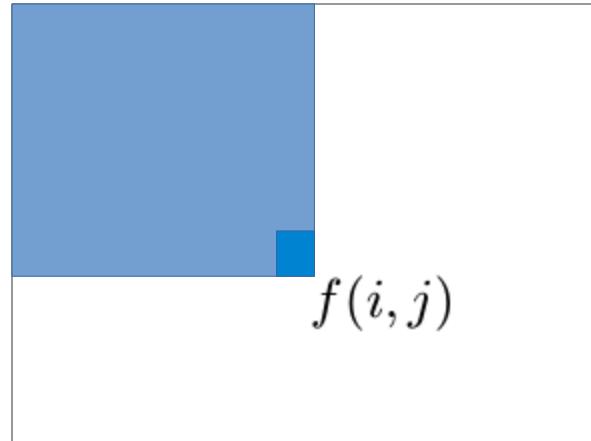
Sharpening: realzar los “detalles” en la imagen original



$$g_{\text{sharp}} = f + \gamma(f - h_{\text{blur}} * f)$$

Imágenes integrales

$$s(i, j) = \sum_{k=0}^i \sum_{l=0}^j f(k, l)$$



Cómputo:

$$s(i, j) = s(i - 1, j) + s(i, j - 1) - s(i - 1, j - 1) + f(i, j)$$

Suma en $[i_0, i_1] \times [j_0, j_1]$:

$$\begin{aligned} S(i_0 \dots i_1, j_0 \dots j_1) &= \sum_{i=i_0}^{i_1} \sum_{j=j_0}^{j_1} s(i_1, j_1) - s(i_1, j_0 - 1) \\ &\quad - s(i_0 - 1, j_1) + s(i_0 - 1, j_0 - 1) \end{aligned}$$

3	2	7	2	3
1	5	1	3	4
5	1	3	5	1
4	3	2	1	6
2	4	1	4	8

(a) $S = 24$

3	5	12	14	17
4	11	19	24	31
9	17	28	38	46
13	24	37	48	62
15	30	44	59	81

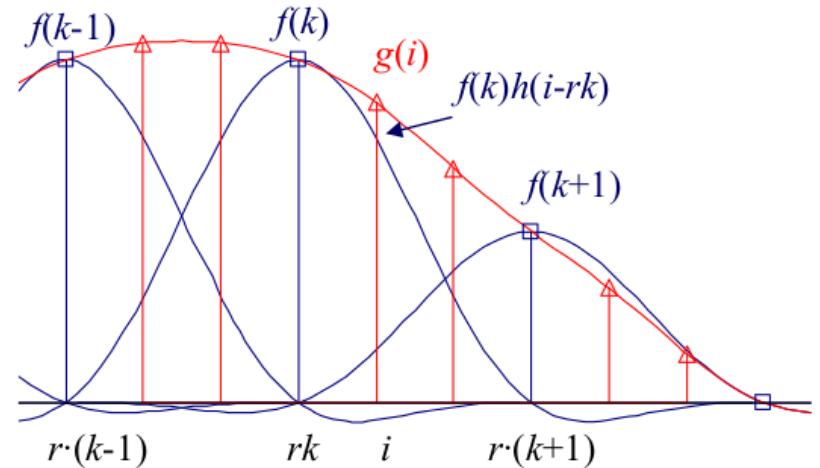
(b) $s = 28$

3	5	12	14	17
4	11	19	24	31
9	17	28	38	46
13	24	37	48	62
15	30	44	59	81

(c) $S = 24$

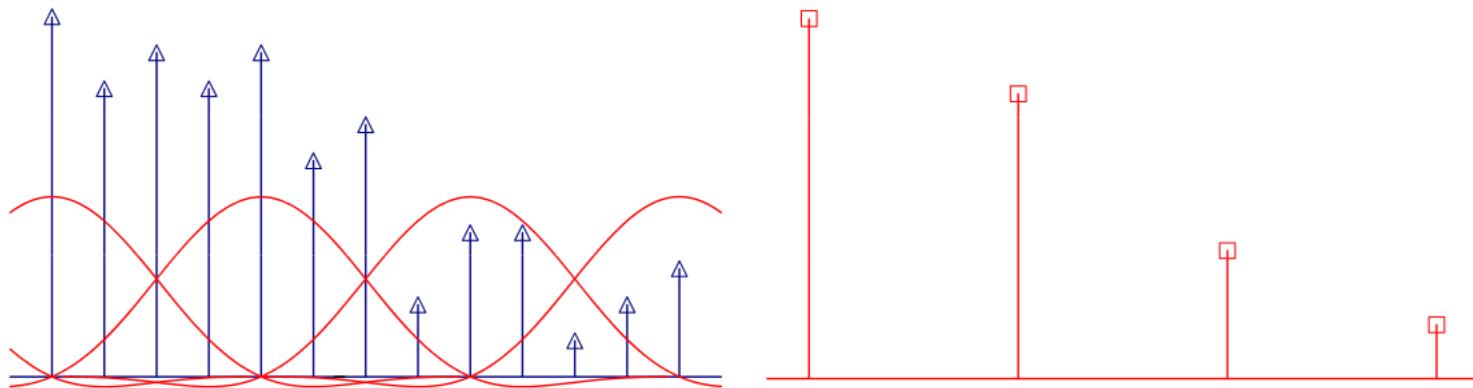
- **Interpolación**

$$g(i, j) = \sum_{k,l} f(k, l) h(i - rk, j - rl)$$



- **Decimación**

$$g(i, j) = \frac{1}{r} \sum_{k,l} f(k, l) h(i - k/r, j - l/r)$$

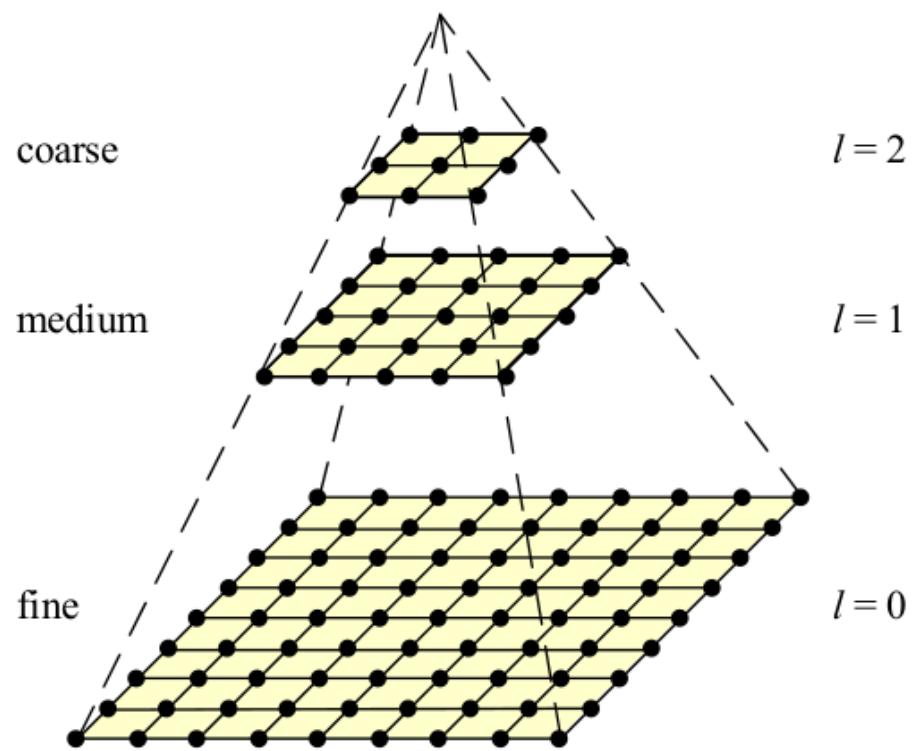


Representaciones multi-resolución



Usos:

- compresión
- detección de puntos de interés
- registro



Jerarquía de transformaciones 2D

- Traslación:

$$\mathbf{x}' = [\begin{array}{cc} \mathbf{I} & \mathbf{t} \end{array}] \bar{\mathbf{x}}$$

- Rotación + traslación:

$$\mathbf{x}' = [\begin{array}{cc} \mathbf{R} & \mathbf{t} \end{array}] \bar{\mathbf{x}}$$

$$\mathbf{R} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

$$\mathbf{R}\mathbf{R}^T = \mathbf{I} \text{ and } |\mathbf{R}| = 1$$

- Rotación y escalado:

$$\mathbf{x}' = [\begin{array}{cc} s\mathbf{R} & \mathbf{t} \end{array}] \bar{\mathbf{x}}$$

- Afín:

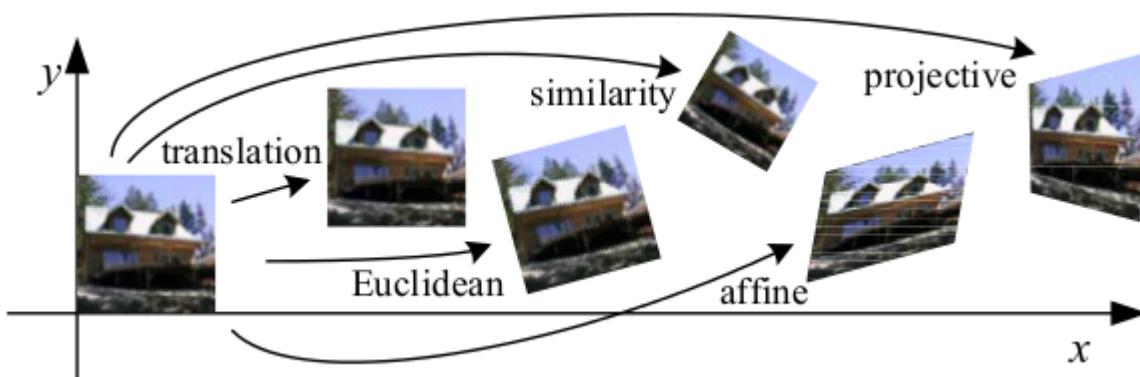
$$\mathbf{x}' = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \end{bmatrix} \bar{\mathbf{x}}$$

- Proyectiva:

$$\tilde{\mathbf{x}}' = \tilde{\mathbf{H}} \tilde{\mathbf{x}}$$

$$x' = \frac{h_{00}x + h_{01}y + h_{02}}{h_{20}x + h_{21}y + h_{22}}$$

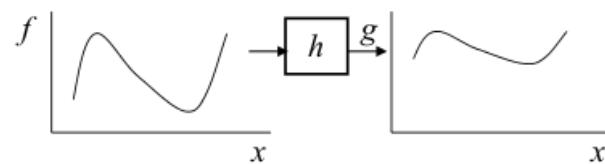
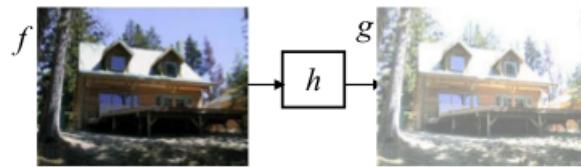
$$y' = \frac{h_{10}x + h_{11}y + h_{12}}{h_{20}x + h_{21}y + h_{22}}$$



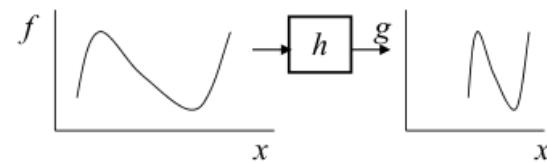
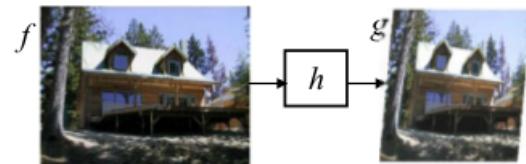
Transformation	Matrix	# DoF	Preserves	Icon
translation	$[\ I \ \ t \]_{2 \times 3}$	2	orientation	
rigid (Euclidean)	$[\ R \ \ t \]_{2 \times 3}$	3	lengths	
similarity	$[\ sR \ \ t \]_{2 \times 3}$	4	angles	
affine	$[\ A \]_{2 \times 3}$	6	parallelism	
projective	$[\ \tilde{H} \]_{3 \times 3}$	8	straight lines	

Transformaciones geométricas

Procesos puntuales: $g(x) = h(f(x))$ (rango)



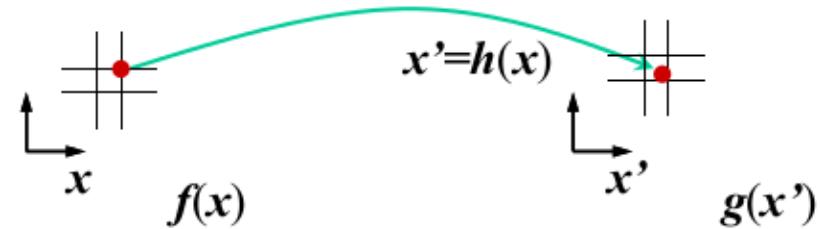
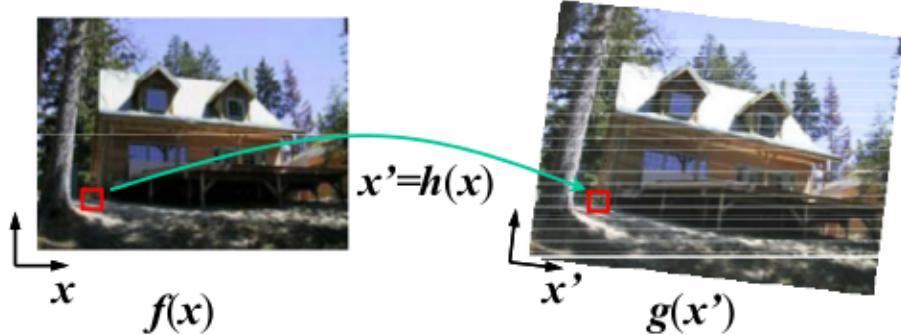
Transf. geométricas: $g(x) = f(h(x))$ (dominio)



procedure *forwardWarp*(f, h , **out** g):

For every pixel x in $f(x)$

1. Compute the destination location $x' = h(x)$.
2. Copy the pixel $f(x)$ to $g(x')$.

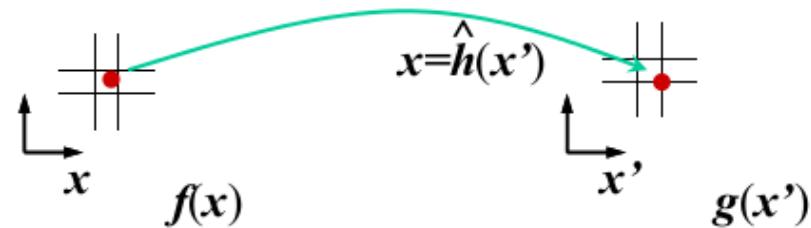
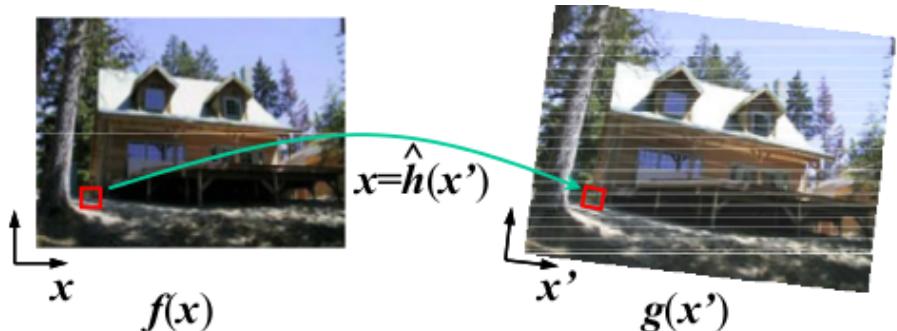


¿problemas?

procedure *inverseWarp*(f, h , **out** g):

For every pixel x' in $g(x')$

1. Compute the source location $x = \hat{h}(x')$
2. Resample $f(x)$ at location x and copy to $g(x')$



Próxima clase: **detección de *features*** y matching (cap. 4)

