Tratamiento de Señales

Version 2025-1

Transformaciones geométricas e Interpolación

[Capítulo 3]

Dr. José Ramón Iglesias

DSP-ASIC BUILDER GROUP

Director Semillero TRIAC

Ingenieria Electronica

Universidad Popular del Cesar

Contenido

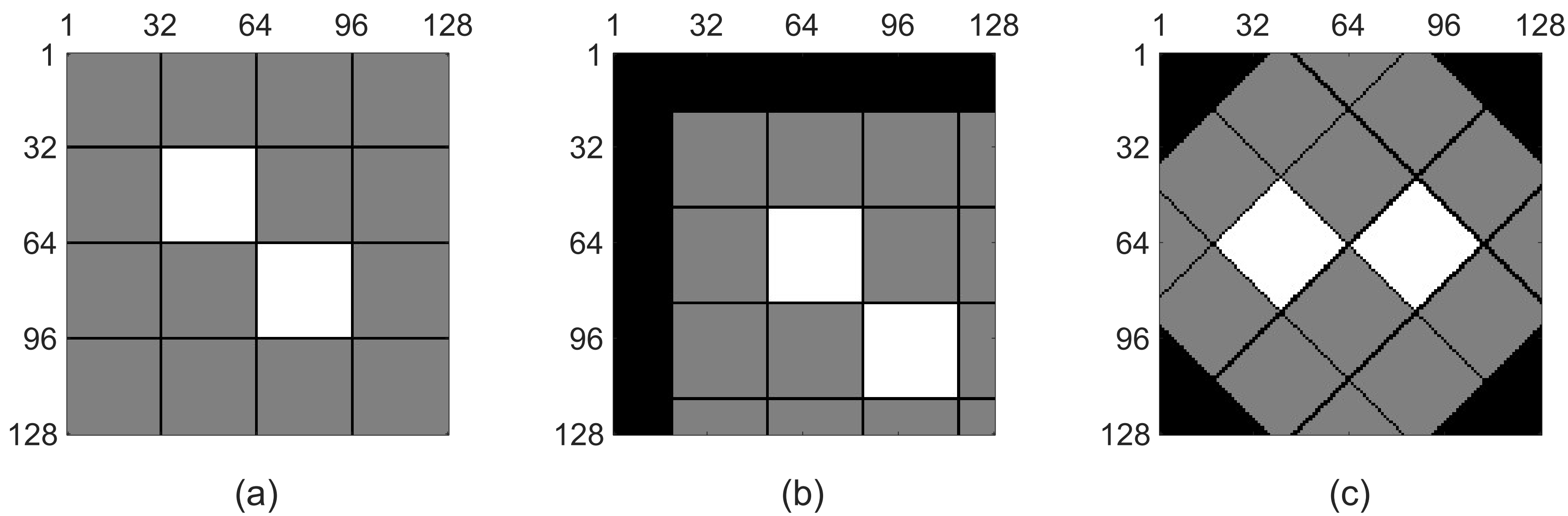
Introducción

- *Transformación geométrica. F*unción que modifíca la relación espacial entre píxeles en una imagen.
- Clasificación de acuerdo con las propiedades que se preservan:
    - *Rígida* preserva distancias.
    - *Semejanza* preserva ngulos.
    - *Afinidad* preserva paralelismo.
    - *Proyectiva* preserva colinearidad.
- La transformación de coordenadas se expresa como:

$$(\hat{x}, \hat{y}) = \mathbf{T}\{(x, y)\} \qquad (1)$$

donde (x, y) son coordenadas en la imagen original y $(\hat{x}, \hat{y})$ son coordenadas en la imagen transformada.

Transformaciones geométricas



Las transformaciones rígidas preservan la distancia entre cualquier par de puntos p y q de la imagen (isometría). Los objetos tienen la misma forma y tamaño antes y después de la transformación. (a) Imagen original, (b) traslación y (c) rotación.

Operaciones geométricas

Las operaciones geométricas modifican la distribución espacial de los píxeles en una imagen. Un píxel con coordenadas (x, y), se traslada a las coordenadas (x', y') de acuerdo a la transformación lineal establecida por la matriz .

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \mathbf{T} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} t_{11} & t_{12} \\ t_{21} & t_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Transformaciones afines

Las transformaciones afines son un tipo de operaciones geométricas que preservan los puntos, líneas rectas y planos en una imagen.

Las transformaciones afines se representan de manera general como una matriz de 3×3, que dependiendo de los valores de sus coeficientes realiza las operaciones de:

- Escalamiento
- Rotación
- Traslación
- Shearing

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \mathbf{A} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Transformaciones afines

Una característica de las transformaciones afines, y de su representación como matrices, es que se pueden realizar múltiples transformaciones en una sola operación.

Esto se obtiene a partir de la multiplicación previa de las matrices de transformación.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \mathbf{A} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Transformaciones afines

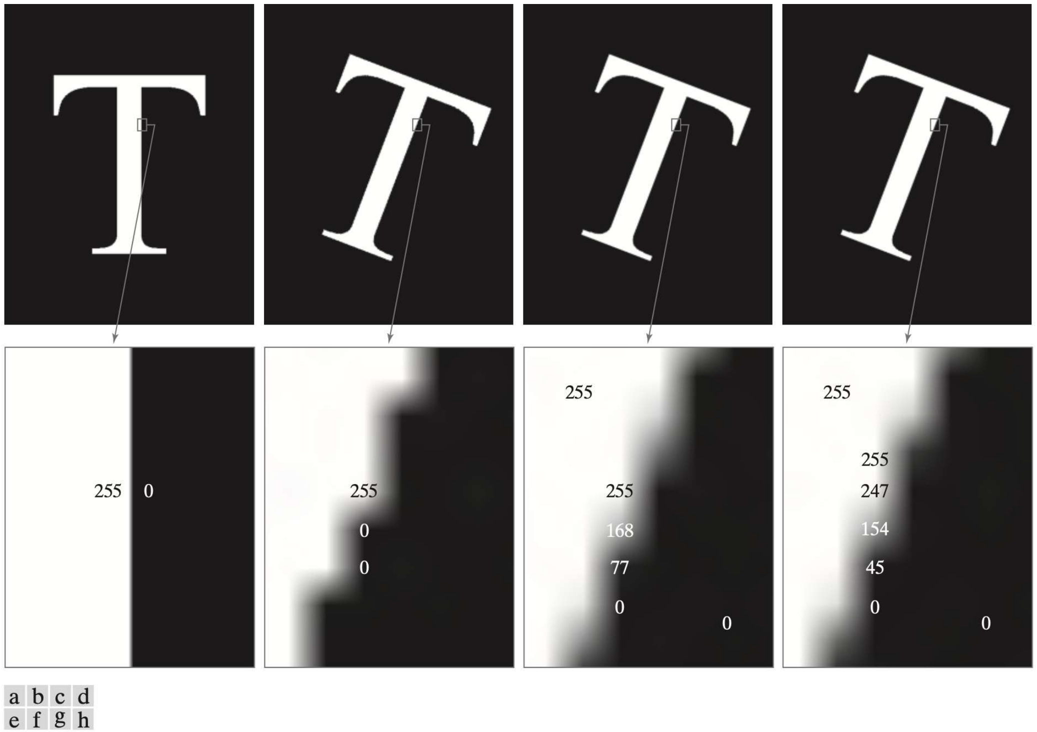
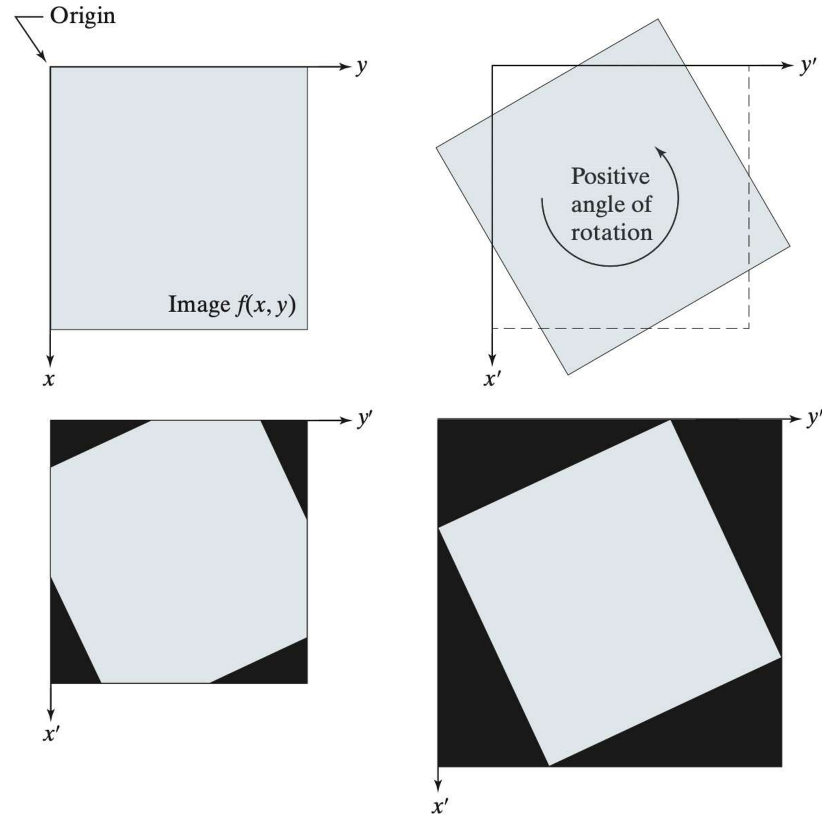
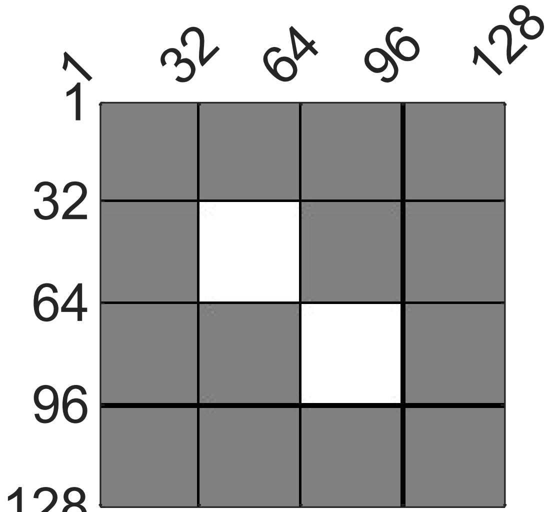


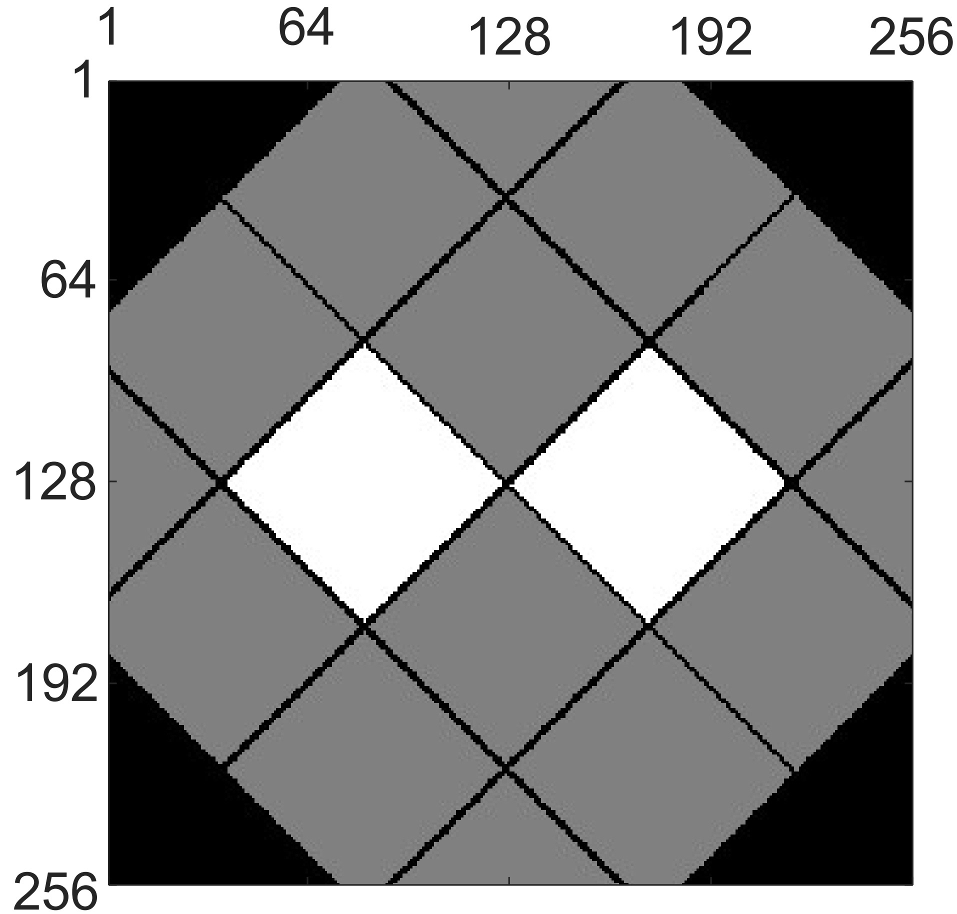
FIGURE 2.40 (a) A 541 × 421 image of the letter T. (b) Image rotated −21° using nearest-neighbor interpolation for intensity assignments. (c) Image rotated −21° using bilinear interpolation. (d) Image rotated −21° using bicubic interpolation. (e)-(h) Zoomed sections (each square is one pixel, and the numbers shown are intensity values).



Transformaciones geométricas



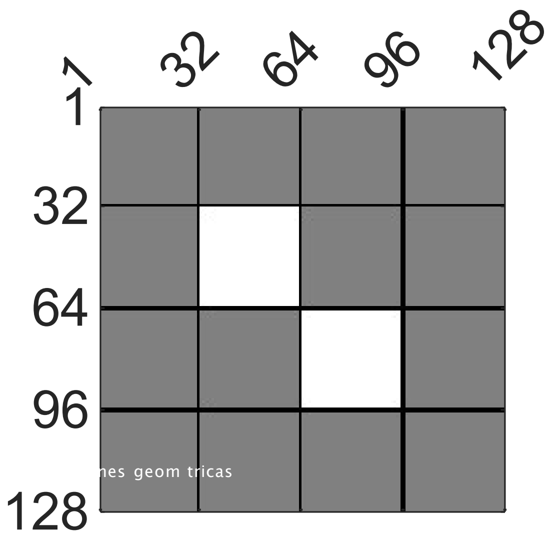
(a)



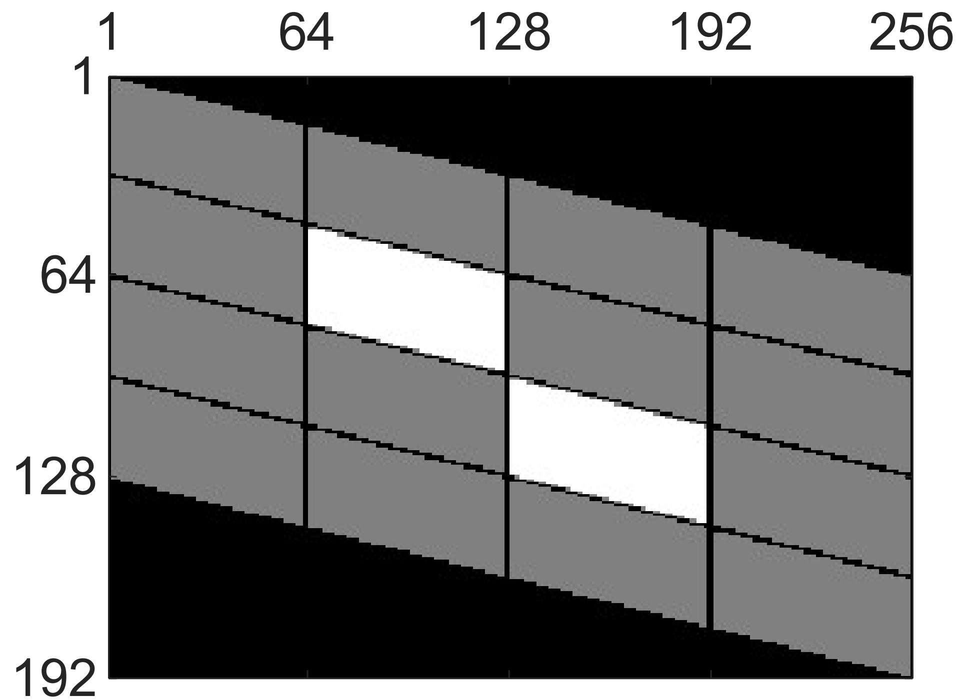
(b)

Las transformaciones semejantes, a partir de un punto fijo, multiplican todas las distancias por un mismo factor, de manera que se preservan los ángulos de los objetos en la imagen. (a) Imagen original y (b) imagen escalada uniformemente y rotada.

Transformaciones geométricas



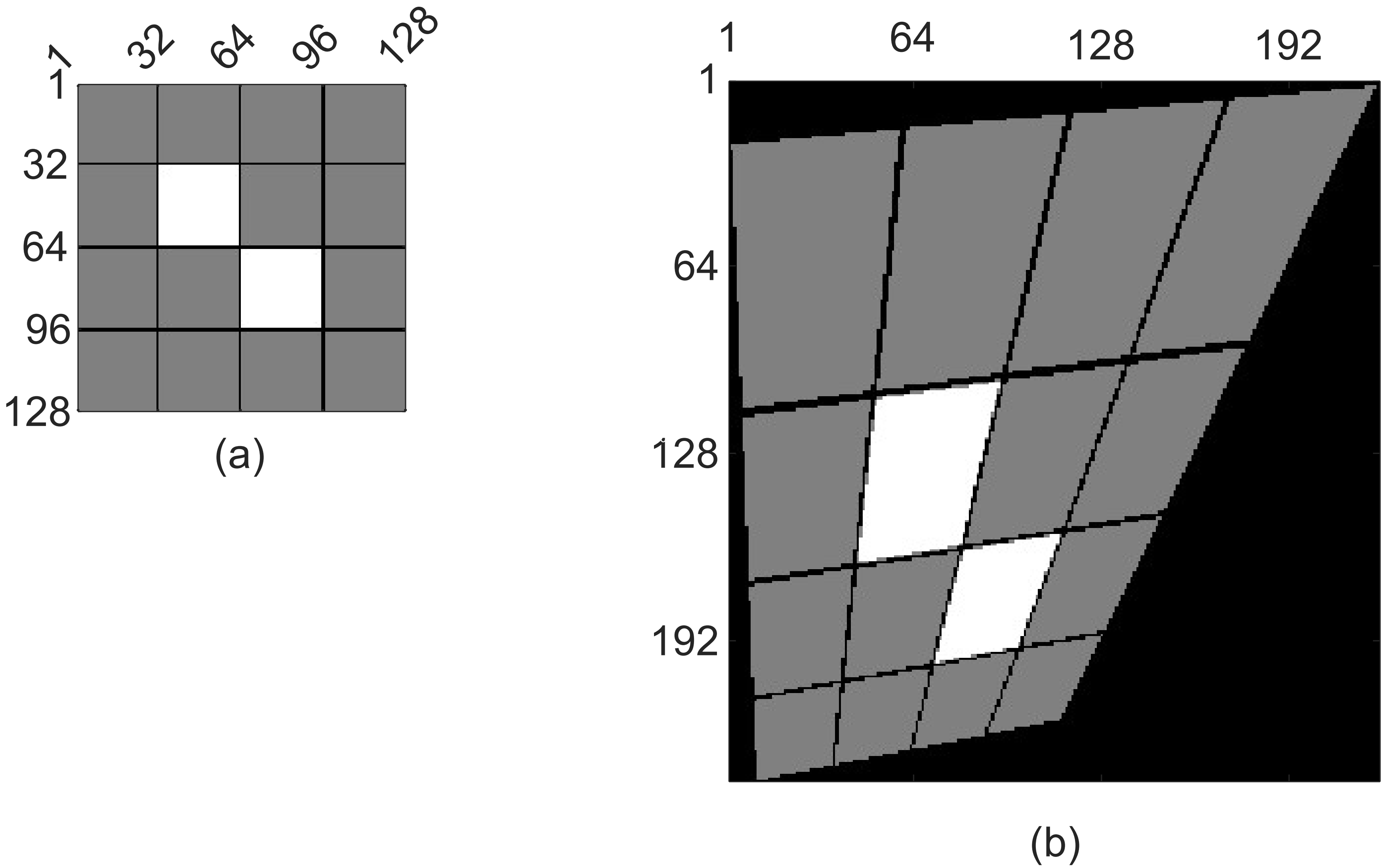
(a)



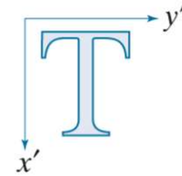
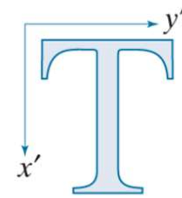
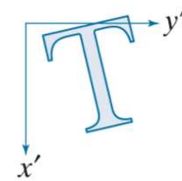
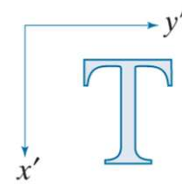
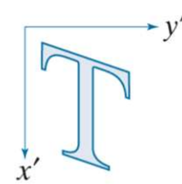
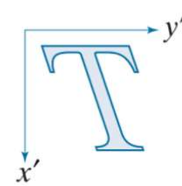
(b)

Las transformaciones afines preservan las relaciones de las distancias entre dos puntos cualquiera de la imagen, manteniendo así el paralelismo entre segmentos de rectas. (a) Imagen original y (b) imagen escalada con diferente factor en cada eje coordenado.

Transformaciones geomé tricas

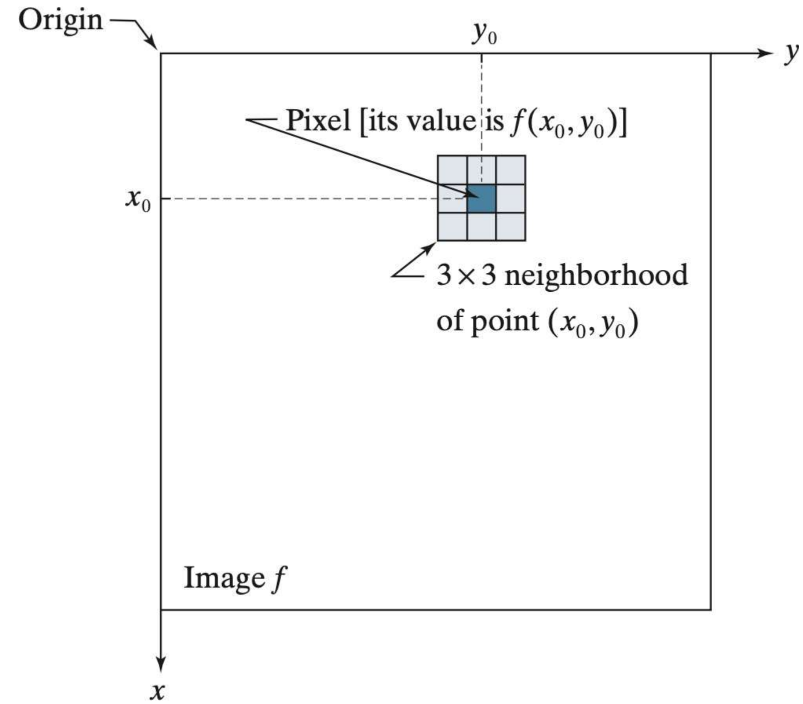


Las transformaciones proyectiva preservan la colinearidad. (a) Imagen original y (b) imagen con proyección a partir de un punto de fuga.

Nombre de la transformación	Matriz afin	Ecuaciones de coordenadas	Ejemplo
Identidad	$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x$ $y' = y$	
Escalamiento/Reflexión (Para reflexión, poner uno de los factores de escala en -1 y el otro en 0)	$\begin{bmatrix} c_x & 0 & 0 \\ 0 & c_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = c_x x$ $y' = c_y y$	
Rotación (entorno al origen)	$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x\cos\theta - y\sin\theta$ $y' = x\sin\theta + y\cos\theta$	
Traslación	$\begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x + t_x$ $y' = y + t_y$	
Sesgo (vertical)	$\begin{bmatrix} 1 & s_v & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x + s_v y$ $y' = y$	
Sesgo (horizontal)	$\begin{bmatrix} 1 & 0 & 0 \\ s_h & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$	$x' = x$ $y' = s_h x + y$	

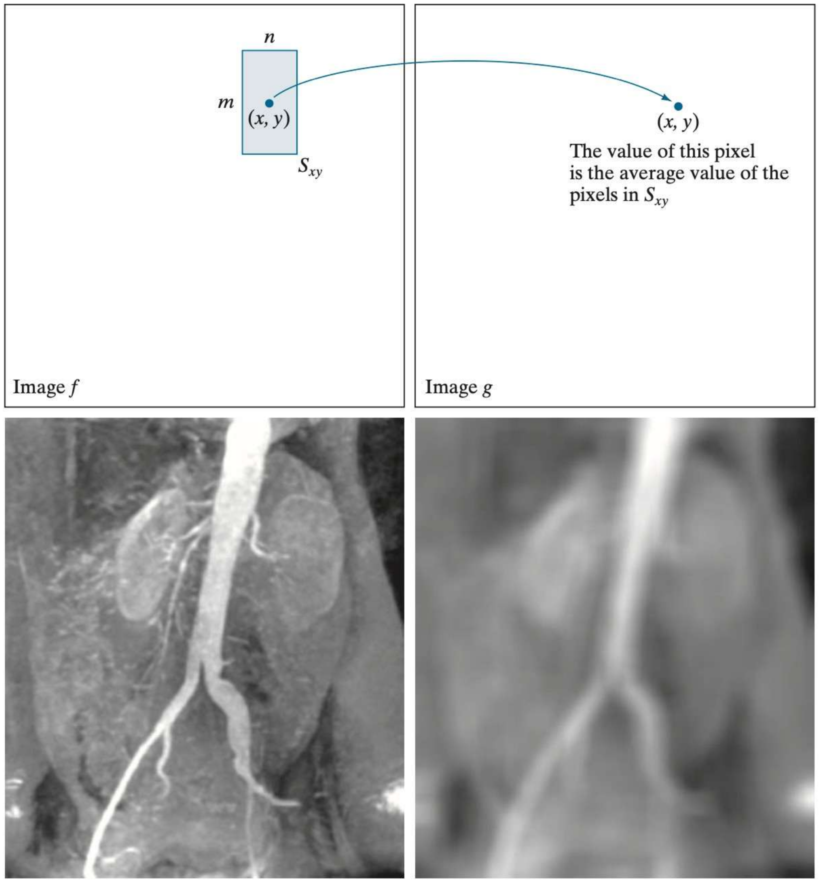
Operaciones locales

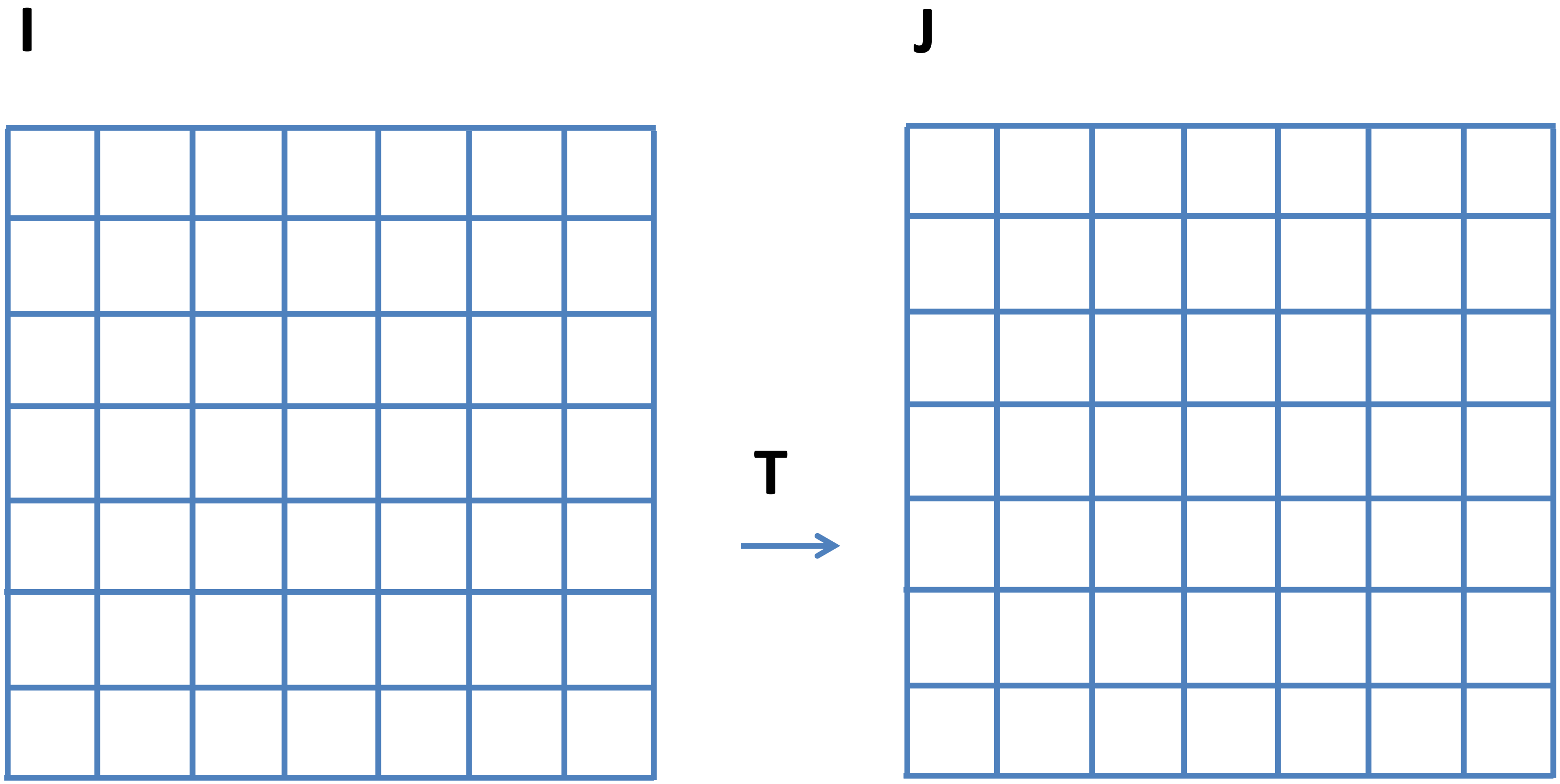
Al aplicar una operación local sobre el píxel con coordenadas (x_0, y_0), el nuevo valor de intensidad es el resultado de aplicar una operación sobre la región S_{xy} que incluye a y sus vecinos.

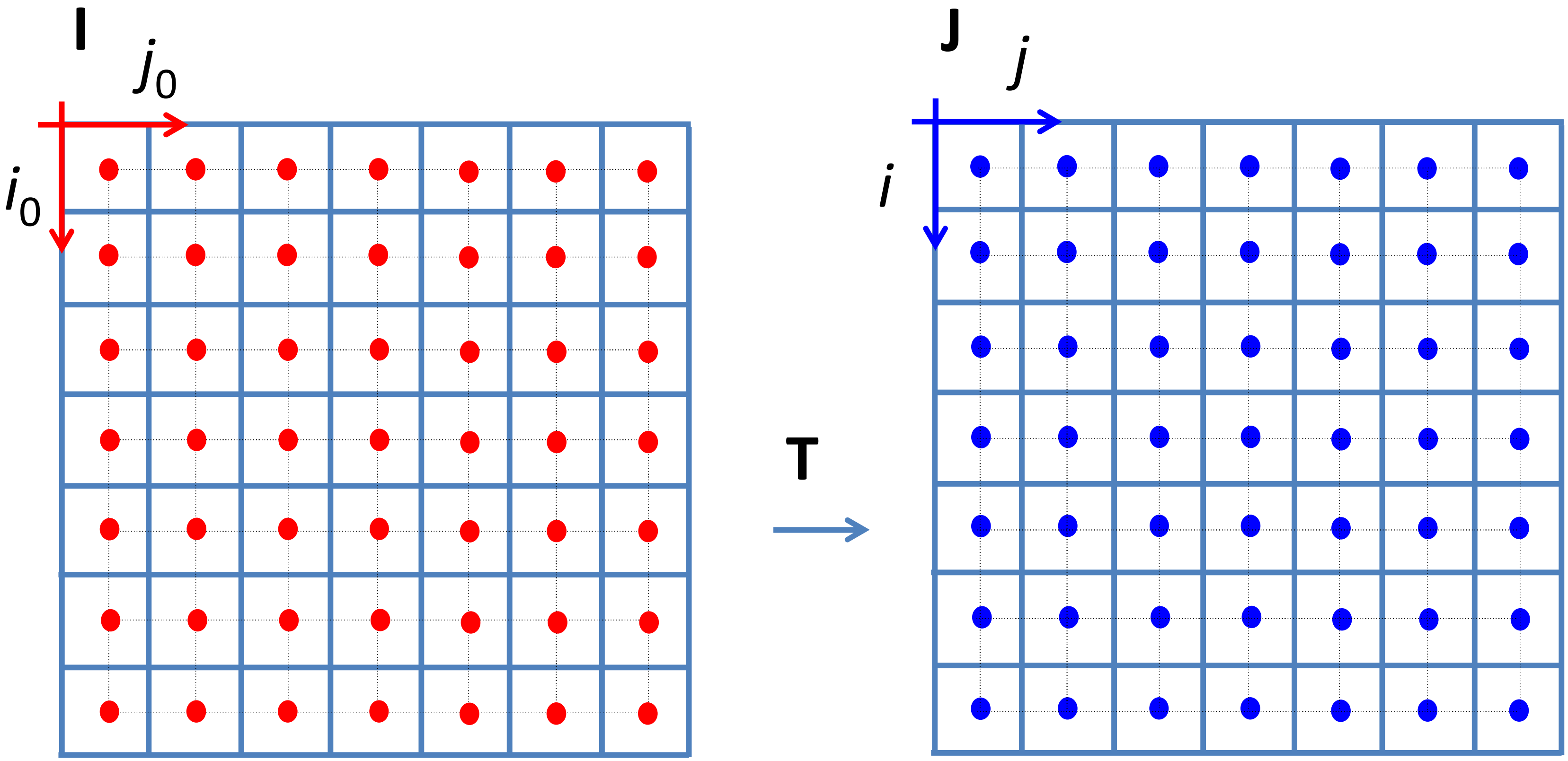


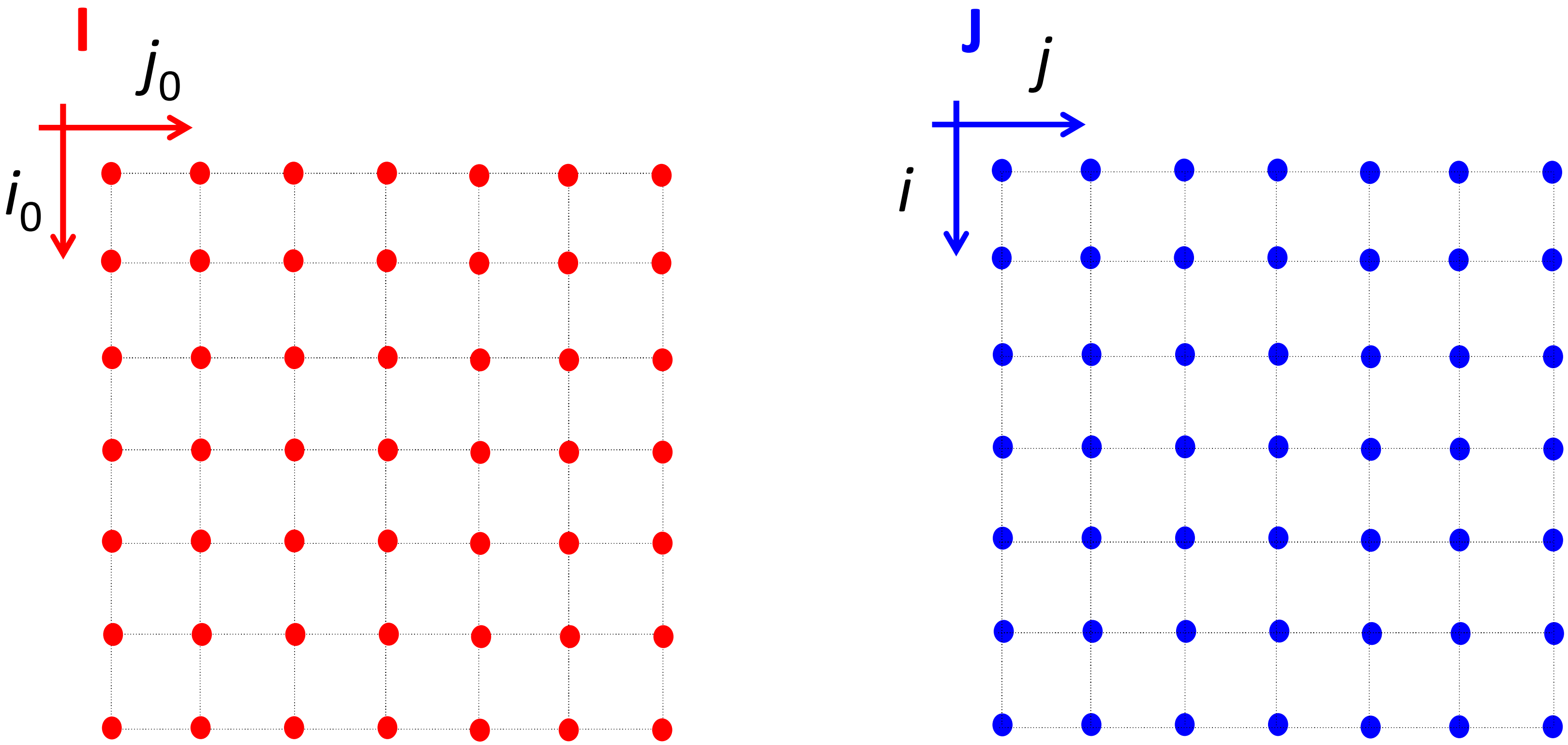
Operaciones locales

- Un ejemplo con la operación de promedio sobre una región de
- 41×41 píxeles. Resultado: imagen borrosa (blur) con menos detalles.



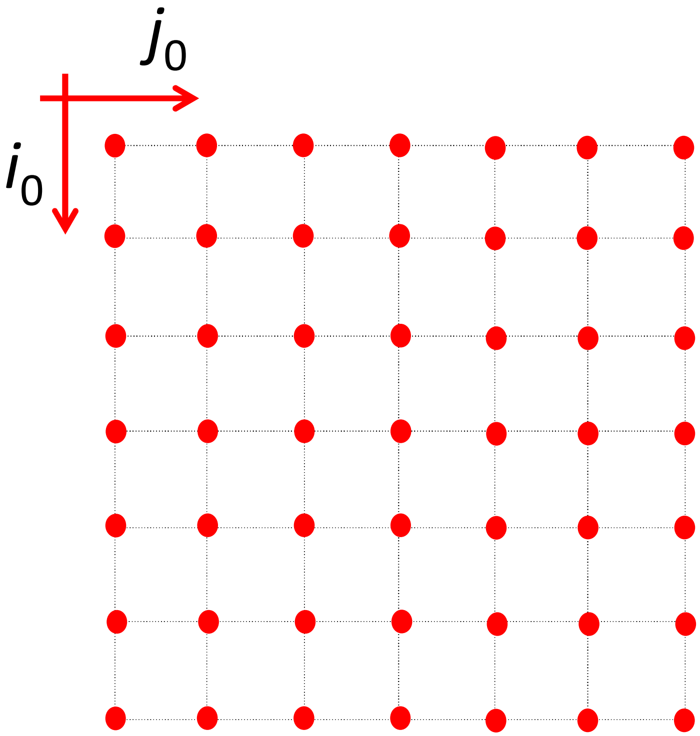

I
T
J

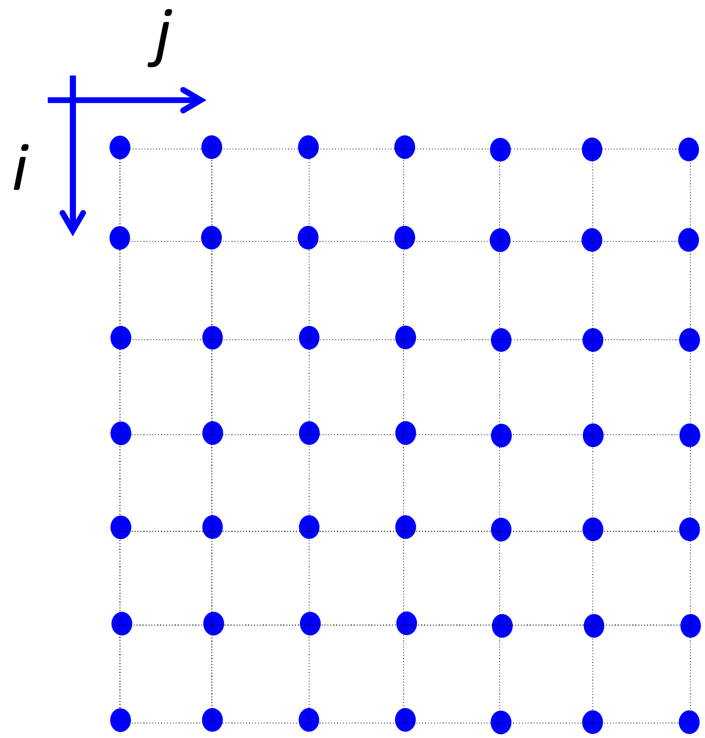

I
j_0
i_0
T
J
j
i



Coordinate Transformation: $i_0 = f_i(i,j)$

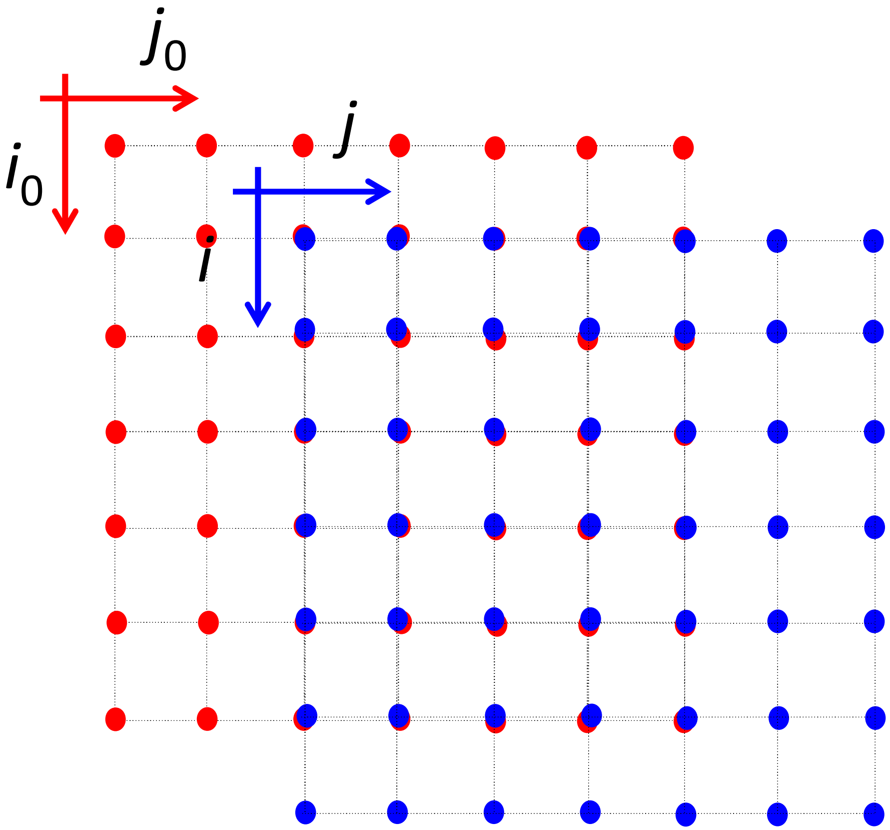
$$j_0 = f_j(i,j)$$





Coordinate Transformation: $i_0 = i + 1$
(Example: Translation)

$$j_0 = j + 2$$



Algorithm:

1) For each (i,j) of J compute (i_0,j_0).
2) $J(i,j) = I(i_0,j_0)$

Coordinate Transformation:
(Example: Translation)

$$i_0 = i + 1$$

$$j_0 = j + 2$$

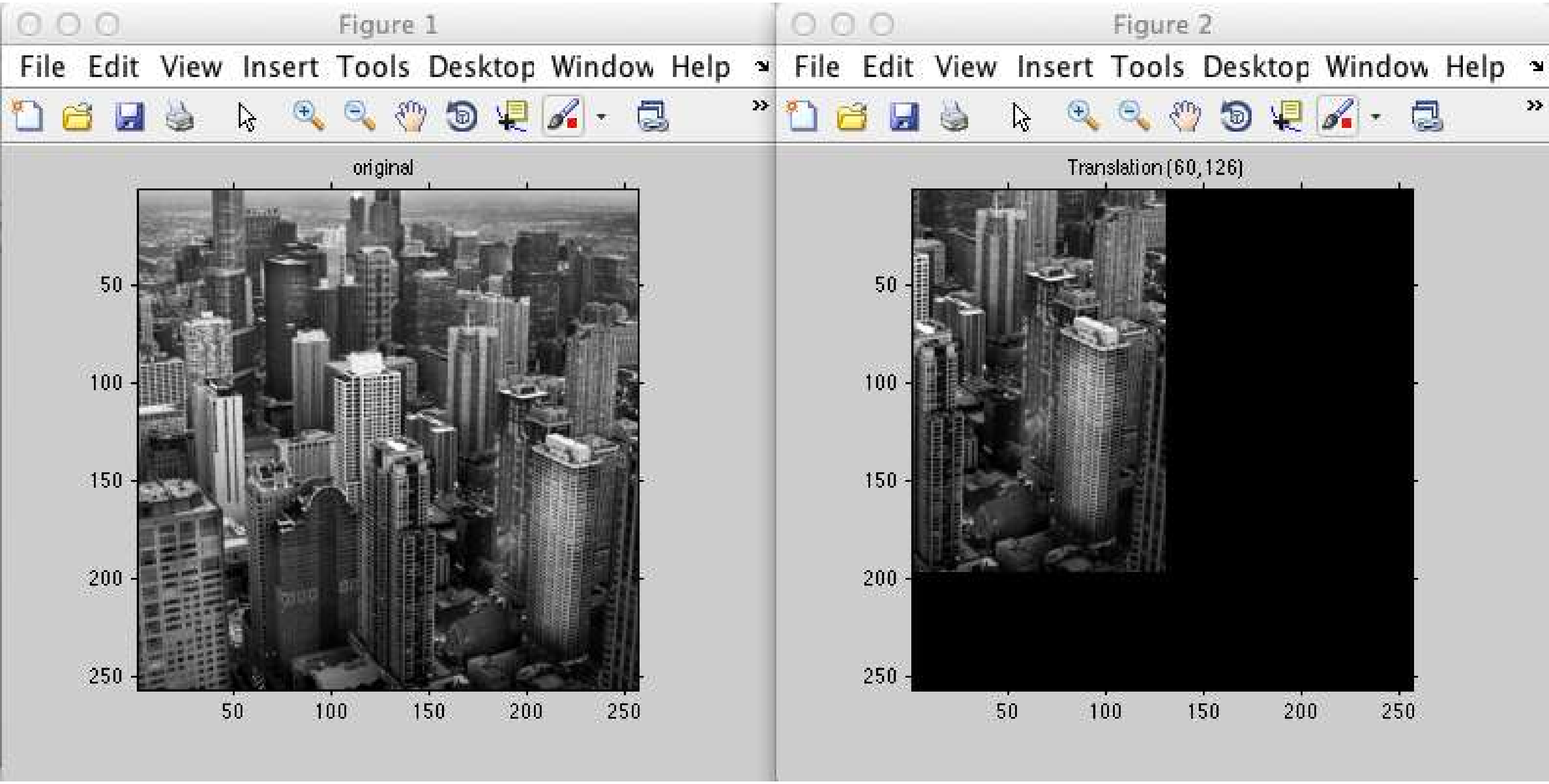
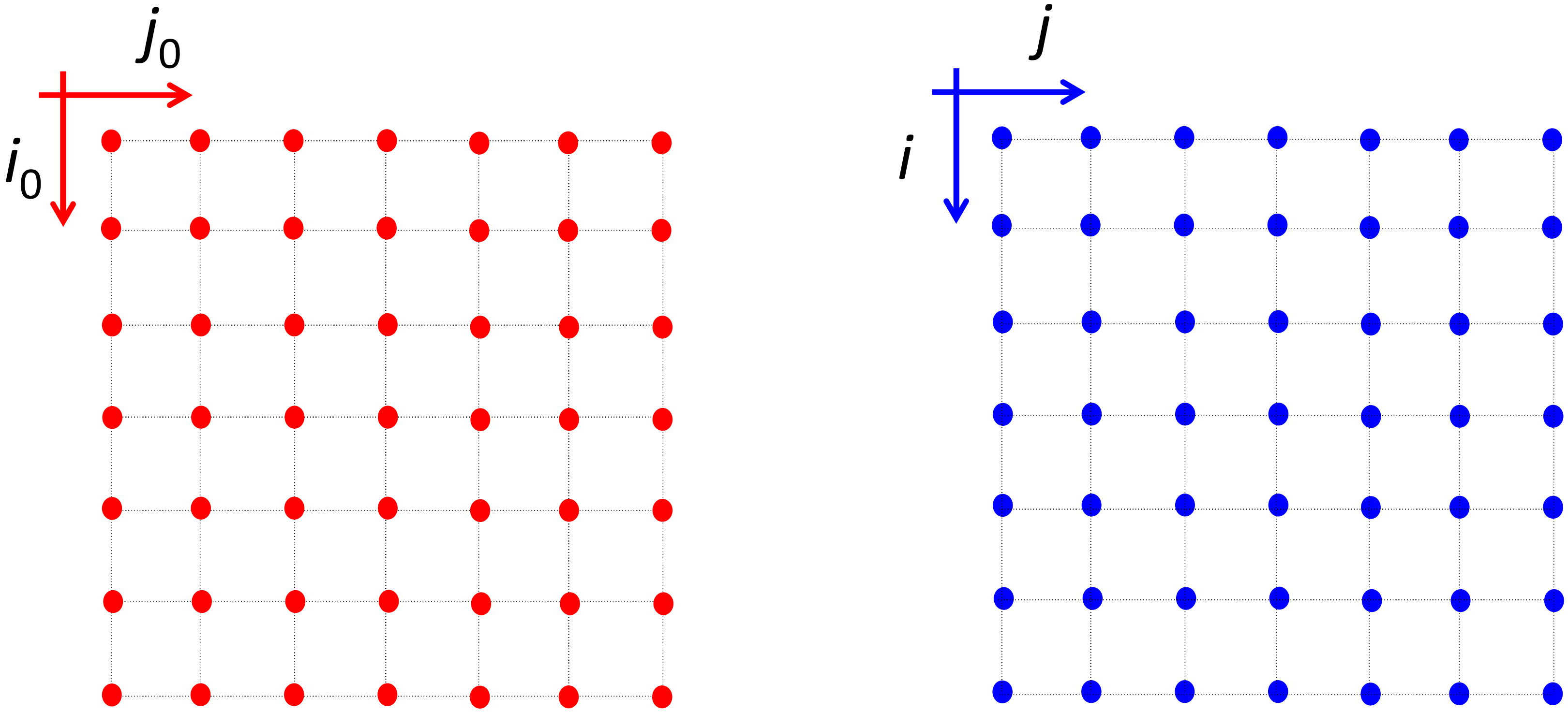


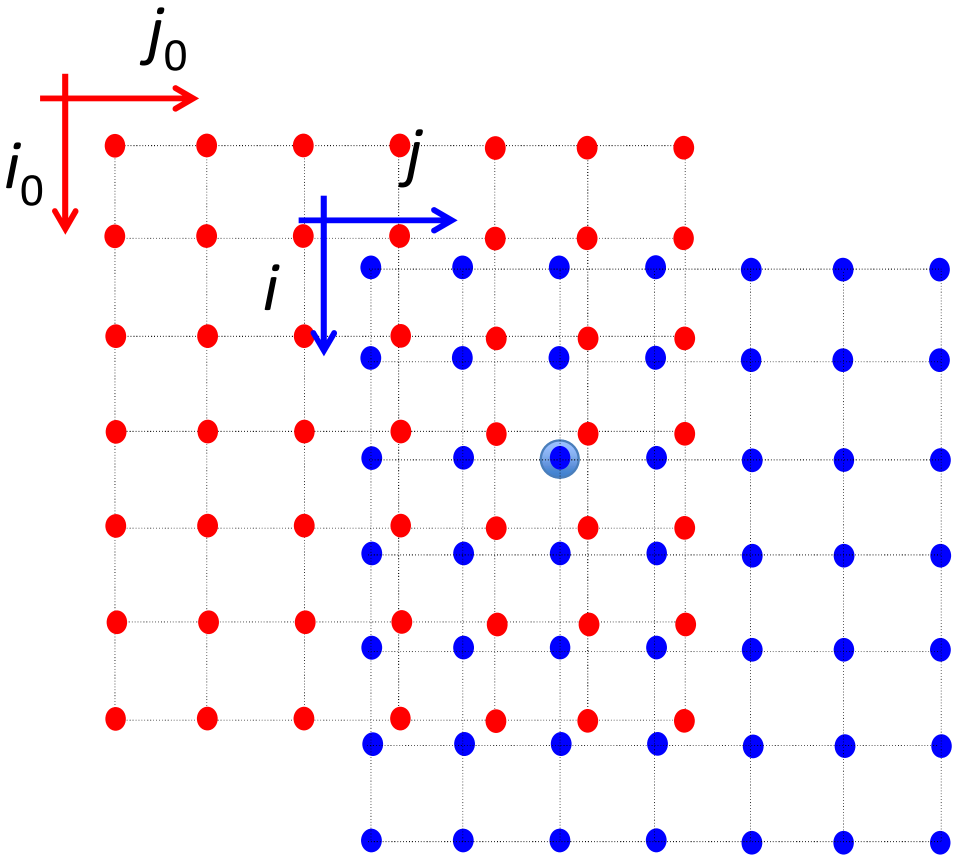
Figure 1
File Edit View Insert Tools Desktop Window Help
original
50
100
150
200
250
50 100 150 200 250
Figure 2
File Edit View Insert Tools Desktop Window Help
Translation (60,126)
50
100
150
200
250
50 100 150 200 250



Coordinate Transformation:
(Example: Translation)

$$i_0 = i + 1.25$$

$$j_0 = j + 2.75$$



Algorithm:

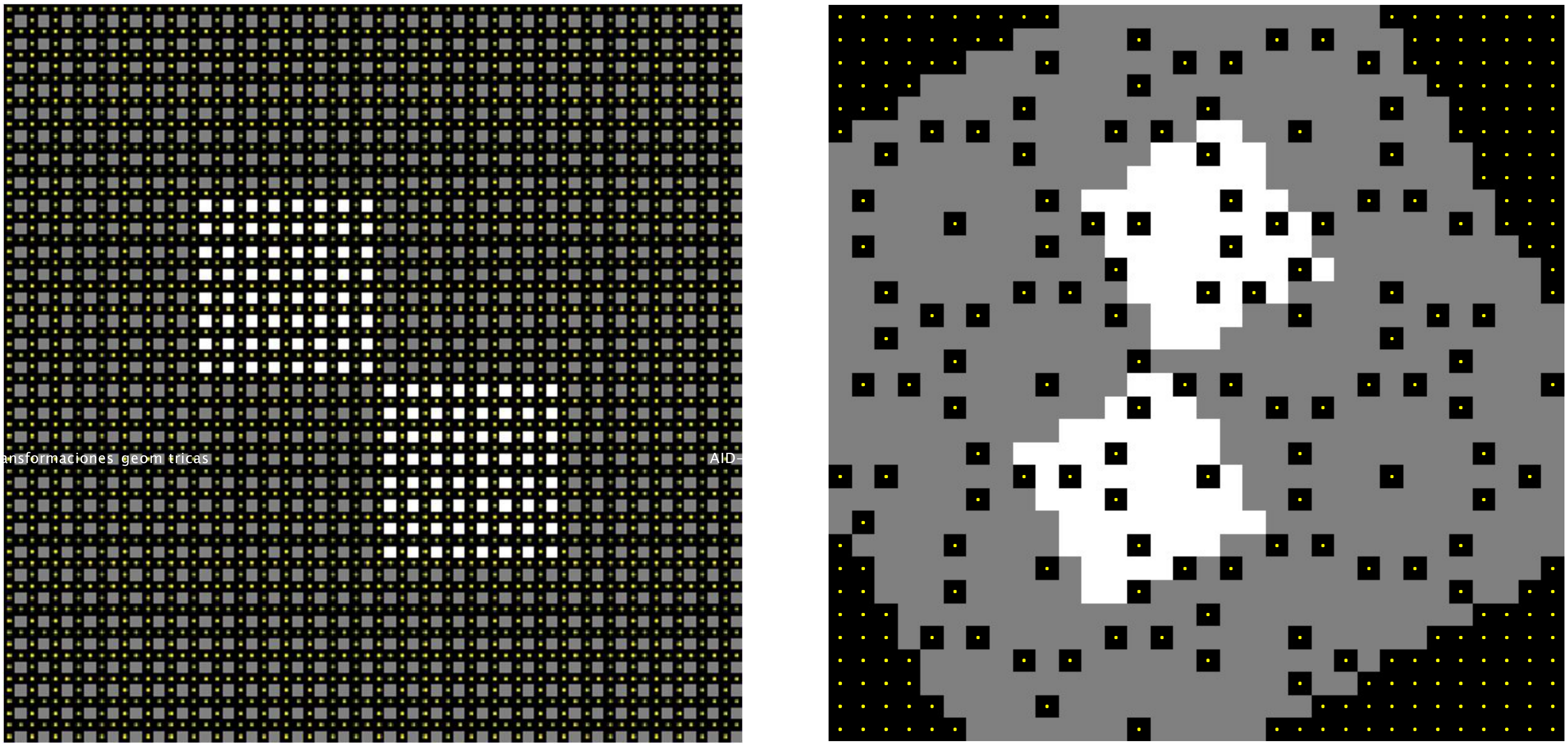
1) For each (i,j) of J compute (i_0,j_0).
2) $J(i,j)$ = interpolation $\{I(i_0,j_0)\}$

Coordinate Transformation:
(Example: Translation)

$$i_0 = i + 1.25$$

$$j_0 = j + 2.75$$

Interpolación



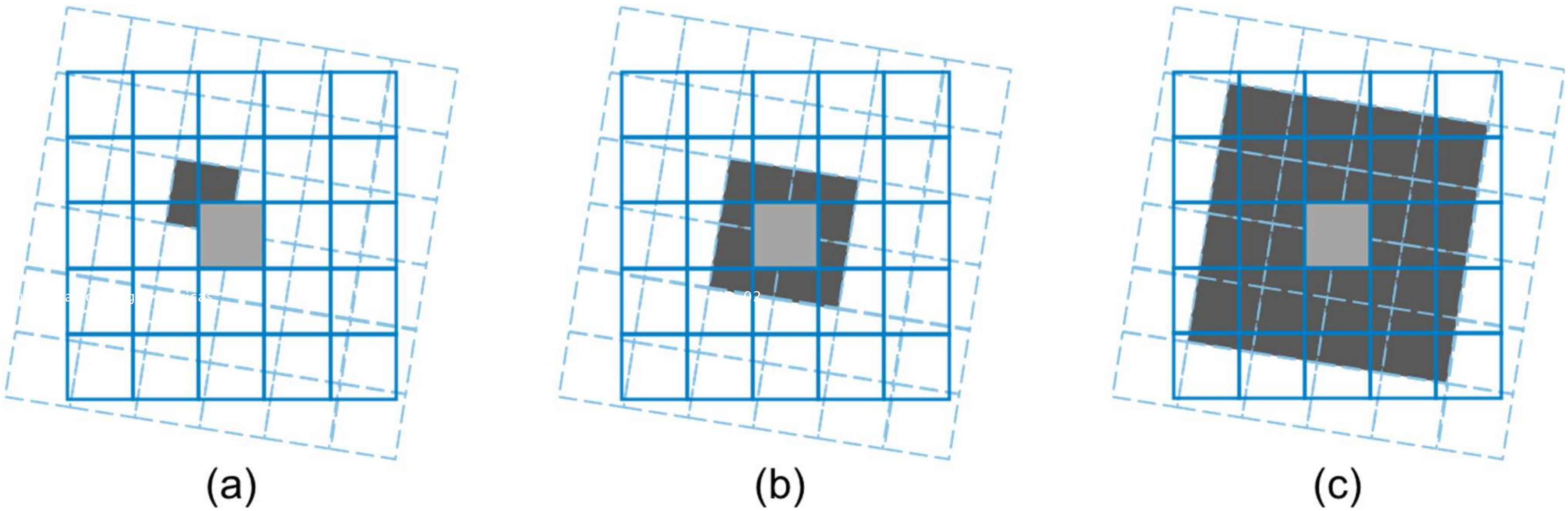
(a) (b)

La transformación geométrica de una imagen puede generar valores no enteros de las coordenadas mapeadas. Al redondear las nuevas coordenadas se generan píxeles que no tienen definidos sus niveles de intensidad: (a) escalamiento con un factor de 2 y (c) rotación a 60°. Los puntos amarillos marcan píxeles indefinidos.

Interpolación

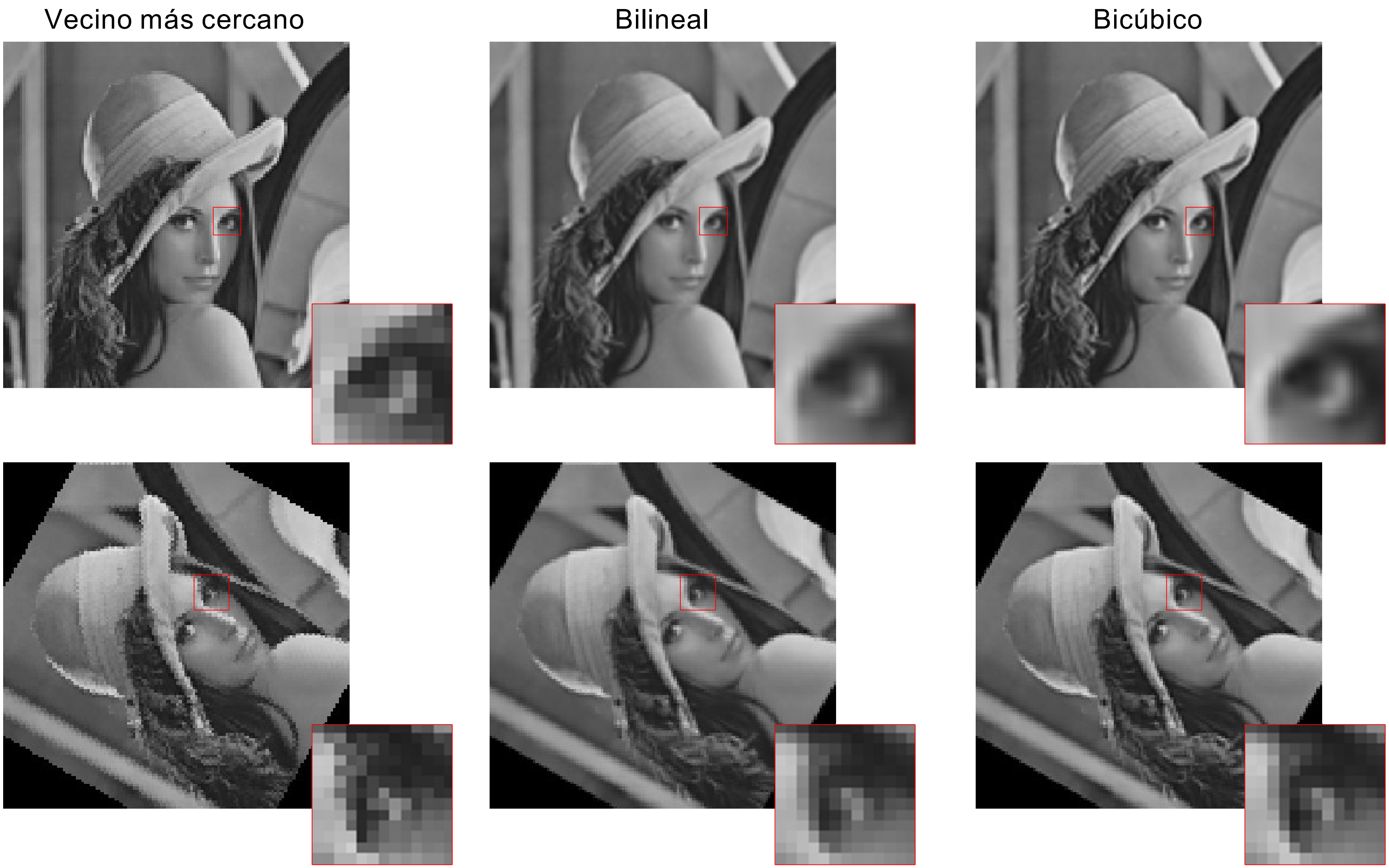
- *Interpolación de los niveles de intensidad* asignar valores de in tensidad a los píxeles indefinidos a partir de píxeles definidos:
    - *Vecino más cercano:* considera el valor del píxel que está más cercano a la nueva localización del píxel.
    - *Bilineal:* promedio de los 4 píxeles más cercanos a la nueva localización del píxel.
    - *Bicúbico:* promedio de los 16 píxeles que rodean a la nueva localización del píxel.
- Una transformación geométrica consta de dos pasos principales: (a) transformación espacial de las coordenadas y (b) interpolación de intensidad de los píxeles transformados.

Interpolación



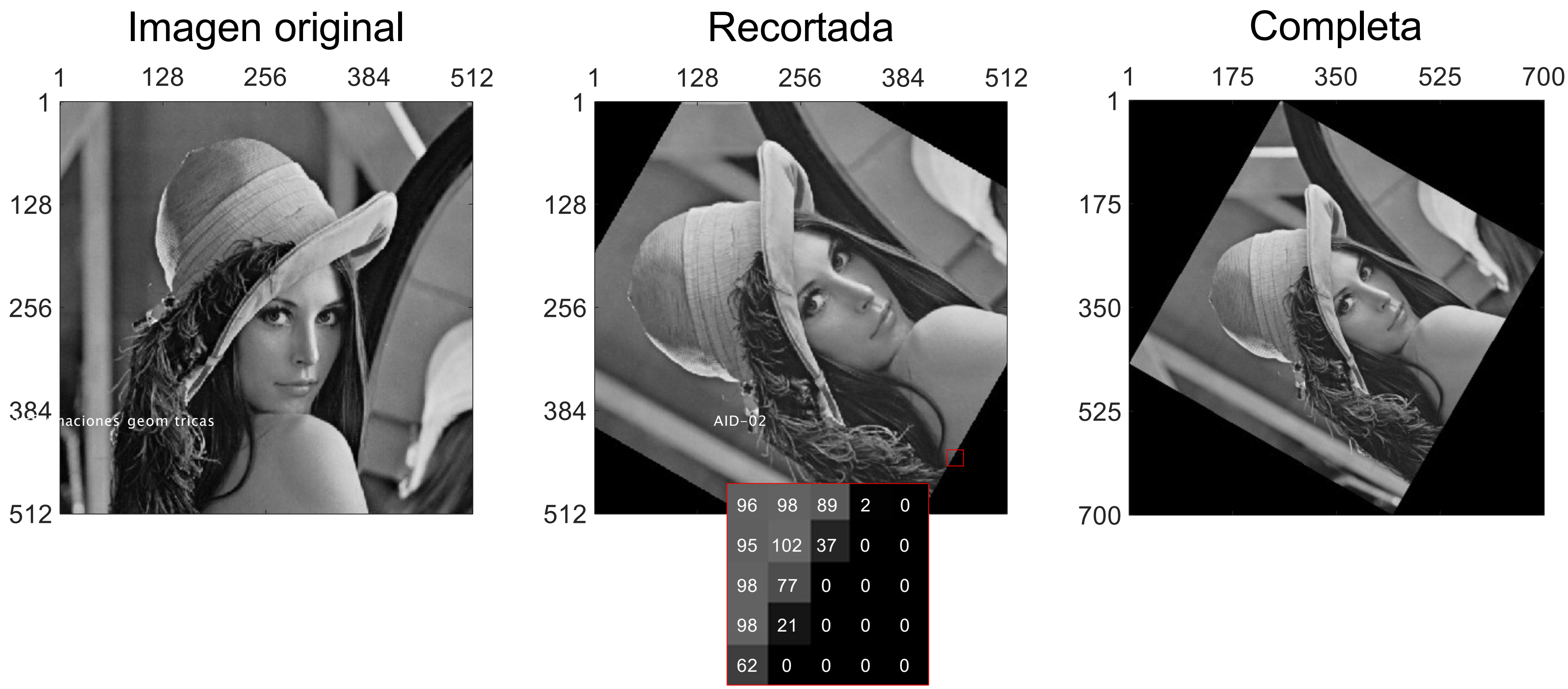
Métodos de interpolación: (a) vecino más cercano, (b) bilineal y (c) bicúbico.

Interpolación

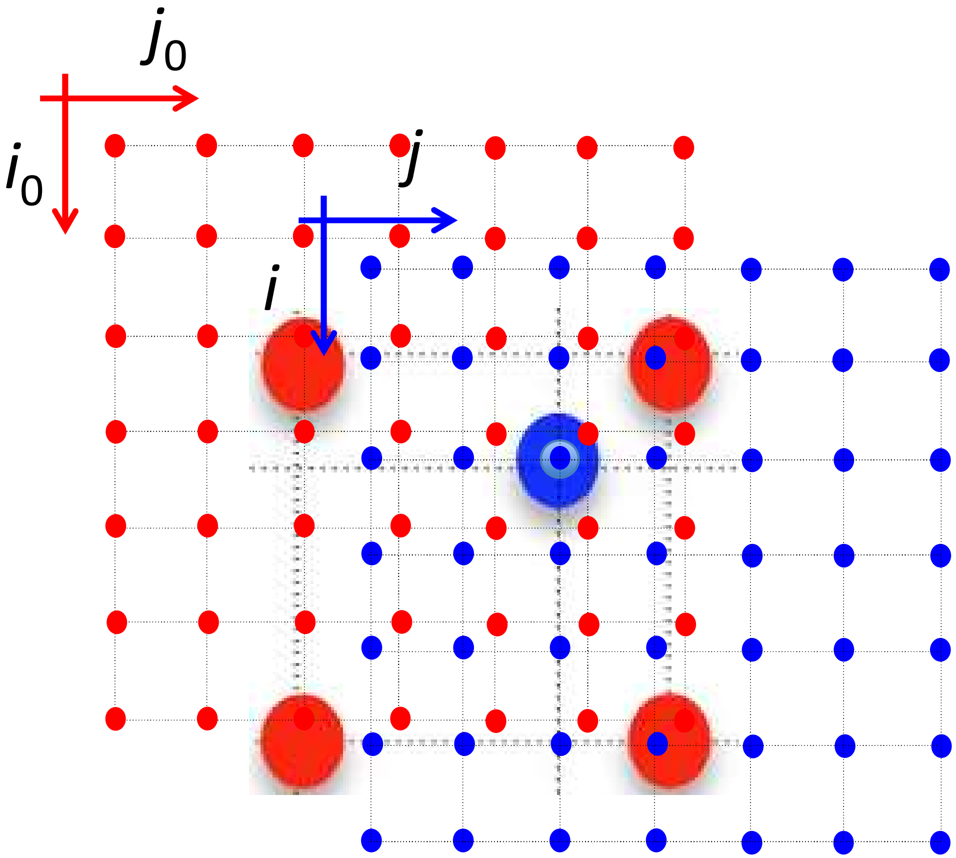


Arriba: imagen escalada 4x. Abajo: imagen rotada 60°.

Interpolación



Consideraciones: El resultado de una transformación geométrica puede generar una imagen de mayor tamaño que la original. La imagen transformada se puede recortar al mismo tamaño que la imagen original, o mantener la imagen completa para evitar pérdida de información. Los píxeles que no pudieron ser interpolados son rellenados con ceros u otro valor deseado.



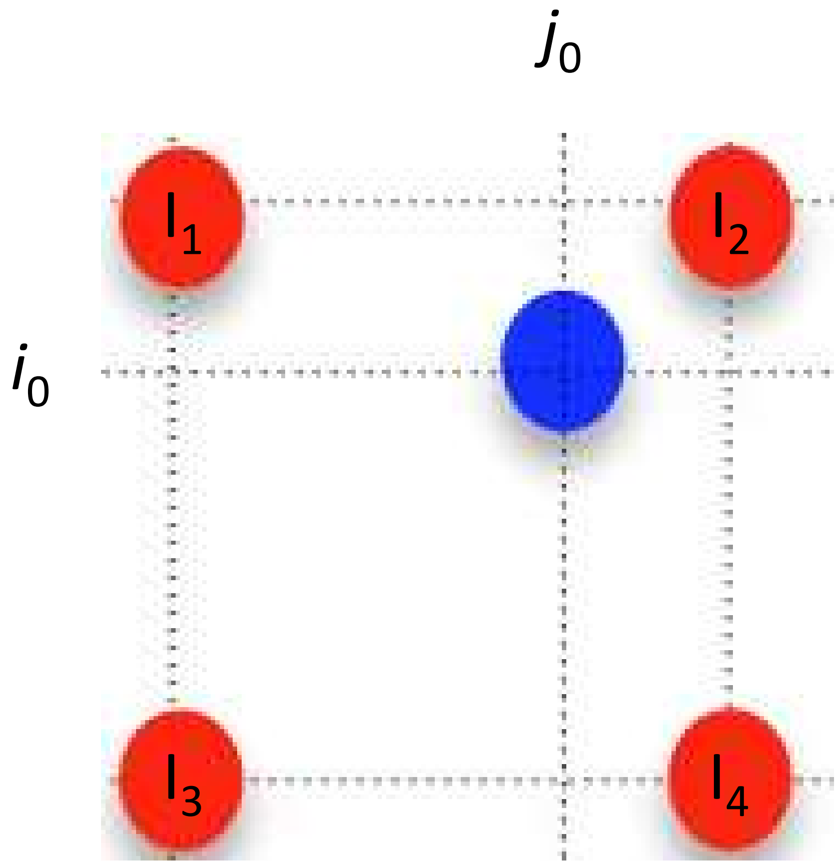
Algorithm:

1) For each (i,j) of J compute (i_0,j_0).
2) $\mathrm{J}(i,j)$ = interpolation $\{\mathrm{I}(i_0,j_0)\}$

Coordinate Transformation:
(Example: Translation)

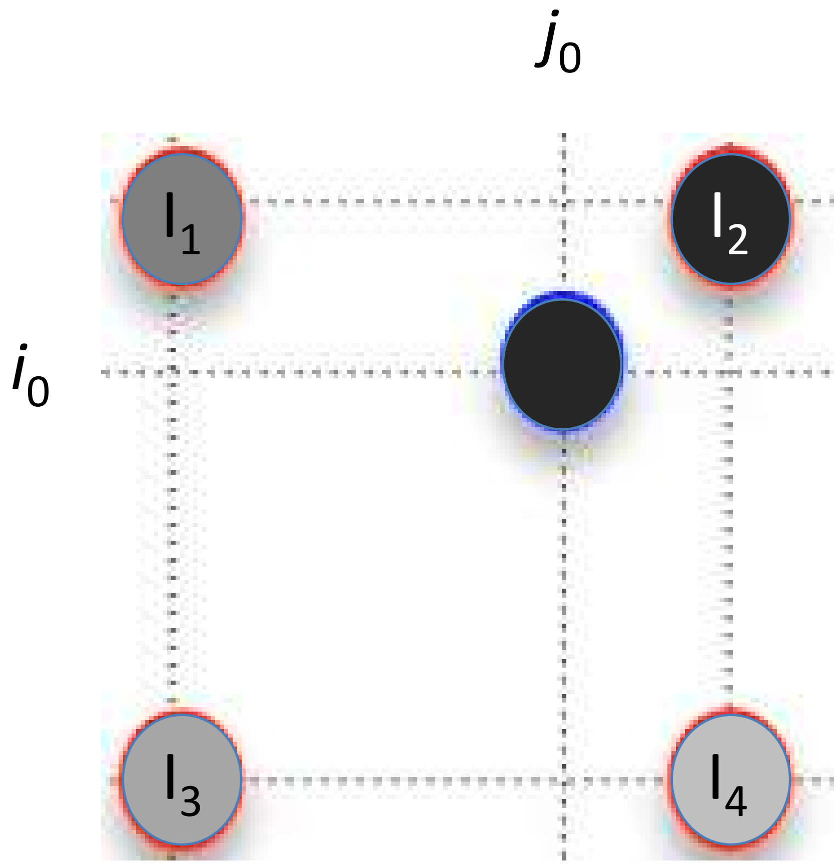
$$i_0 = i + 1.25$$

$$j_0 = j + 2.75$$



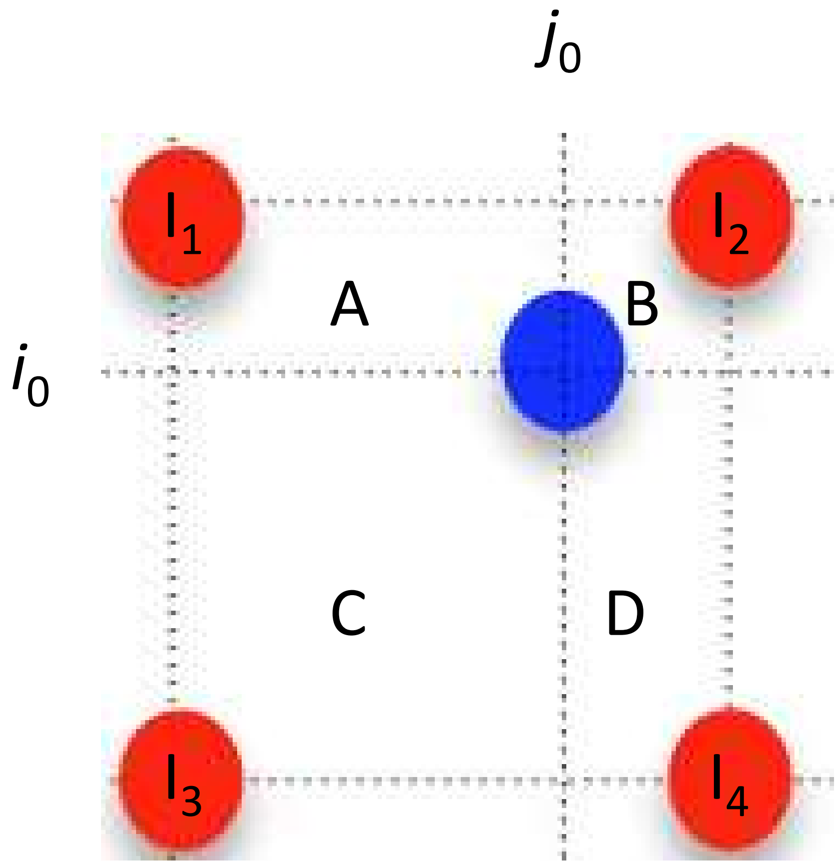
Nearest pixel

$J(i,j)$ = interpolation $\{I(i_0,j_0)\} = I_2$



Nearest pixel

$J(i,j)$ = interpolation $\{I(i_0,j_0)\} = I_2$

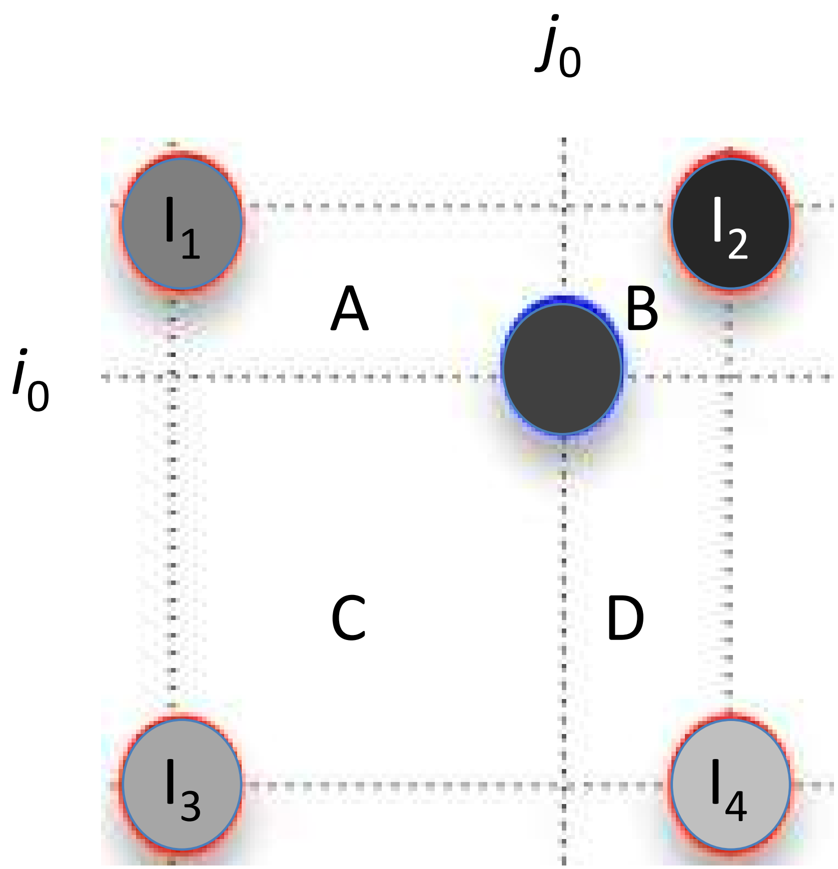


Bilinear interpolation

$J(i,j)$ = interpolation $\{I(i_0,j_0)\}$ =

$AI_4+BI_3+CI_2+DI_1$

$A+B+C+D = 1$

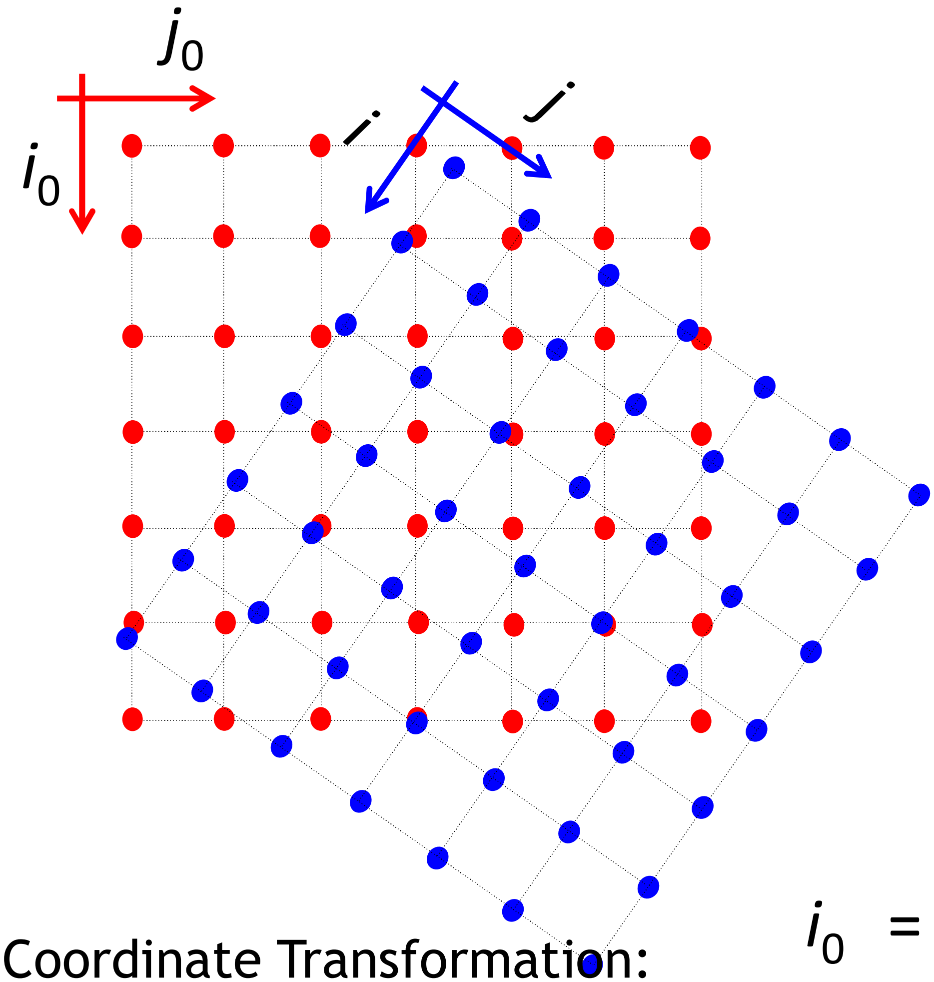


Bilinear interpolation

$J(i,j)$ = interpolation $\{I(i_0,j_0)\}$ =

$AI_4+BI_3+CI_2+DI_1$

$A+B+C+D = 1$



Algorithm:

1) For each (i,j) of J compute (i_0,j_0).
2) J(i,j) = interpolation {I(i_0,j_0)}

Coordinate Transformation:
(Example: Rotation & Translation)

$$i_0 = i \cos\Theta + j \sin\Theta + a$$

$$j_0 = -i \sin\Theta + j \cos\Theta + b$$

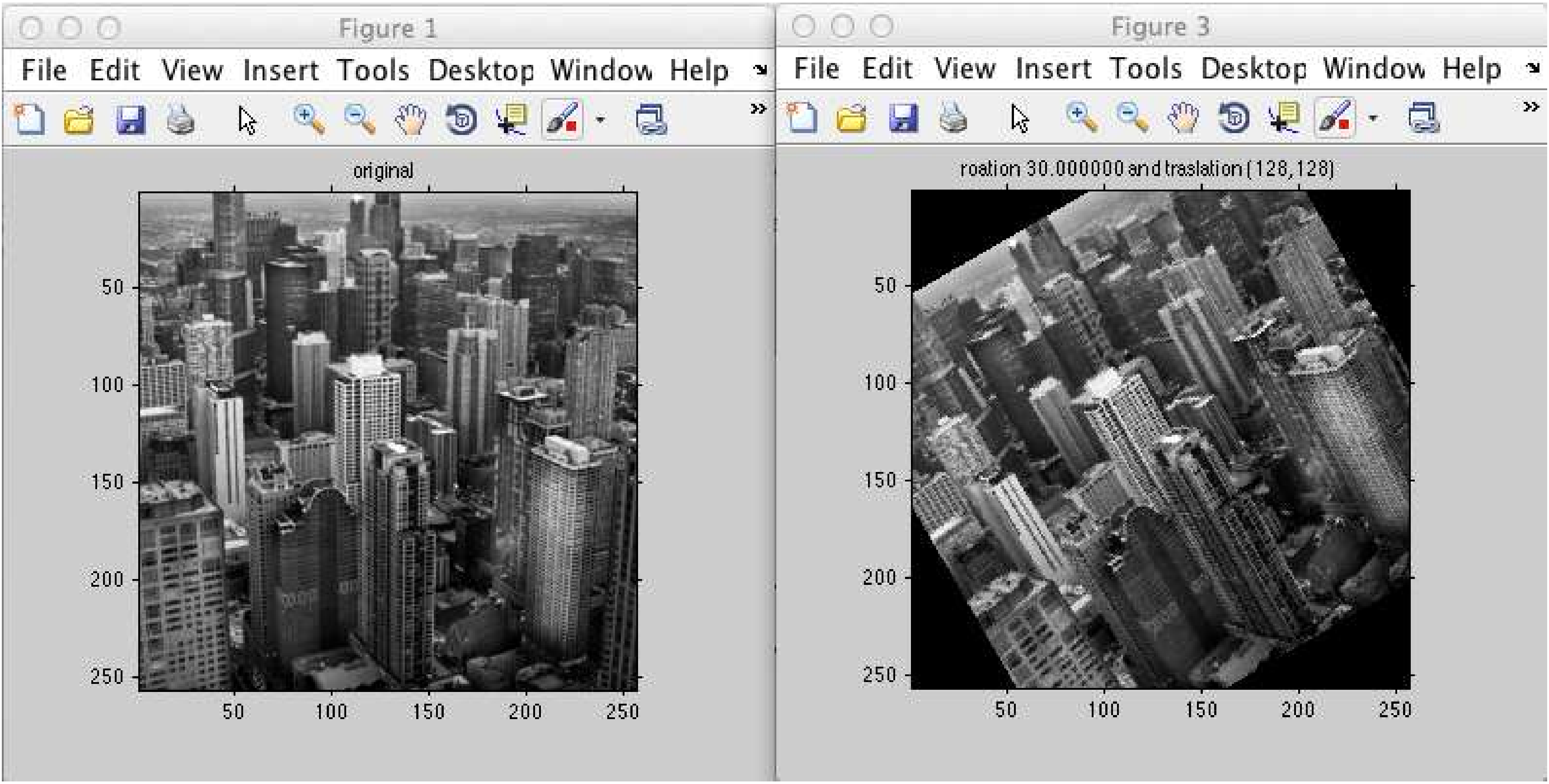
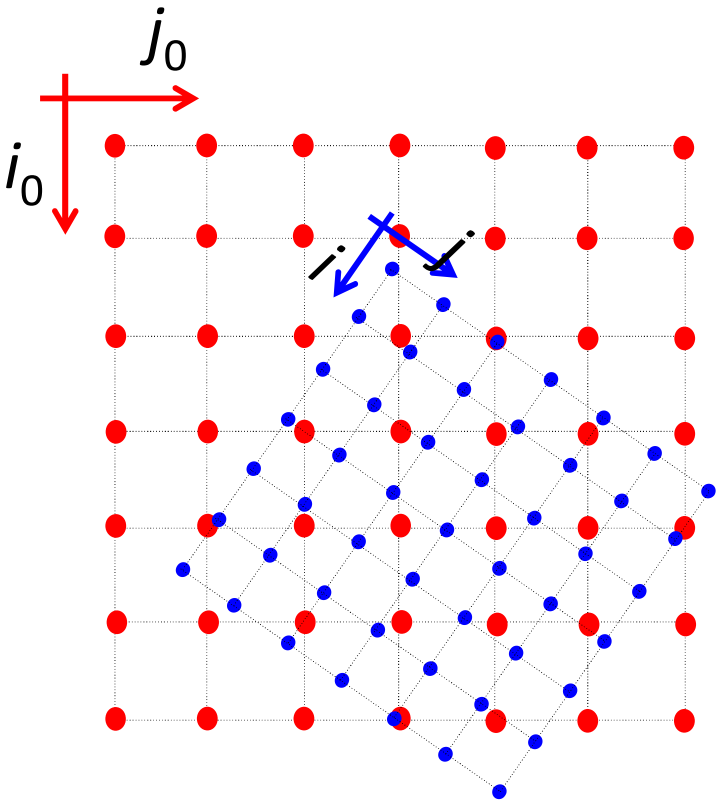


Figure 1
File Edit View Insert Tools Desktop Window Help
original
50
100
150
200
250
50 100 150 200 250
Figure 3
File Edit View Insert Tools Desktop Window Help
roation 30.000000 and traslation (128,128)
50
100
150
200
250
50 100 150 200 250



Algorithm:

1) For each (i,j) of J compute (i_0,j_0).
2) J(i,j) = interpolation $\{$I$(i_0,j_0)\}$

Coordinate Transformation:
(Example: Rotation & Translation & Scale)

$$i_0 = s\,i\cos\Theta + s\,j\sin\Theta + a$$

$$j_0 = -s\,i\sin\Theta + s\,j\cos\Theta + b$$

Registro de imágenes

- Registro de imágenes procedimiento para alinear dos o más imágenes de una misma escena:
  - Imágenes tomadas en tiempos diferentes
  - Imágenes adquiridas con distintos sistemas
- Transformar la imagen de entrada, $f_i(x, y)$, para que sea similar a la imagen de referencia, $f_r(x, y)$:

$$f_r(x, y) \sim f_i(x\hat{}, y\hat{}), \quad \text{con} \quad (x\hat{}, y\hat{}) = \boldsymbol{T}\{(x, y); \boldsymbol{\theta}\}$$

donde $\boldsymbol{\theta} = [\vartheta_1, \ldots, \vartheta_k]^T$ son los parámetros de transformación (e.g., ángulo de rotación, factor de escala, desplazamiento, etc.).

- Función de error:

$$J(\theta) = \sum_{x,y} \left[f_i\left(x\hat{}, y\hat{}\right) - f_r(x, y) \right]^2$$

Registro de imágenes

- Problema de optimización:

$$\underset{\boldsymbol{\theta}}{\text{minimizar}} \sum_{x,y} [f_i(x^\wedge, y^\wedge) - f_r(x,y)]^2$$

- Solución mediante descenso de gradiente:

$$\boldsymbol{\theta}^{t+1} = \boldsymbol{\theta}^t - \eta \nabla_\theta J(\boldsymbol{\theta})$$

donde $\eta > 0$ es la tasa de aprendizaje y el gradiente es:

$$\nabla_\theta J(\boldsymbol{\theta}) = \left[\frac{\partial J(\boldsymbol{\theta})}{\partial \theta_1}, \dots, \frac{\partial J(\boldsymbol{\theta})}{\partial \theta_k}\right]^T$$

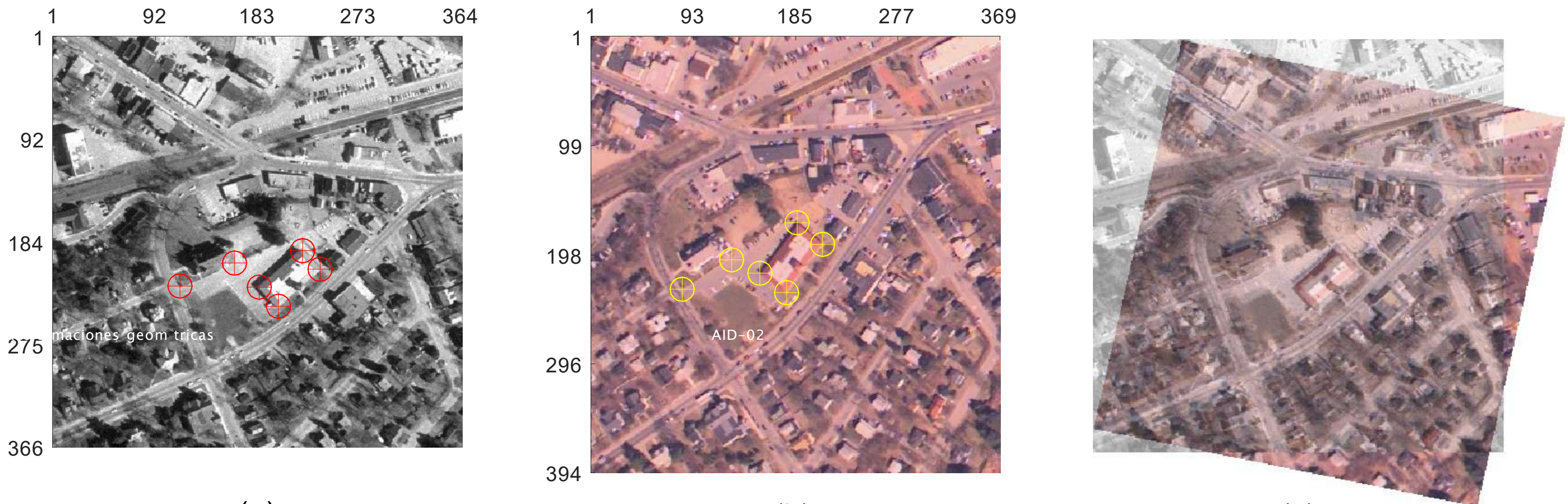
Registro de imágenes

- Clasificación de métodos:
  - *Extrínsecos:* puntos de control o marcadores
  - *Intrínsecos:* puntos de interés o valores de los píxeles
- *Puntos de control* indican la localización de puntos comunes en las imágenes de entrada y referencia.
- Aplicar *transformaciones geométricas* para empatar los puntos de control de la imagen de entrada, (v, w), con los puntos en la imágen de referencia, (x, y):

$$x = c_1 v + c_2 w + c_3 vw + c_4, \quad y = c_5 v + c_6 w + c_7 vw + c_8$$

donde c_1 a c_8 son coeficientes desconocidos que deben estimarse.

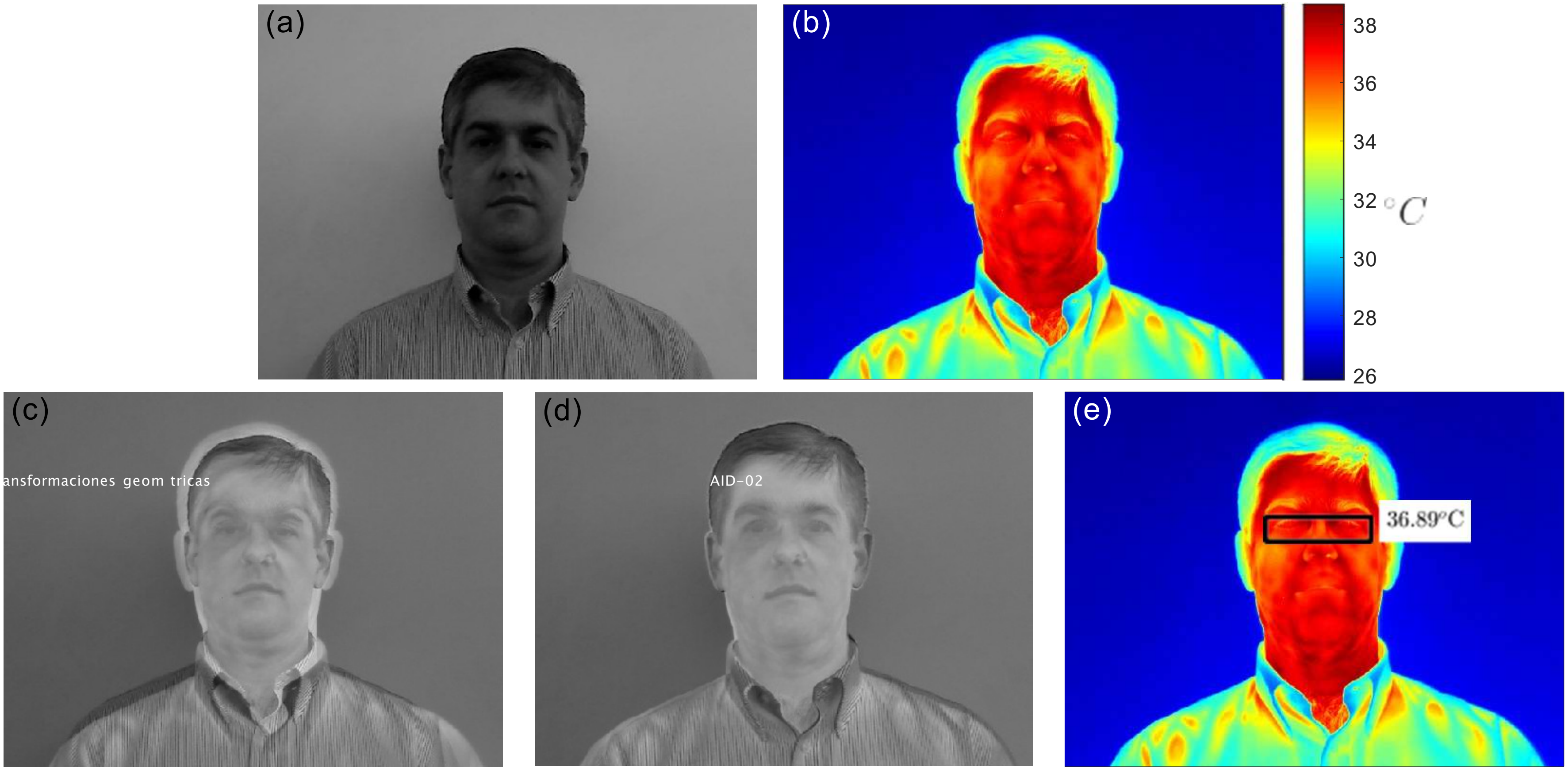
Registro de imágenes



(a) (b) (c)

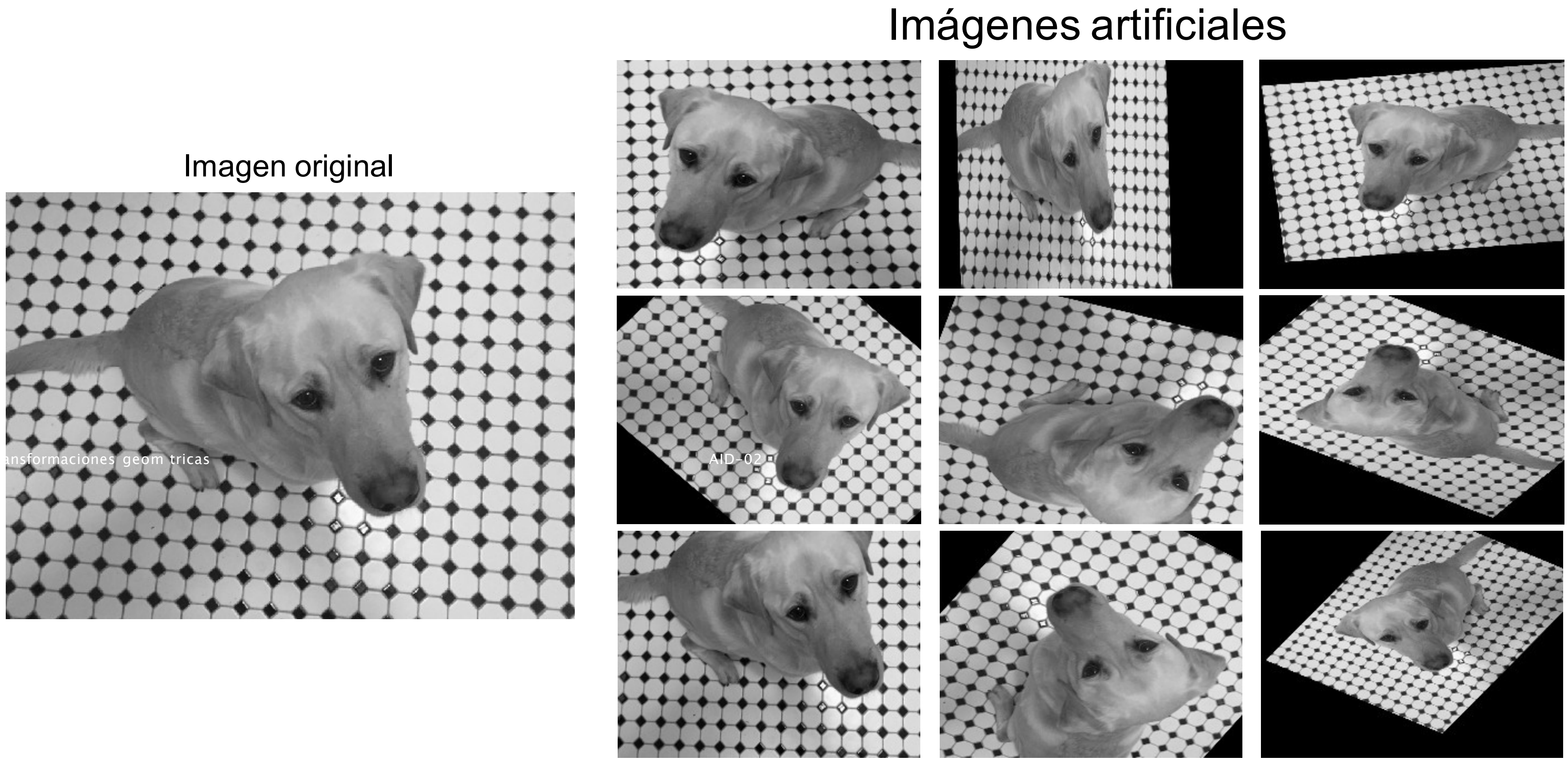
Registro de imágenes áreas (a) imagen de referencia, (b) imagen sin registrar, y (c) imagen registrada. Mediante el método de mínimos cuadrados, se empatan los puntos de control de la imagen (b) (en amarillo) con los puntos de referencia de la imagen (a) (en rojo).

Registro de imágenes



Registro de imágenes obtenidas de sensores diferentes: (a) imagen en escala de grises, (b) imagen térmica, (c) imágenes (a) y (b) desalineadas, (d) imagen en escala de grises registrada alineada con la imagen térmica, y (e) medición de temperatura en la zona de los ojos. La detección de los ojos se realiza sobre la imagen registrada.

Imágenes artificiales

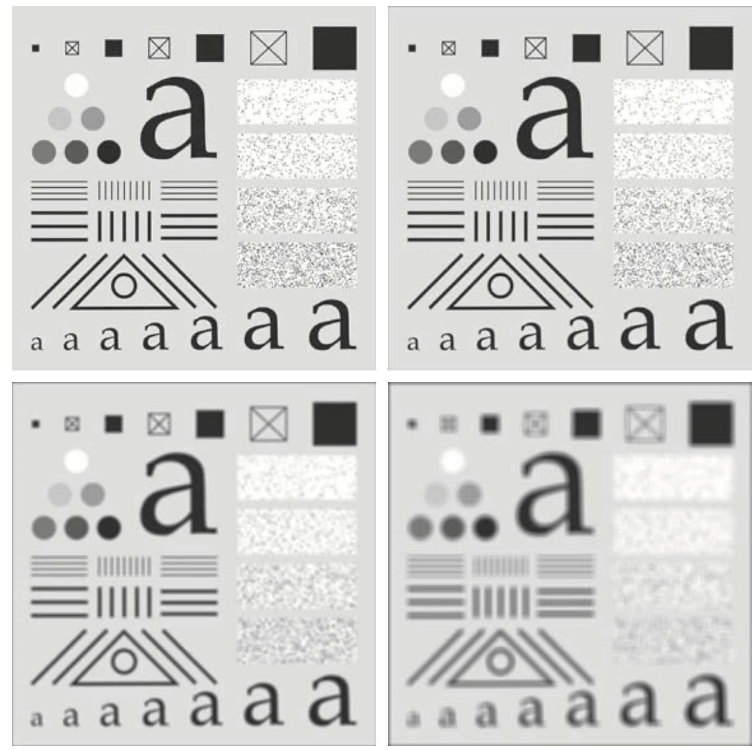


En aplicaciones de clasificación de imágenes con deep learning, conjuntos de datos pequeños se suelen aumentar de manera artificial aplicando distintas transformaciones geométricas a cada imagen para agregar diferentes posibles vistas para mejorar la generalización del modelo de clasificación.

Introducción al filtrado espacial

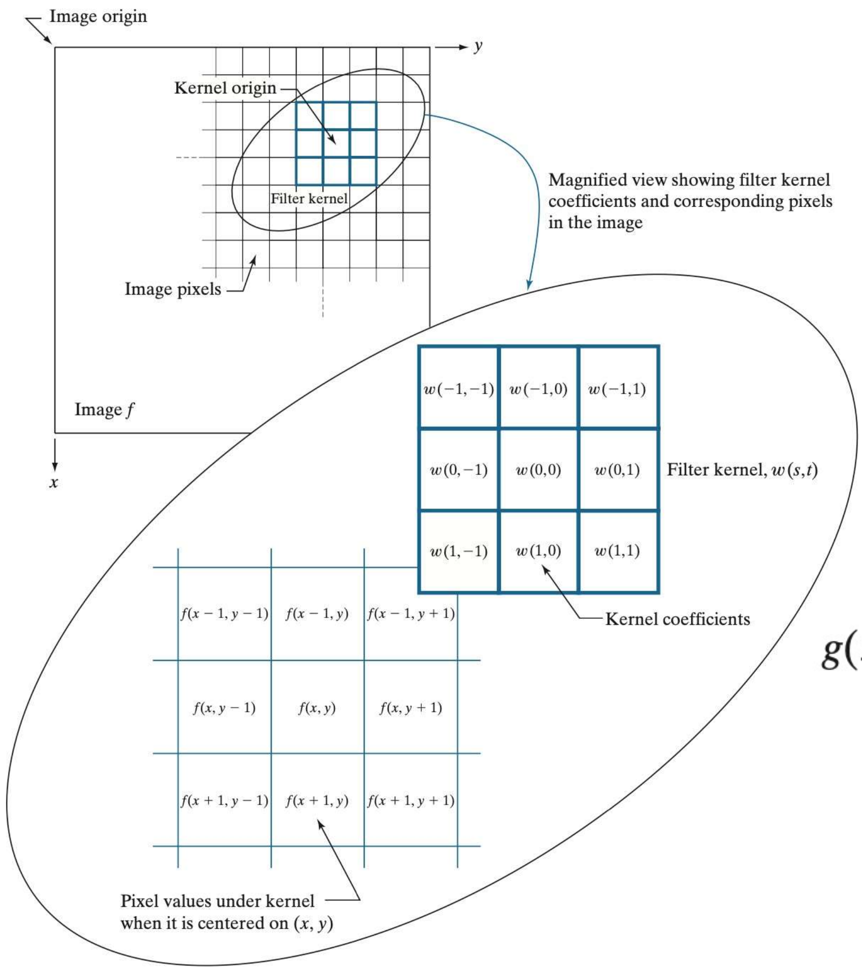
El término *filtro* viene del procesamiento de señales en frecuencia, donde “filtrar” una señal se refiere a dejar pasar, modificar o rechazar determinado componente de frecuencia.

En el filtrado espacial se modifica una imagen al reemplazar el valor de intensidad de cada píxel en función de los valores del píxel y sus vecinos.



Ejemplo de un filtro pasabajas usando regiones de diferentes tamaños.

Introducción al filtrado espacial



Un filtro espacial lineal realiza una operación de suma de productos entre una imagen f y un *filtro,* o *kernel,* .

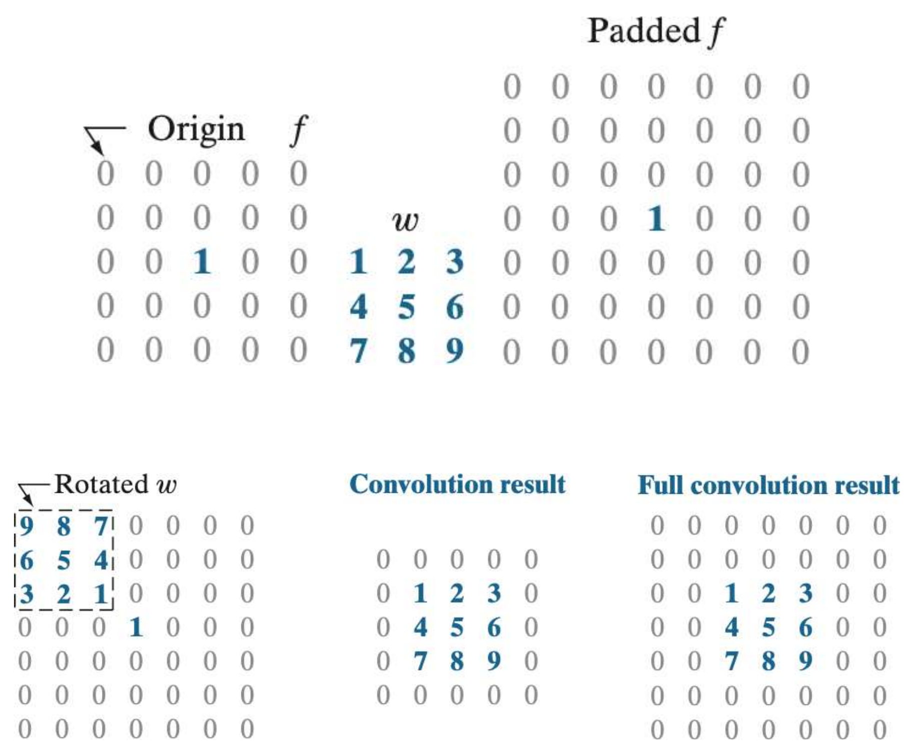
Usualmente, los *filtros* o *kernels* son matrices cuadradas con un número impar de filas y columnas, y mucho menor a las dimensiones de la imagen.

$$g(x,y) = w(-1,-1)f(x-1,y-1) + w(-1,0)f(x-1,y) + \ldots \\ + w(0,0)f(x,y) + \ldots + w(1,1)f(x+1,y+1)$$

$$g(x,y) = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w(s,t)f(x+s,y+t)$$

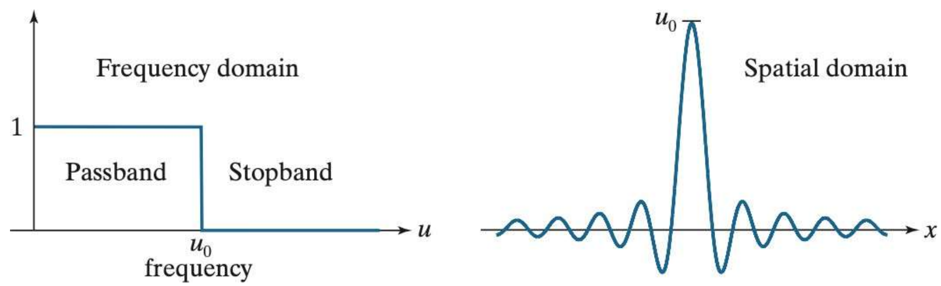
Introducción al filtrado espacial

- La operación de *convolución* se refiere a desplazar el *kernel* por todos los píxeles de la imagen calculando la suma de productos. Dada la naturaleza de la operación, la imagen resultante es de una menor dimensión que la original, para evitar esto usualmente se aplica *padding* alrededor de la imagen antes de realizar la operación.



Introducción al filtrado espacial

Vale la pena hacer la relación entre el dominio de la frecuencia y el dominio espacial, la multiplicación en frecuencia se convierte una *convolución* en el espacio, por lo que podemos obtener el mismo resultado en cualquiera de los dos dominios.



Función de transferencia de un filtro pasabajas
y su representación en el dominio espacial
Transformada y transformada inversa de Fourier

Introducción al filtrado espacial

El filtro de promedio o suavizado (*smoothing*) se usa para reducir transiciones bruscas de intensidad en la imagen. Algunas de las aplicaciones son:

- Reducción de ruido aleatorio
- Reducción de *aliasing*
- Eliminación de detalles "irrelevantes"
- Suavizado de falsos contornos (número de bits insuficiente)

$$\frac{1}{9} \times \begin{array}{|c|c|c|} \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline 1 & 1 & 1 \\ \hline \end{array}$$

Box kernel para suavizado de imágenes

Referencias

- Digital Image Processing 4th ed. – R. Gonzalez, R. Woods