

Tratamiento de Señales

Version 2024-I

Transformaciones geométricas e Interpolación

[Capítulo 3]

Dr. José Ramón Iglesias

DSP-ASIC BUILDER GROUP

Director Semillero TRIAC

Ingeniería Electronica

Universidad Popular del Cesar

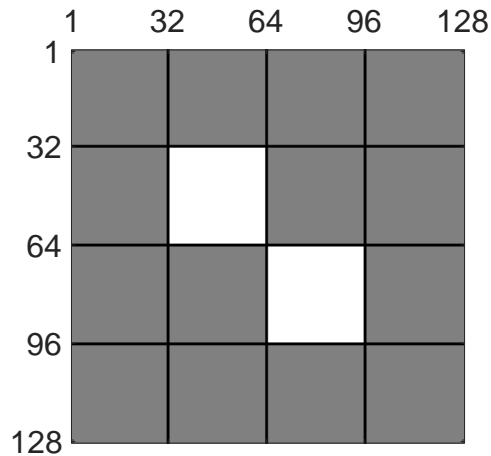
Introducción

- *Transformación geométrica*. Función que modifica 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 colinealidad.
- La transformación de coordenadas se expresa como:

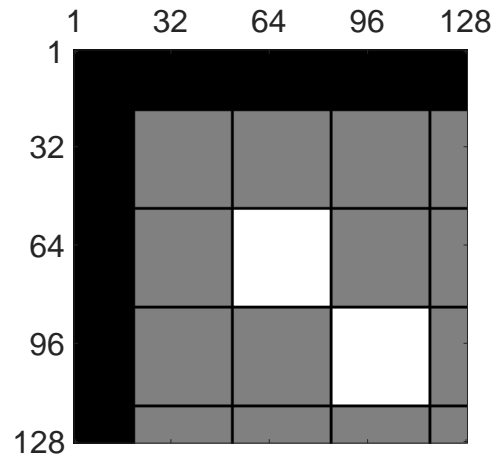
$$(\hat{x}, \hat{y}) = \mathbf{T}\{(x, y)\} \quad (1)$$

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

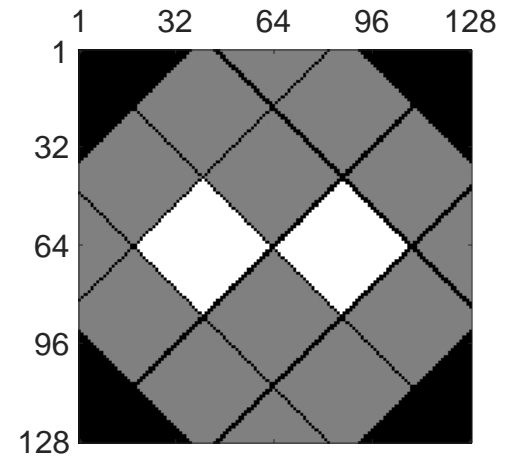
Transformaciones geométricas



(a)



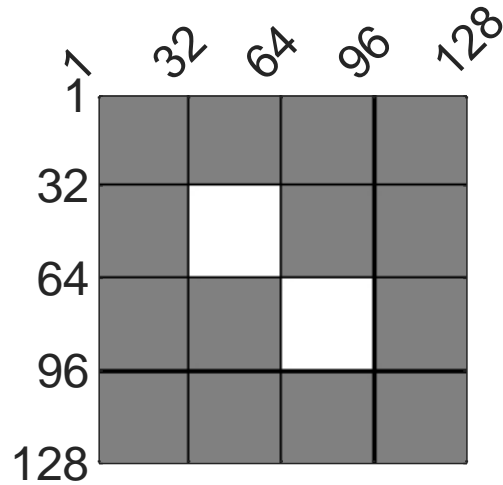
(b)



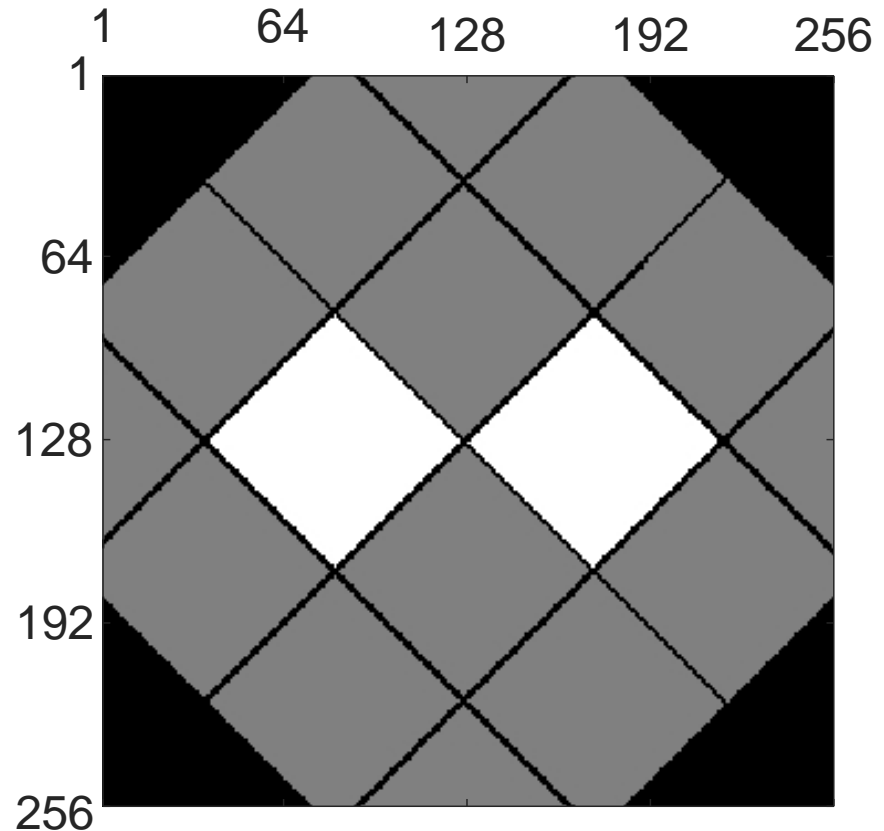
(c)

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.

Transformaciones geométricas



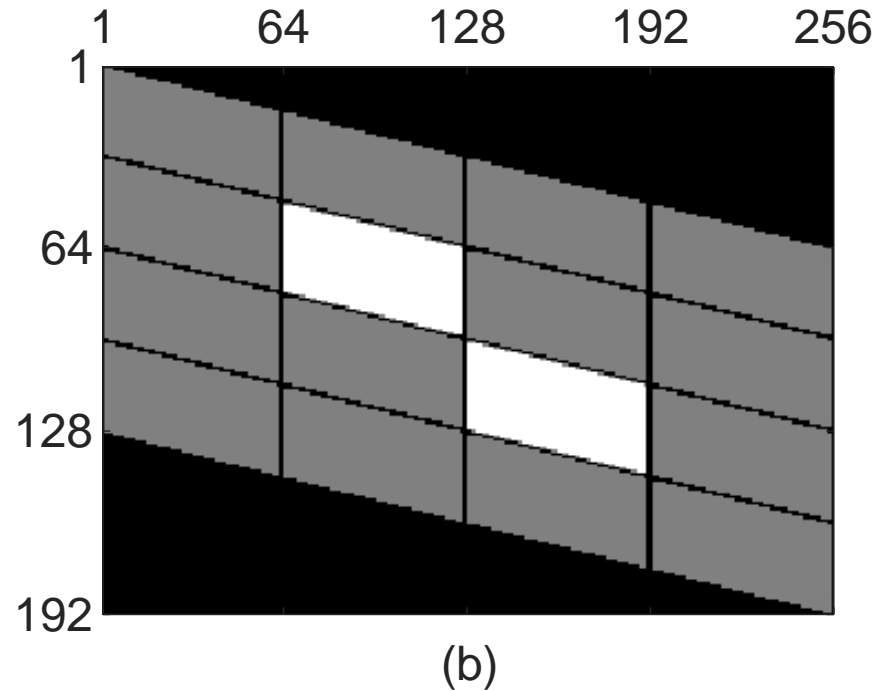
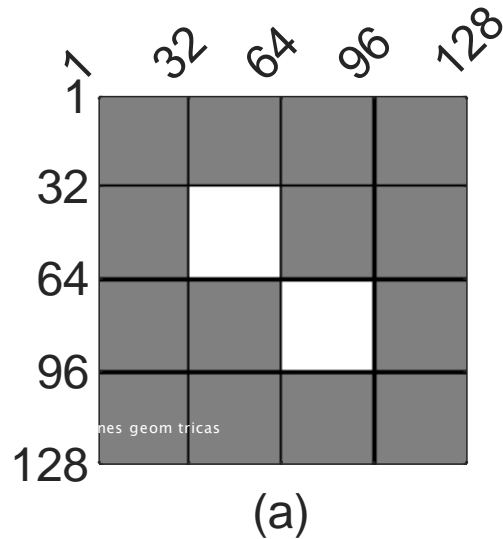
(a)



(b)

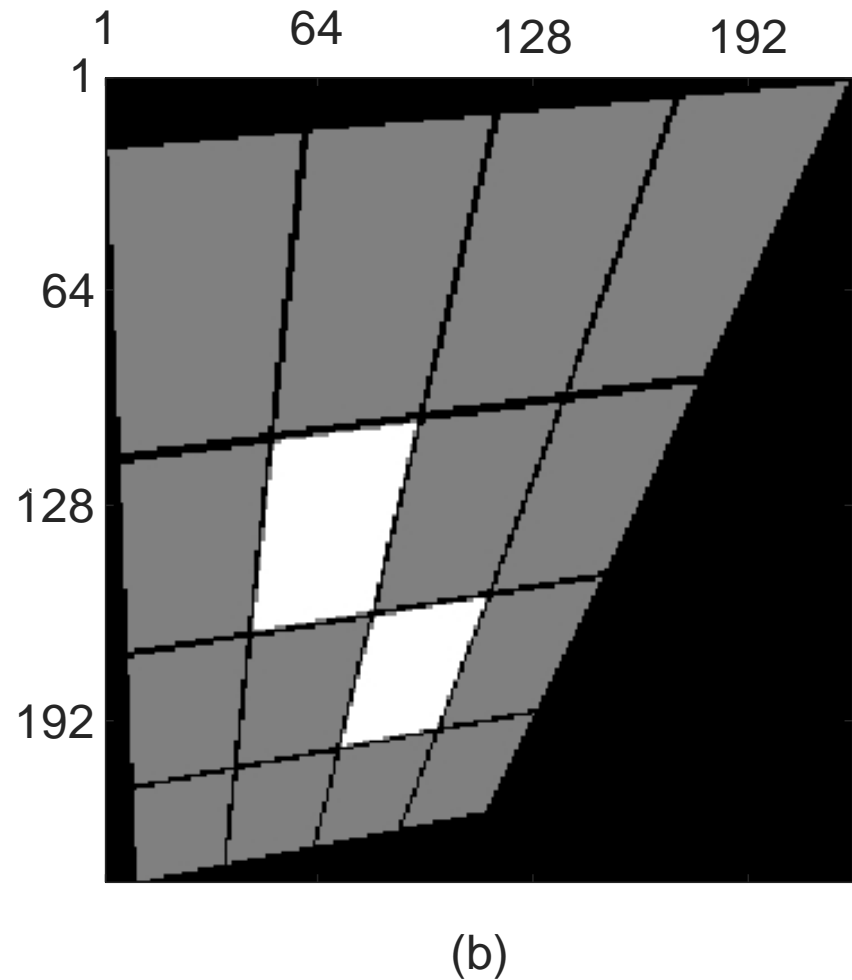
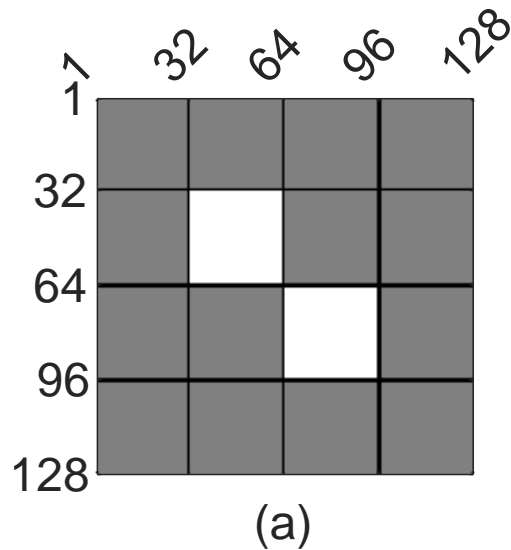
Las transformaciones semejantes, a partir de un punto fijo, multiplican todas las distancias por un mismo factor, de manera que se preservan los ángulos de los objetos en la imagen. (a) Imagen original y (b) imagen escalada uniformemente y rotada.

Transformaciones geométricas

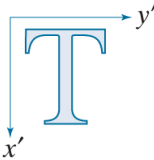
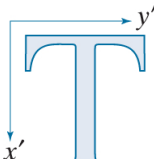
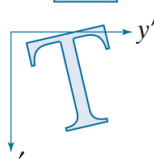
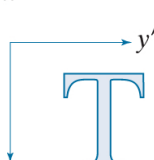
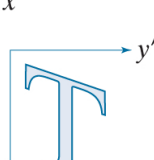
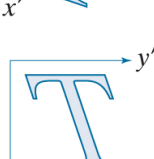


Las transformaciones afines preservan las relaciones de las distancias entre dos puntos cualquiera de la imagen, manteniendo así el paralelismo entre segmentos de rectas. (a) Imagen original y (b) imagen escalada con diferente factor en cada eje coordenado.

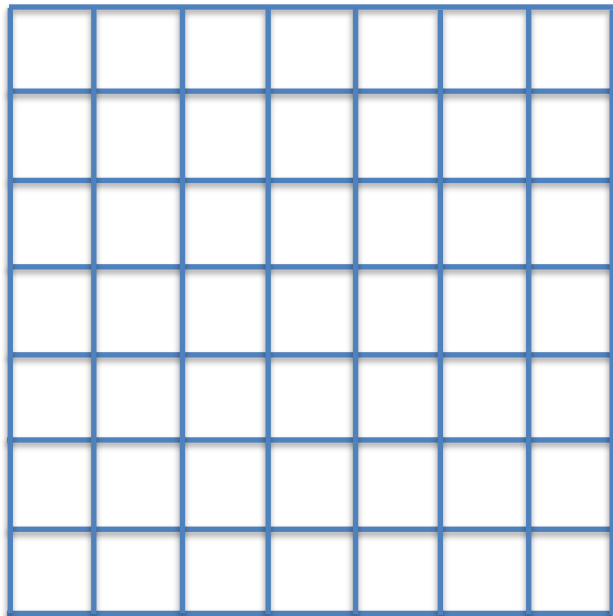
Transformaciones geométricas



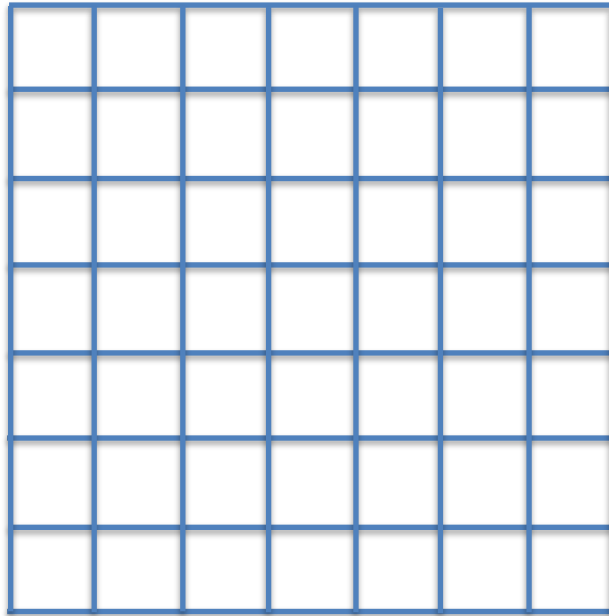
Las transformaciones proyectiva preservan la colinealidad. (a) Imagen original y (b) imagen con proyección a partir de un punto de fuga.

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

I

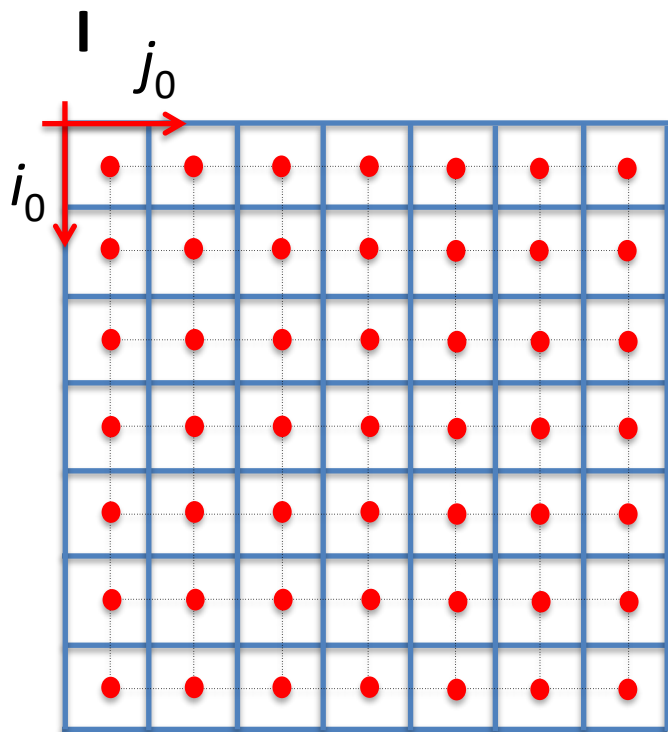


J

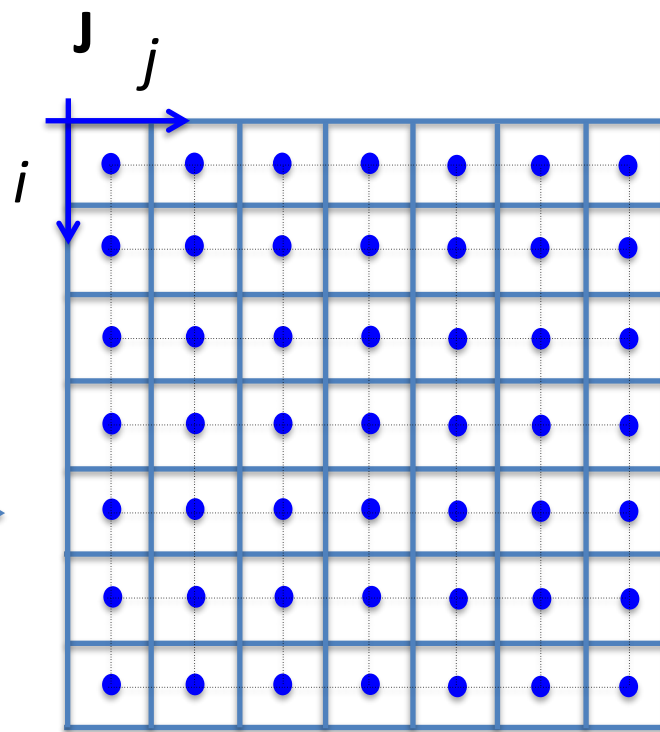


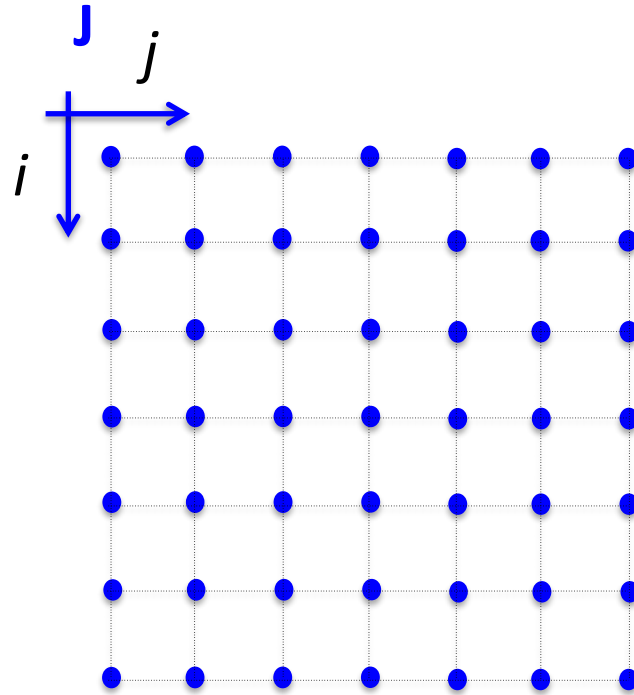
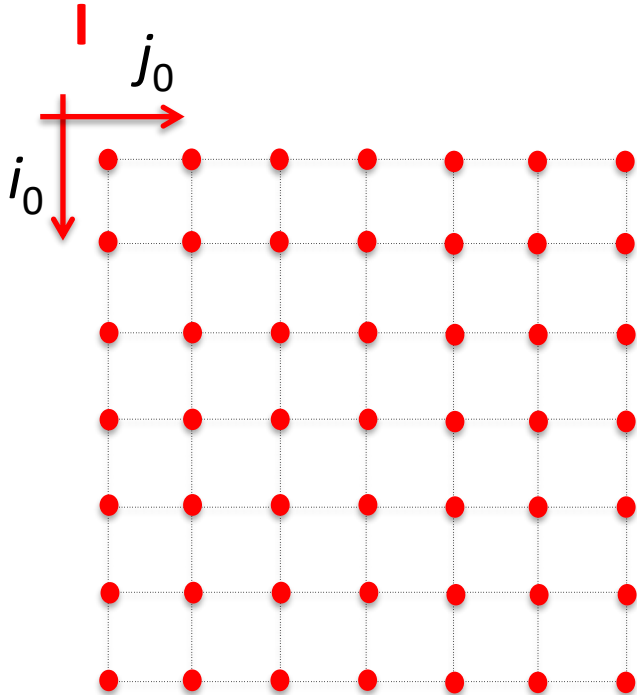
T





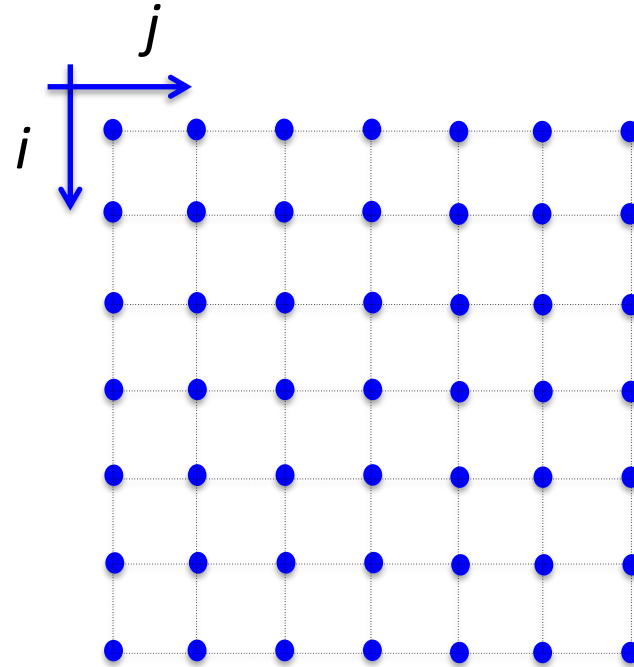
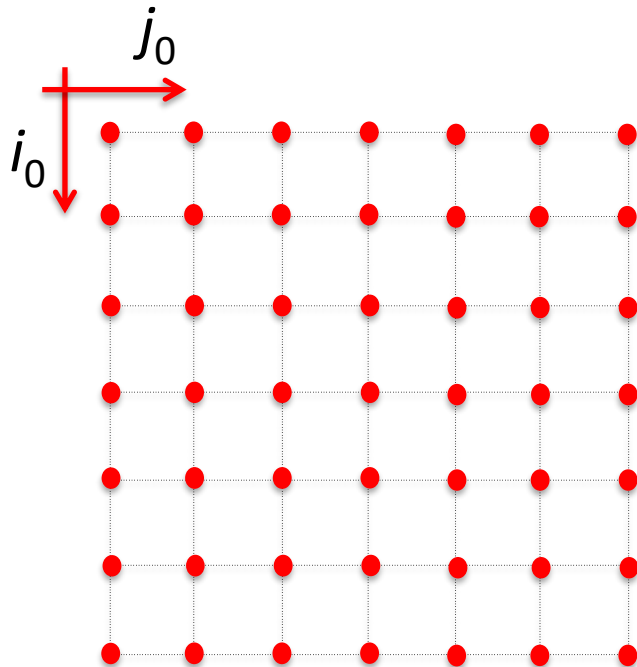
T





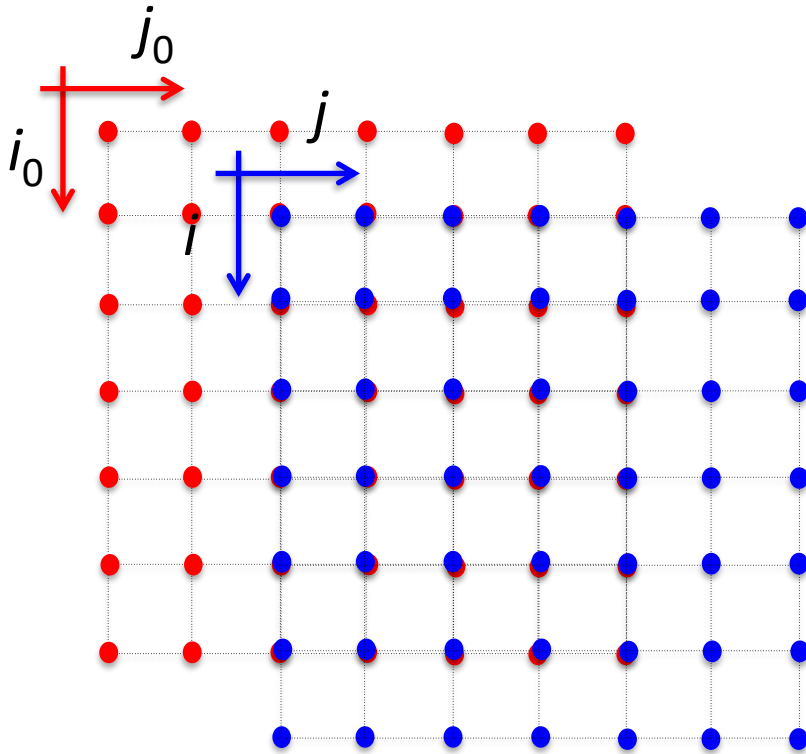
Coordinate Transformation: $i_0 = f_i(i, j)$

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



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

$$j_0 = j + 2$$

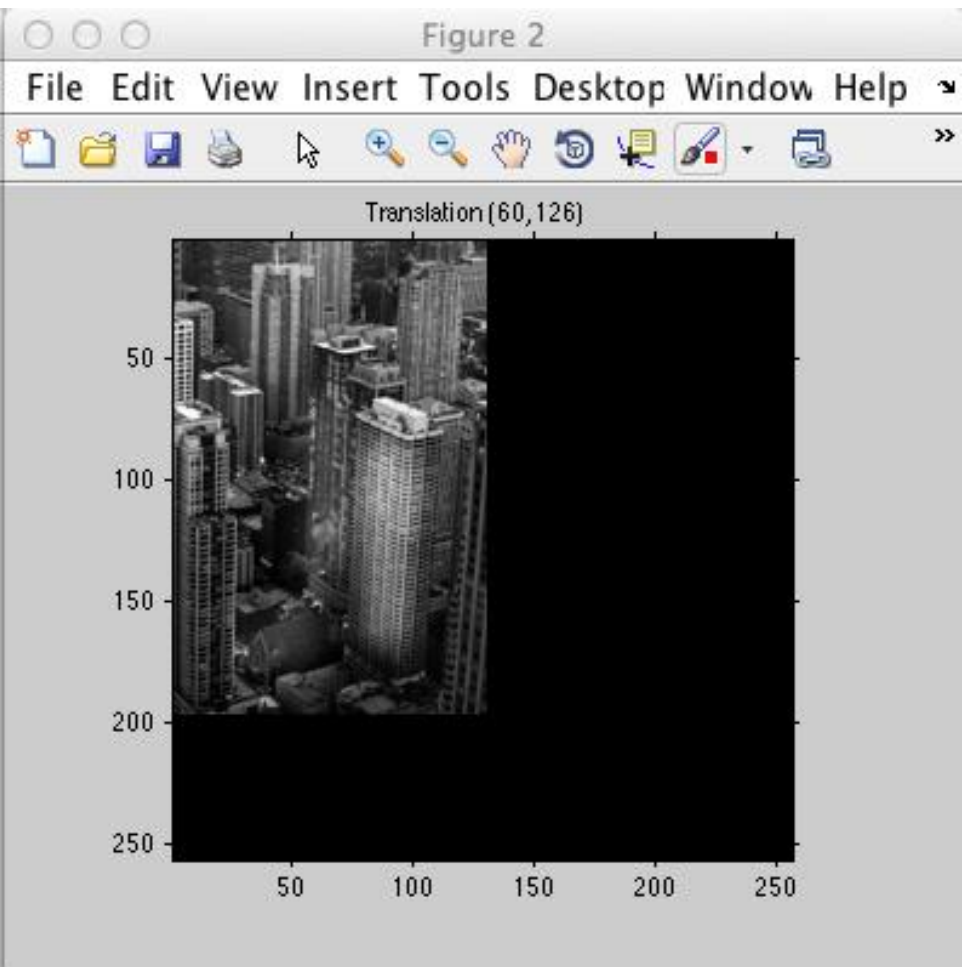
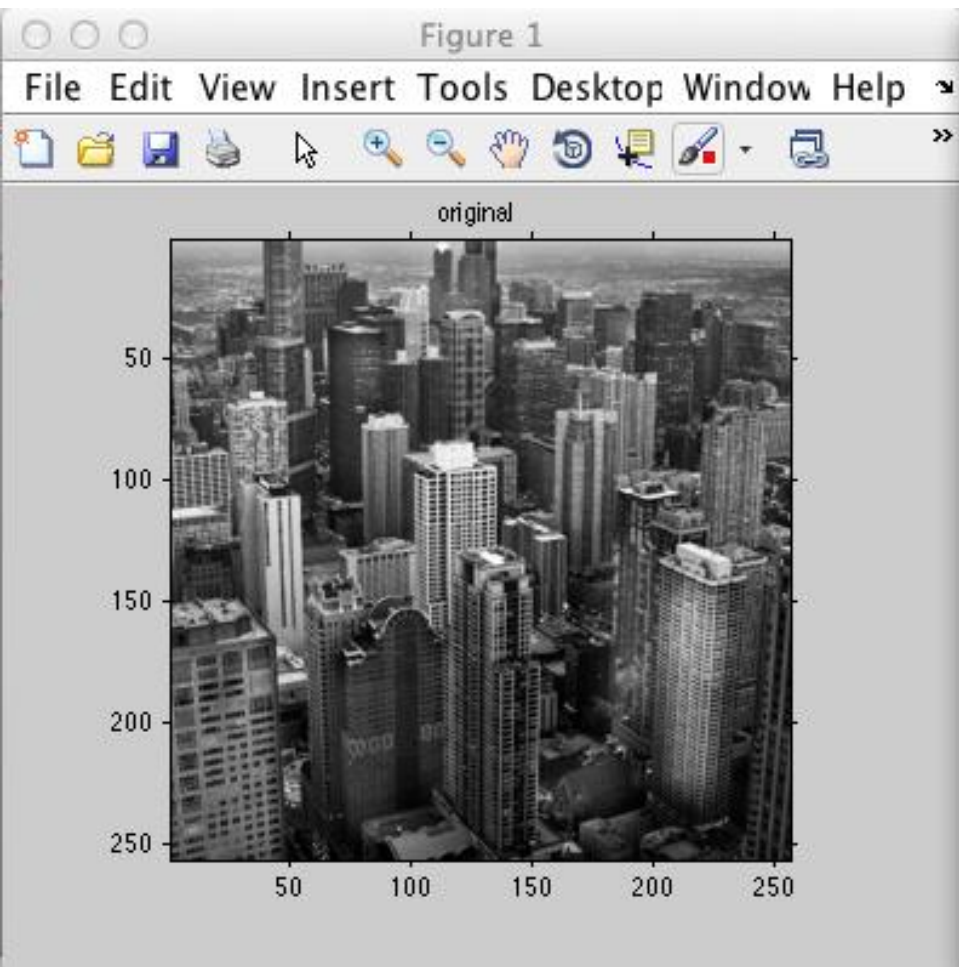


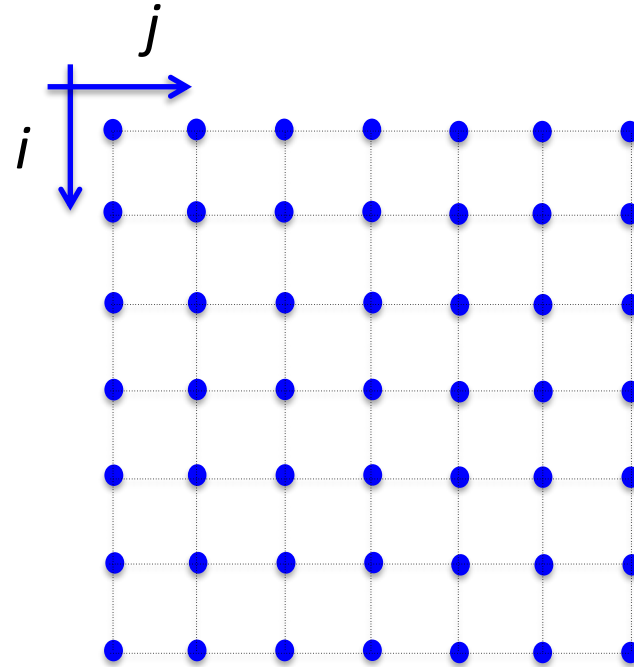
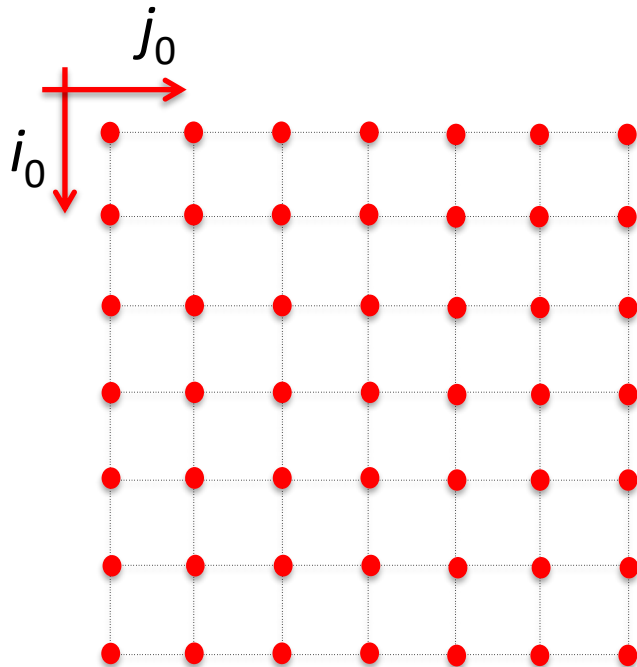
Algorithm:

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

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

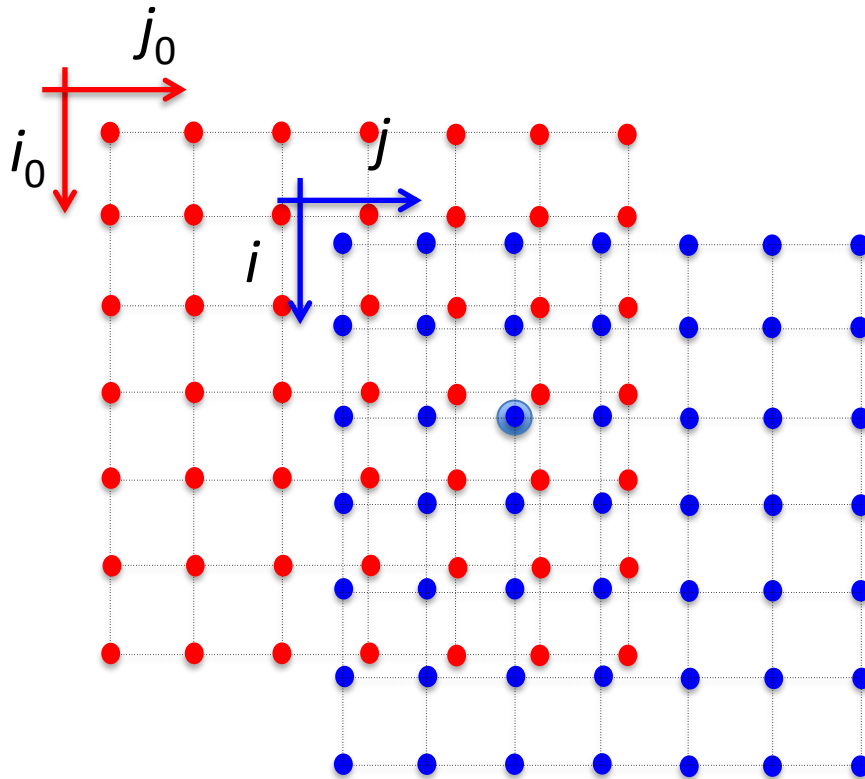
$$j_0 = j + 2$$





Coordinate Transformation: $i_0 = i + 1.25$
(Example: Translation)

$$j_0 = j + 2.75$$



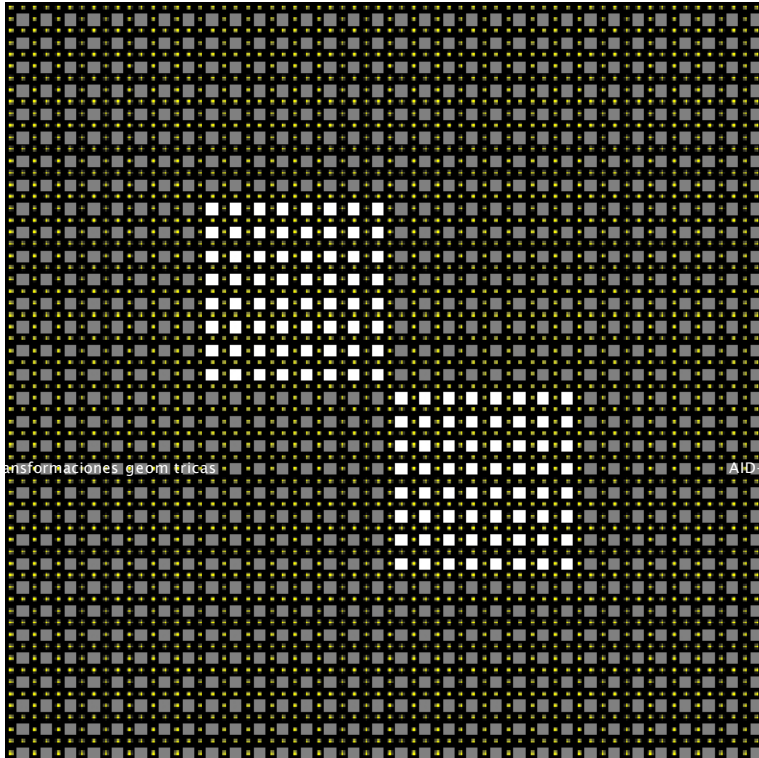
Algorithm:

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

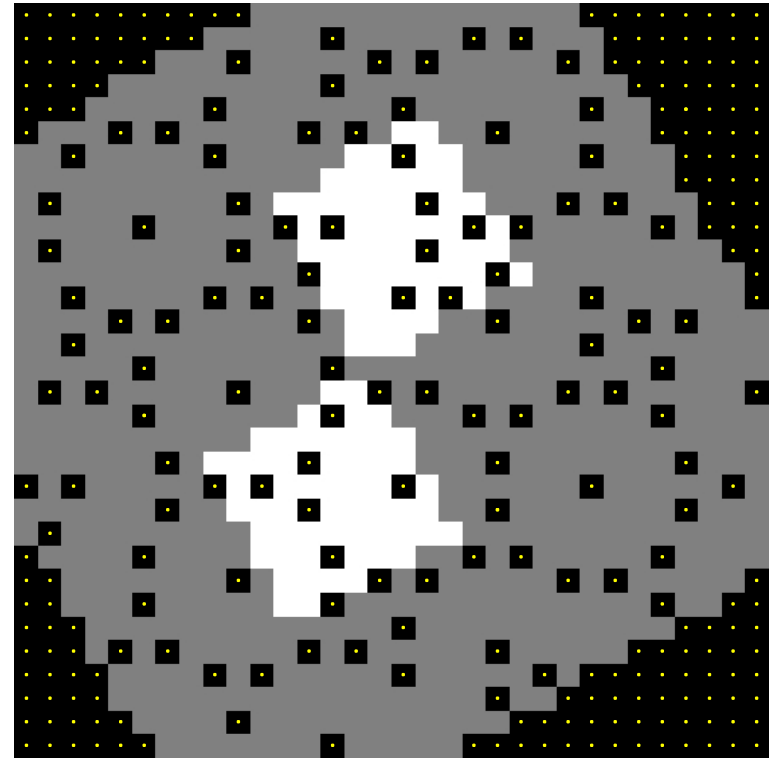
Coordinate Transformation: $i_0 = i + 1.25$
 (Example: Translation)

$$j_0 = j + 2.75$$

Interpolación



(a)



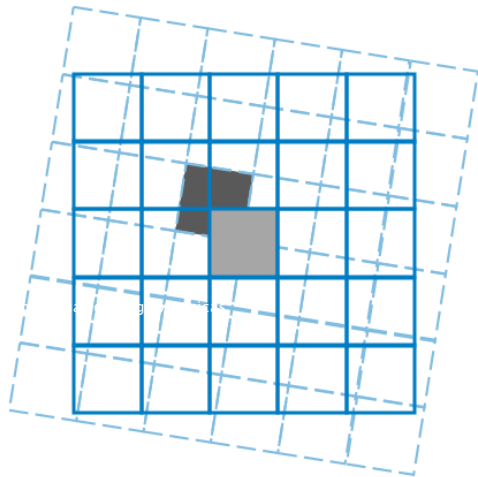
(b)

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.

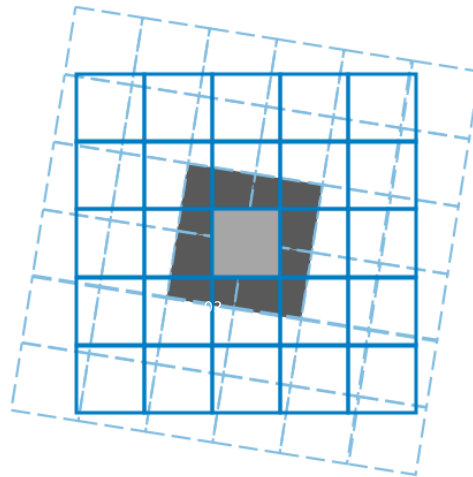
Interpolación

- *Interpolación de los niveles de intensidad* asignar valores de intensidad 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.

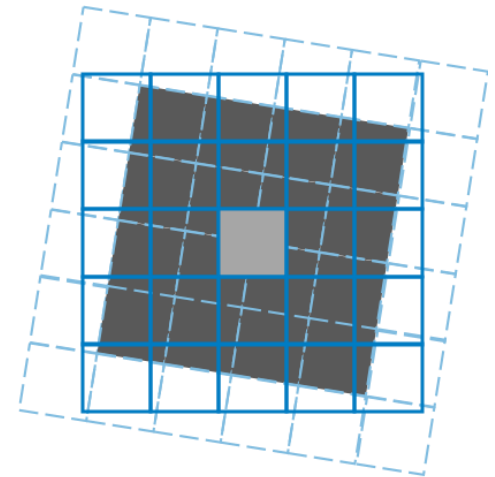
Interpolación



(a)



(b)



(c)

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

Interpolación

Vecino más cercano



Bilineal

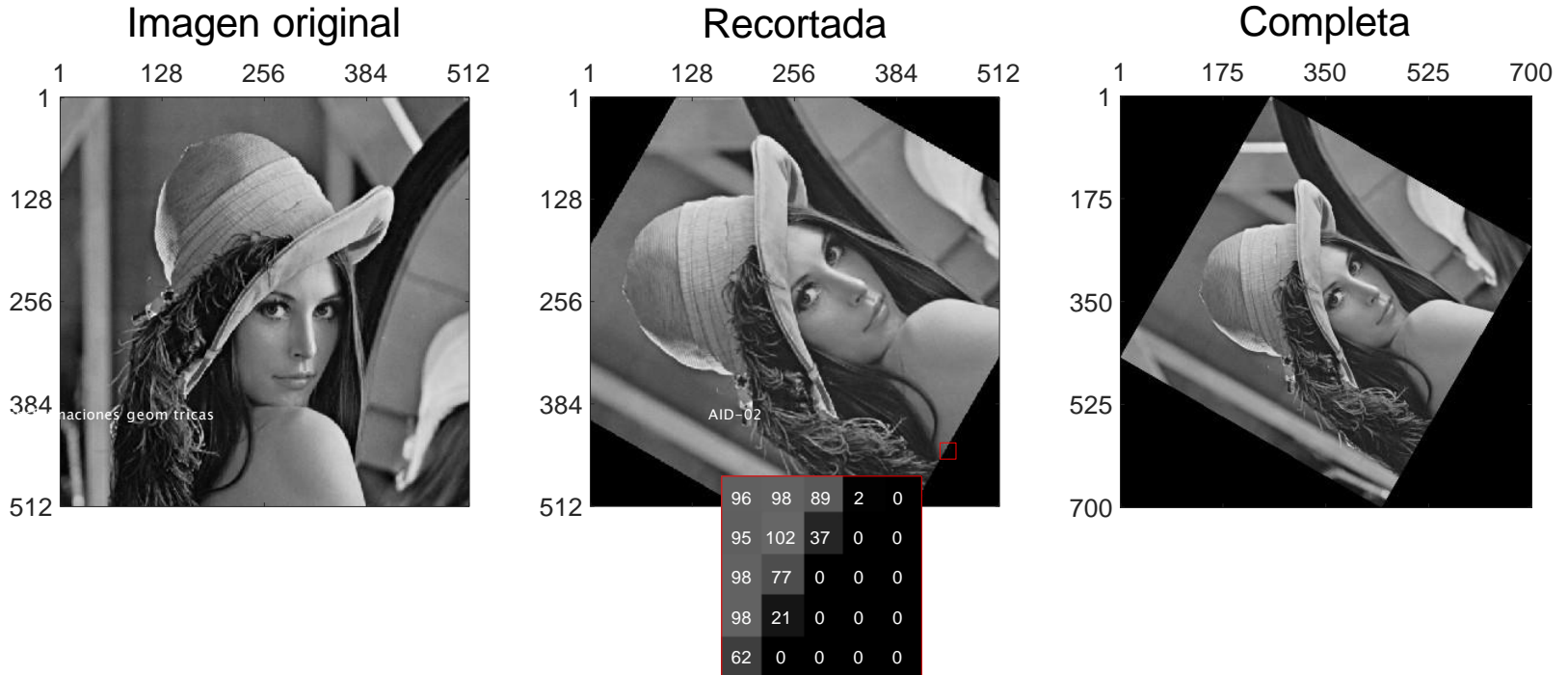


Bicúbico

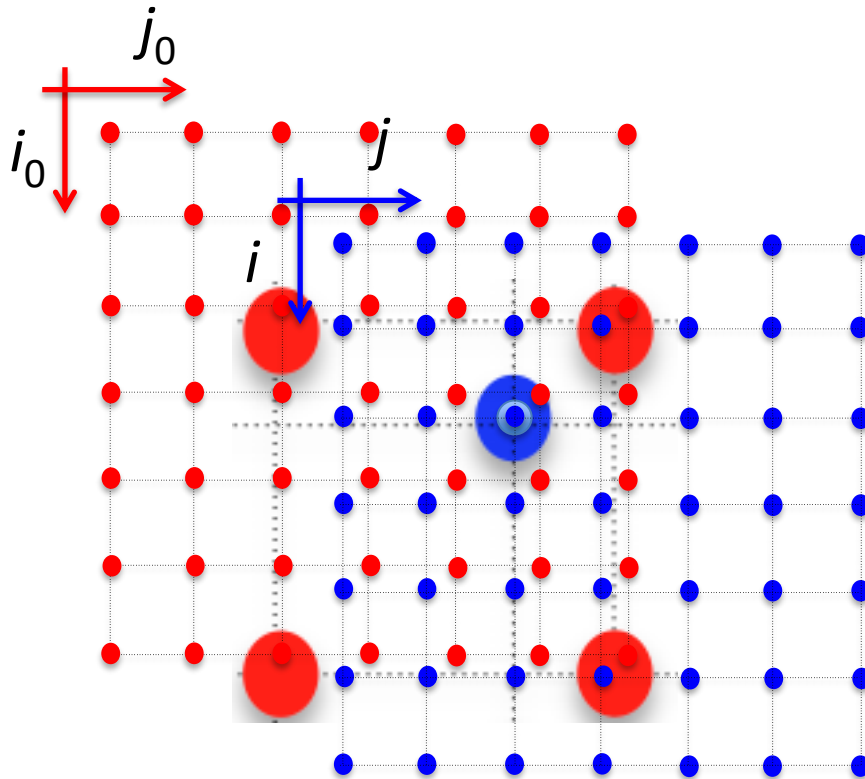


Arriba: imagen escalada 4x. Abajo: imagen rotada 60°.

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.

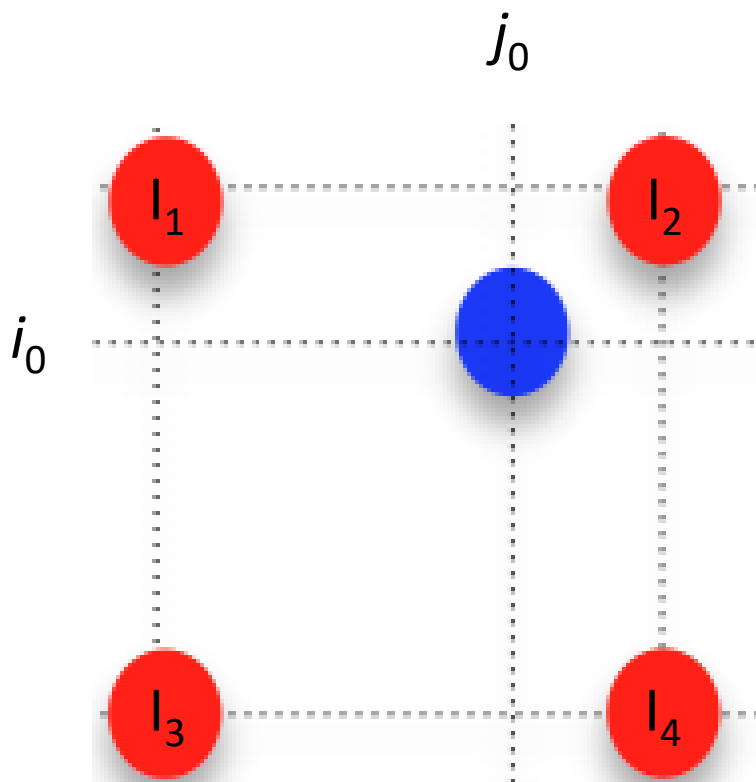


Algorithm:

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

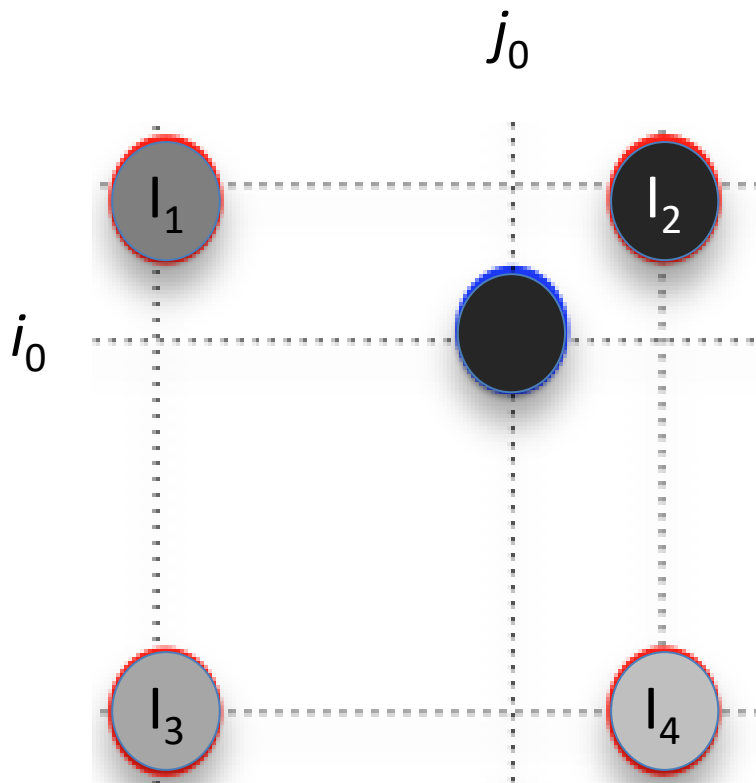
Coordinate Transformation: $i_0 = i + 1.25$
(Example: Translation)

$$j_0 = j + 2.75$$



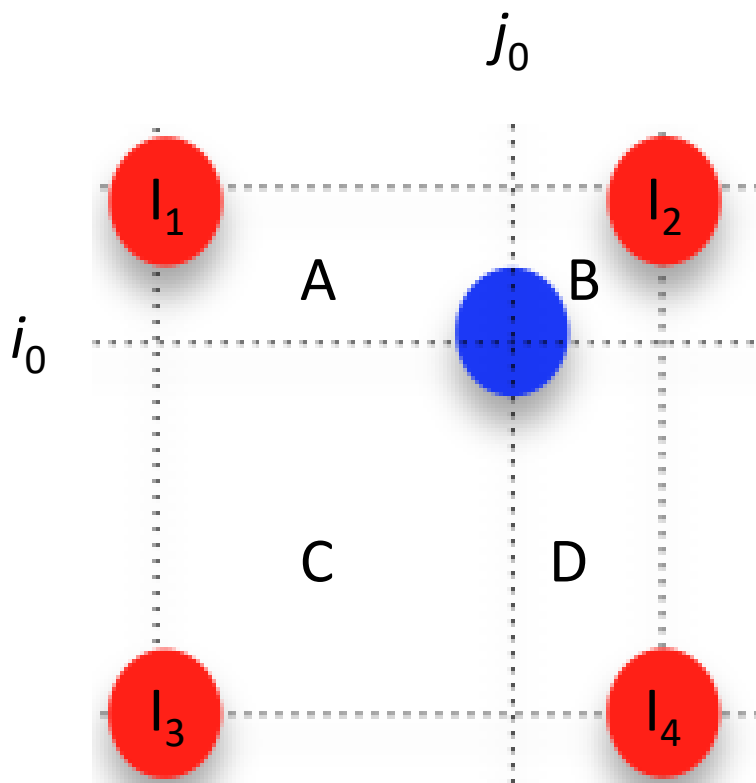
Nearest pixel

$$J(i,j) = \text{interpolation } \{l(i_0, j_0)\} = l_2$$



Nearest pixel

$$J(i,j) = \text{interpolation } \{l(i_0, j_0)\} = l_2$$

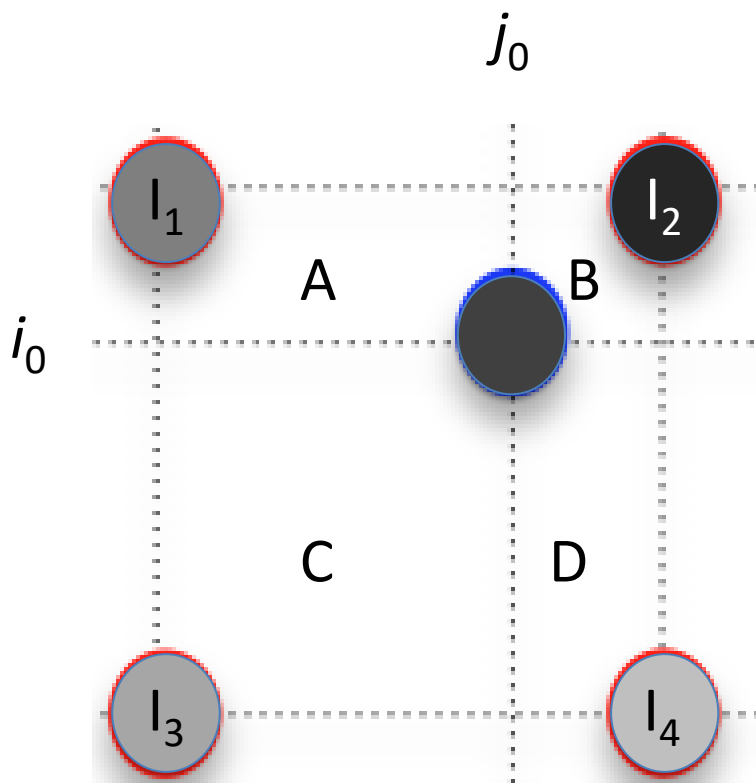


Bilinear interpolation

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

$$AI_4 + BI_3 + CI_2 + DI_1$$

$$A+B+C+D = 1$$

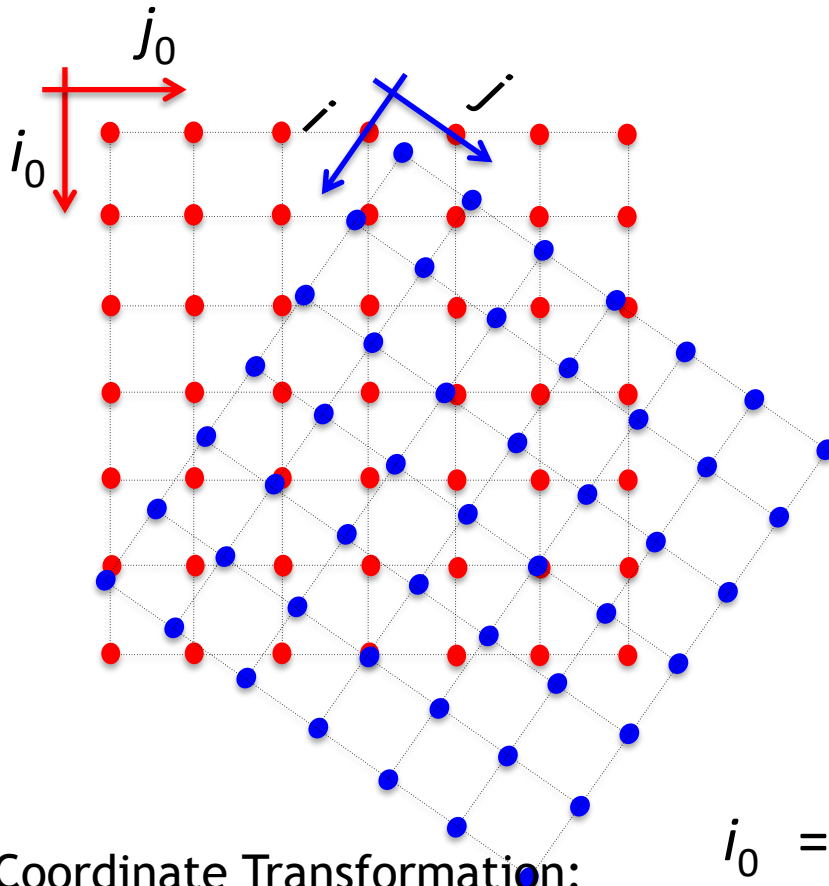


Bilinear interpolation

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

$$AI_4 + BI_3 + CI_2 + DI_1$$

$$A+B+C+D = 1$$



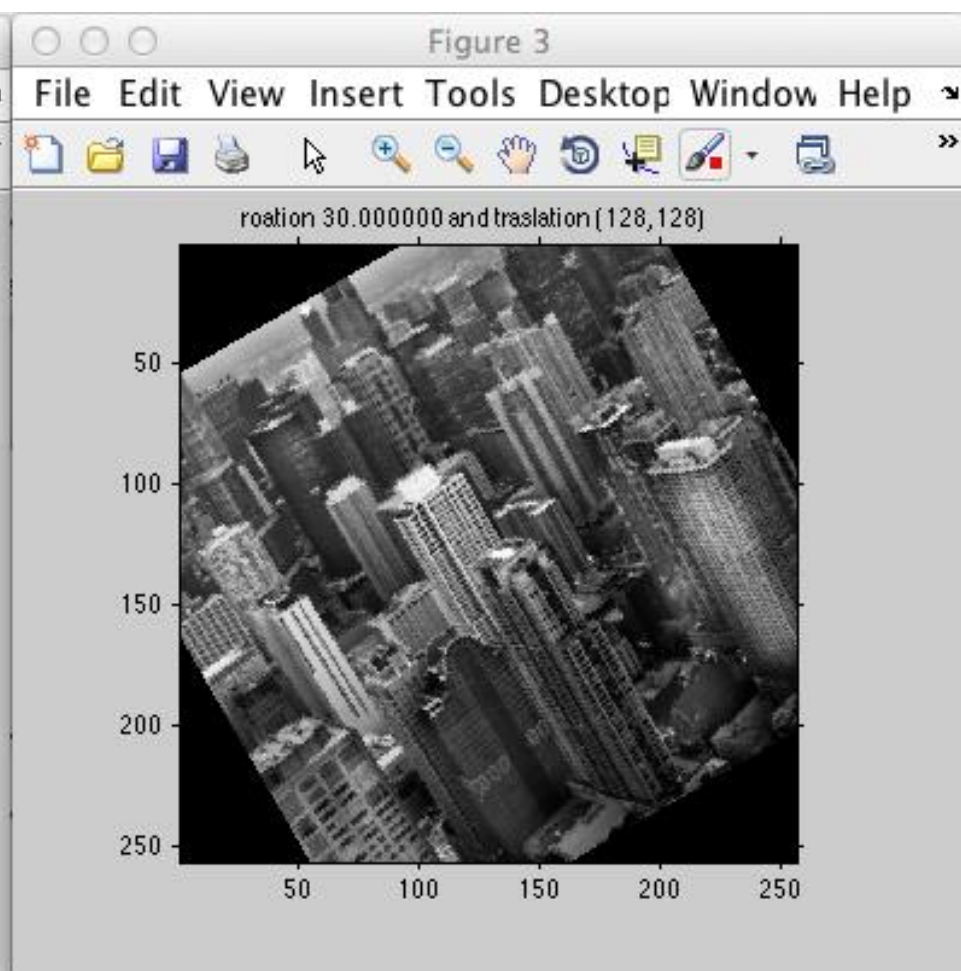
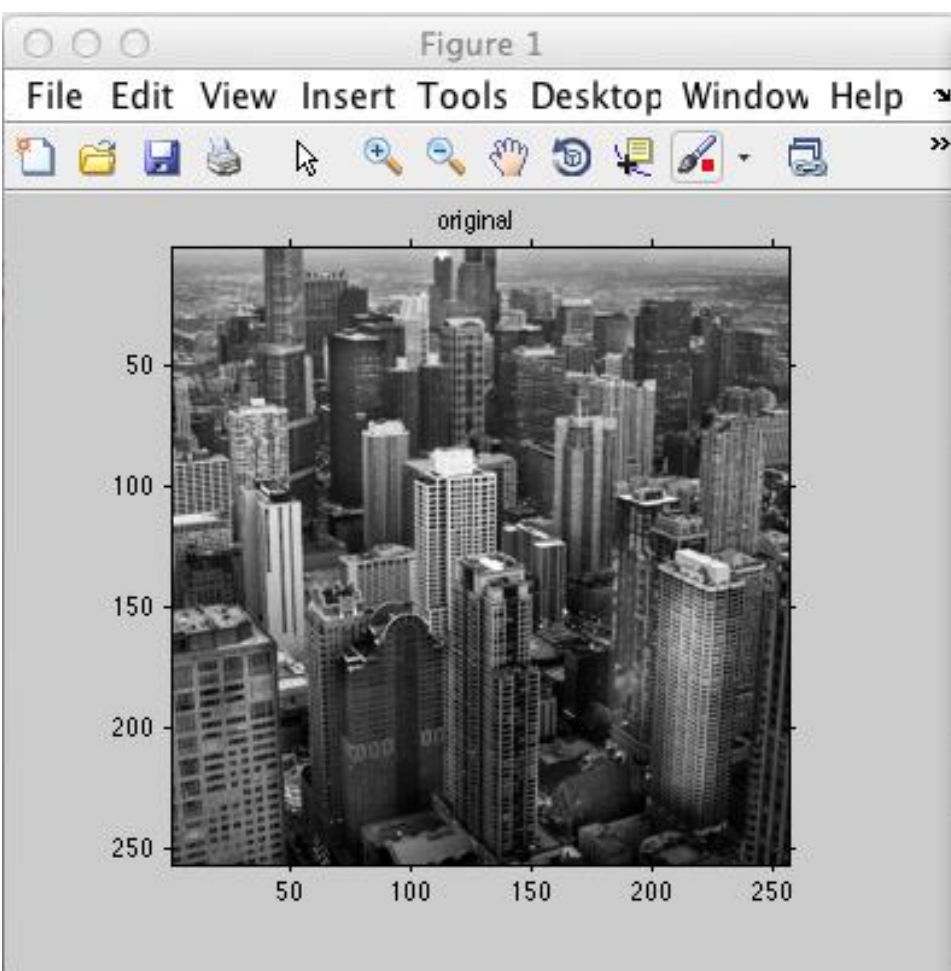
Algorithm:

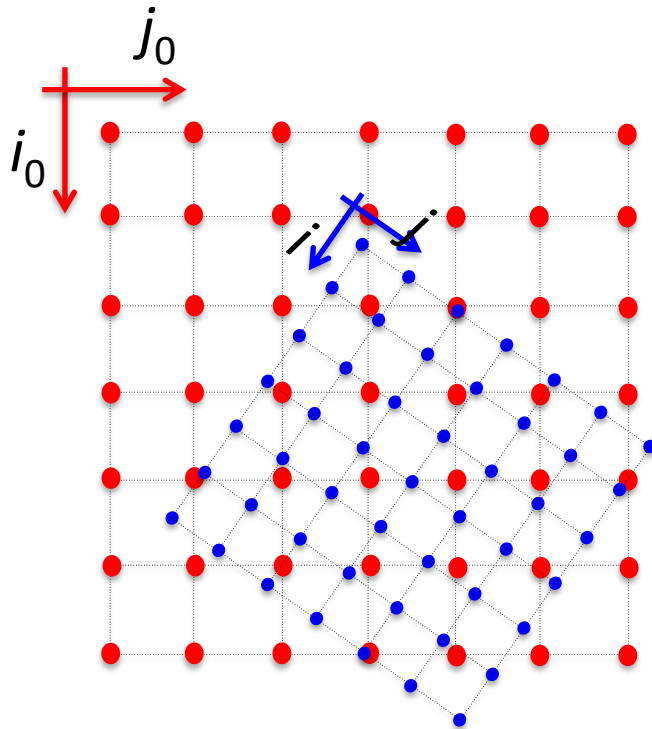
- 1) For each (i, j) of J compute (i_0, j_0) .
- 2) $J(i, j) = \text{interpolation } \{I(i_0, j_0)\}$

Coordinate Transformation:
(Example: Rotation & Translation)

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

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





Algorithm:

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

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, $f_r(x, y)$:

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

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

- Función de error:

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

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:

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

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

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

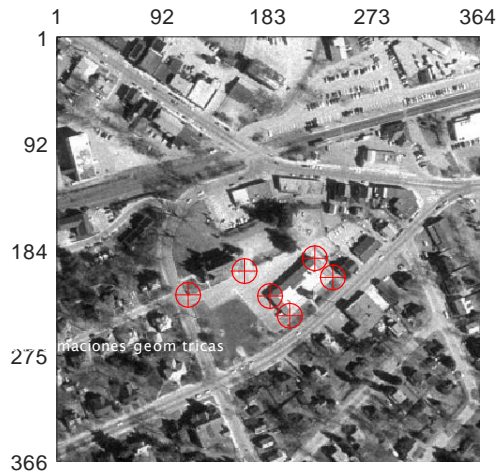
Registro de imágenes

- 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 imagen de referencia, (x, y) :

$$x = c_1v + c_2w + c_3vw + c_4, \quad y = c_5v + c_6w + c_7vw + c_8$$

donde c_1 a c_8 son coeficientes desconocidos que deben estimarse.

Registro de imágenes



(a)



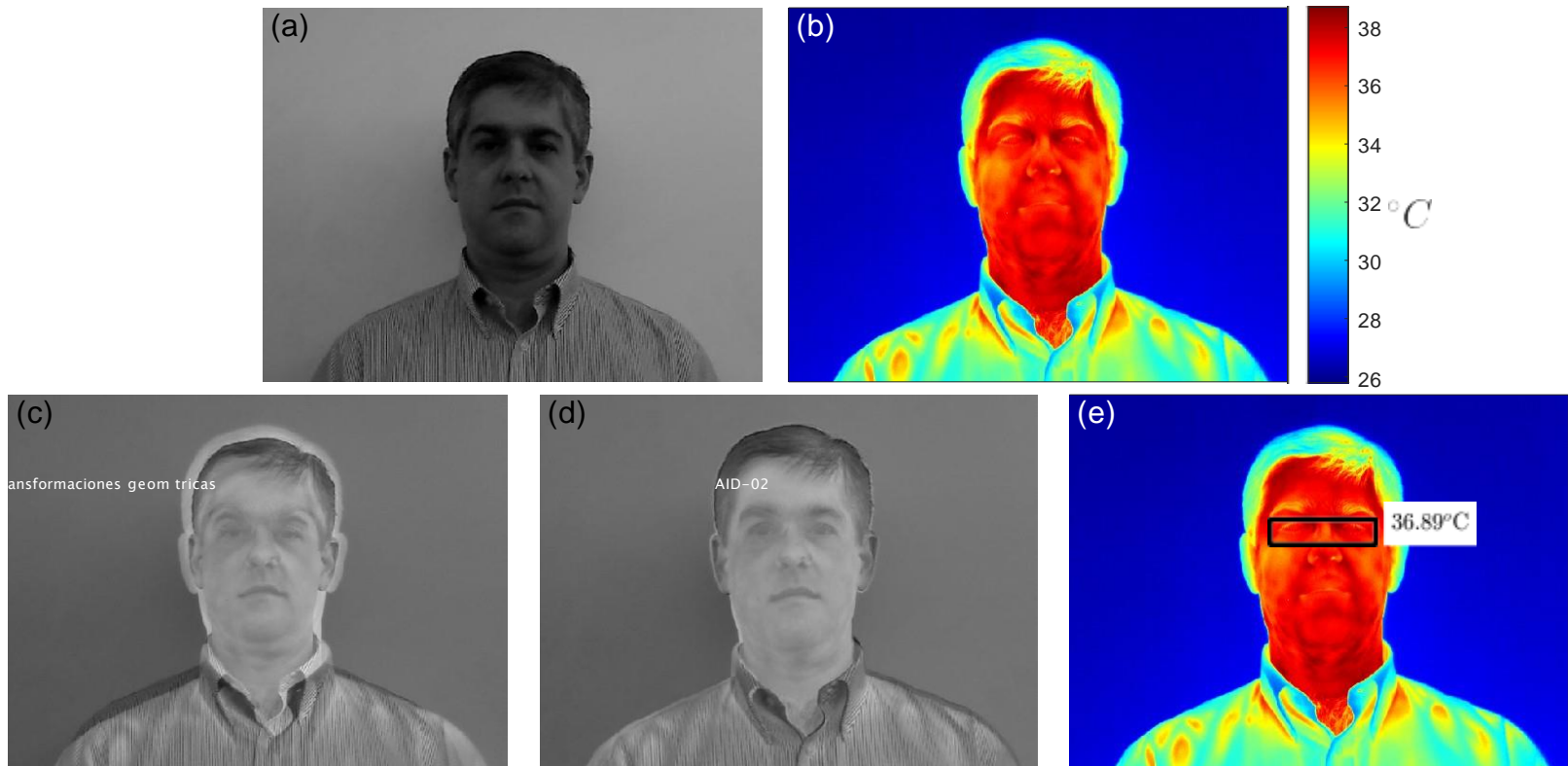
(b)



(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

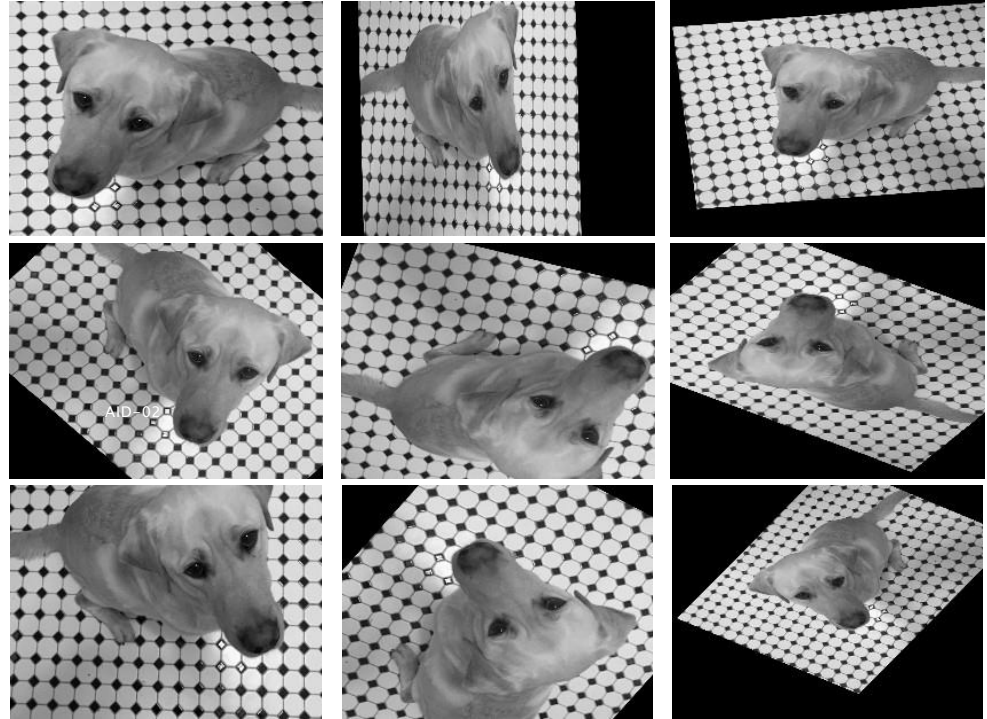


Registro de imágenes obtenidas de sensores diferentes: (a) imagen en escala de grises, (b) imagen térmica, (c) imágenes (a) y (b) desalineadas, (d) imagen en escala de grises registrada alineada con la imagen térmica, y (e) medición de temperatura en la zona de los ojos. La detección de los ojos se realiza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

Imágenes artificiales

Imágenes artificiales

Imagen original



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.