



Ciencia y Tecnología

Secretaría de Ciencia, Humanidades, Tecnología e Innovación

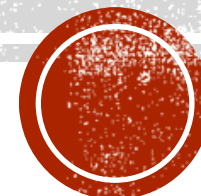


CIMAT
UNIDAD MÉRIDA

TRANSFORMACIONES GEOMÉTRICAS

Francisco J. Hernández López

fcoj23@cimat.mx



TRANSFORMACIONES GEOMÉTRICAS

- Ya vimos como modificar las intensidades de una imagen, ahora vamos a ver cómo podemos modificar las coordenadas de los pixeles de la imagen para cambiar su geometría



Entrada



Traslación



Escalamiento



Rotación

Moeslund, T. B. (2012). Introduction to video and image processing: Building real systems and applications. Springer Science & Business Media.

Transf. Geom. Francisco J. Hernández-López

Ene-Jun 2025

TRANSFORMACIONES GEOMÉTRICAS

- Una transformación geométrica, básicamente calcula las nuevas posiciones $\vec{x}' = (x', y')$ de los pixeles $\vec{x} = (x, y)$ de la imagen f . Este mapeo se denota como:

$$\vec{x}' = w(\vec{x})$$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} w_x(x, y) \\ w_y(x, y) \end{pmatrix}$$

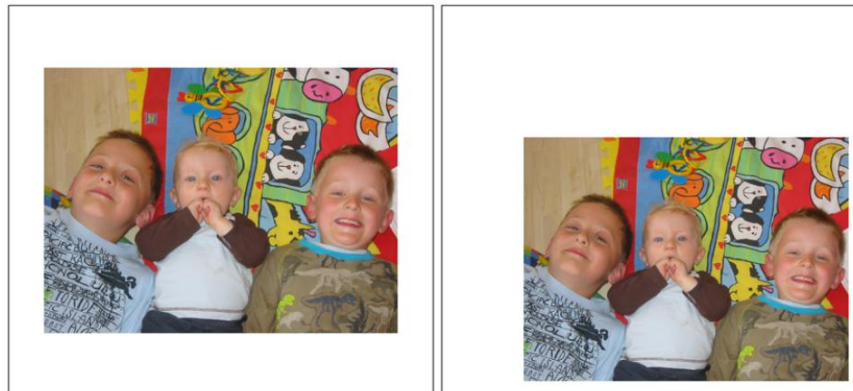
- Transformaciones afines:
 - Traslaciones
 - Escalamiento
 - Rotaciones
 - Cizallamiento (*Shearing*)

TRASLACIÓN

- Traslada las coordenadas de los pixeles de la imagen cierto número de pixeles a la izquierda, derecha, arriba o hacia abajo

$$\vec{x}' = w(\vec{x}) = \vec{x} + \overrightarrow{\Delta d}$$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x + \Delta dx \\ y + \Delta dy \end{pmatrix}$$



Entrada

Traslación

Moeslund, T. B. (2012). Introduction to video and image processing: Building real systems and applications. Springer Science & Business Media.

Transf. Geom. Francisco J. Hernández-López

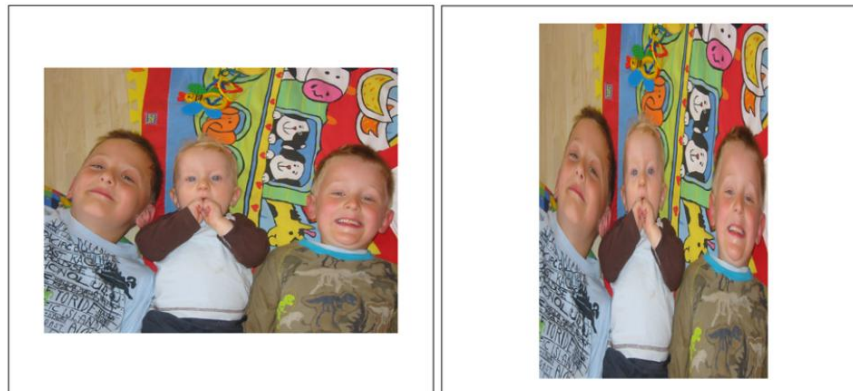
Ene-Jun 2025

ESCALADO

- Esto hace más grande o más pequeña la imagen ya sea en x y/o en y

$$\vec{x}' = w(\vec{x}) = \vec{s}^T \vec{x}, \quad \text{con } \vec{s} = \begin{pmatrix} s_x \\ s_y \end{pmatrix}$$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} s_x x \\ s_y y \end{pmatrix}$$



Entrada

Escalamiento

Moeslund, T. B. (2012). Introduction to video and image processing: Building real systems and applications. Springer Science & Business Media.

Transf. Geom. Francisco J. Hernández-López

Ene-Jun 2025

ROTACIÓN

- Rota la imagen con un cierto ángulo

$$\vec{x}' = w(\vec{x}) = \begin{pmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

Contrario a las
manecillas del reloj

$$\vec{x}' = w(\vec{x}) = \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

Con resp. a las
manecillas del reloj

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x_c \\ y_c \end{pmatrix} + \begin{pmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{pmatrix} \begin{pmatrix} x - x_c \\ y - y_c \end{pmatrix}$$

Con resp. a un punto
central (x_c, y_c)



Entrada



Rotación

Moeslund, T. B. (2012). Introduction to video and image processing: Building real systems and applications. Springer Science & Business Media.

Transf. Geom. Francisco J. Hernández-López

Ene-Jun 2025

CIZALLAMIENTO

- Significa desplazar la imagen a diferente cantidad en x y y , dependiendo de la posición del pixel

$$\vec{x}' = w(\vec{x}) = \begin{pmatrix} 1 & \beta_x \\ \beta_y & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}, \quad \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} x + y\beta_x \\ y + x\beta_y \end{pmatrix}$$



Entrada

Cizallamiento

Moeslund, T. B. (2012). Introduction to video and image processing: Building real systems and applications. Springer Science & Business Media.

Transf. Geom. Francisco J. Hernández-López

Ene-Jun 2025

TRANSFORMACIÓN AFÍN

- Es la combinación de las cuatro transformaciones anteriores

$$\vec{x}' = w(\vec{x}) = \begin{pmatrix} p_1 & p_2 \\ p_4 & p_5 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} + \begin{pmatrix} p_3 \\ p_6 \end{pmatrix}$$



Entrada



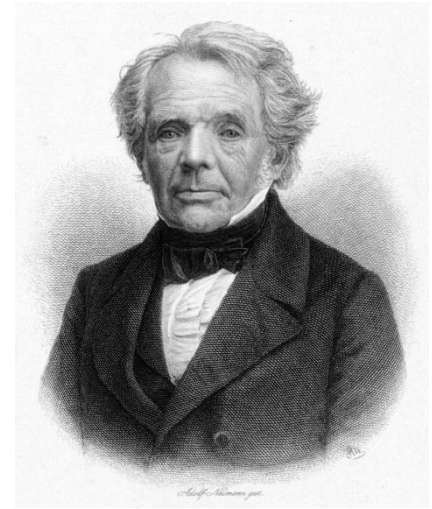
Afín

Moeslund, T. B. (2012). Introduction to video and image processing: Building real systems and applications. Springer Science & Business Media.

Transf. Geom. Francisco J. Hernández-López

Ene-Jun 2025

COORDENADAS HOMOGÉNEAS

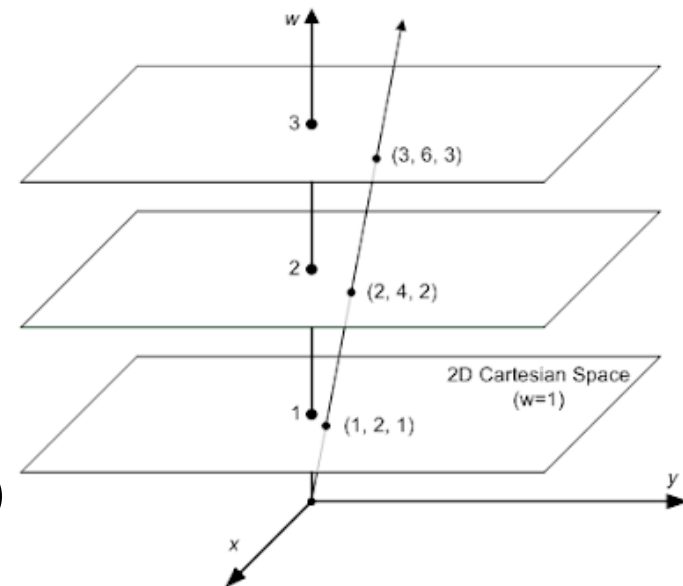


August Ferdinand Möbius
(17/Nov/1790-26/Sep/1868)

- Möbius fue el primero en introducir las Coordenadas Homogéneas en la Geometría Projectiva.
- Utilizado para describir un punto en el espacio proyectivo. Un punto (x, y) se representa por la terna:

$$(\kappa x, \kappa y, \kappa), \text{ con } \kappa \neq 0$$

- $\kappa = 1$, nos da las coordenadas en dos dimensiones
- $\kappa = 0$, nos da los puntos al infinito $(x, y, 0)$



<http://www.fossreview.com/2019/11/homogeneous-coordinates.html>

COORDENADAS HOMOGÉNEAS

- Simplifican la concatenación de mapeos lineales
- Podemos especificar todas las operaciones realizando multiplicaciones de la forma matriz por vector
- Para llevar las coordenadas Cartesianas a coordenadas homogéneas, hacemos lo siguiente:

$$\text{hom}(\vec{x}) = \text{hom} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \underline{\vec{x}}$$

- Note que, si escalamos el vector en coordenadas homogéneas un cierto factor s , este sigue siendo el mismo en coordenadas Cartesianas:

$$\vec{x} = \text{hom}^{-1}(\underline{\vec{x}}) = \text{hom}^{-1}(s\underline{\vec{x}})$$

$$\text{con } \text{hom}^{-1}(\underline{\vec{x}}) = \text{hom}^{-1} \begin{pmatrix} \underline{x} \\ \underline{y} \\ \underline{z} \end{pmatrix} = \frac{1}{\underline{z}} \begin{pmatrix} \underline{x} \\ \underline{y} \end{pmatrix} = \vec{x}$$

Burger, W., Burge, M. J. (2008). Digital Image Processing - An Algorithmic Introduction using Java.. Springer.
ISBN: 978-1-84628-968-2

TRASLACIÓN EN COORD. HOM.

- Ahora podemos definir una translación 2D como un producto matriz por vector:

$$\vec{x}' = \text{hom}^{-1}[T \text{hom}(\vec{x})]$$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \text{hom}^{-1} \left[\begin{pmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{pmatrix} \text{hom} \begin{pmatrix} x \\ y \end{pmatrix} \right]$$

$$= \begin{pmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} = \begin{pmatrix} x + d_x \\ y + d_y \end{pmatrix}$$

Burger, W., Burge, M. J. (2008). Digital Image Processing - An Algorithmic Introduction using Java. Springer.
ISBN: 978-1-84628-968-2

TRANSFORMACIÓN AFÍN EN COORD. HOM.

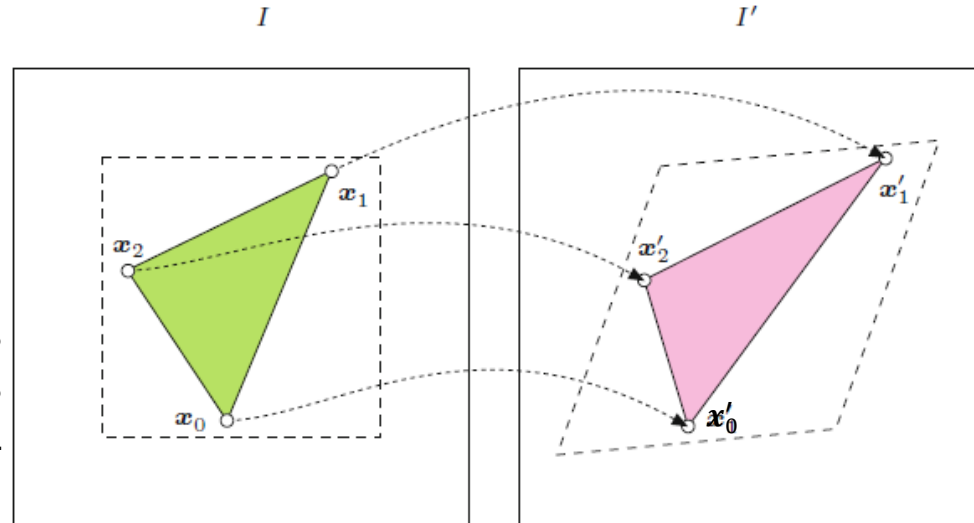
$$\vec{x}' = A_{afin} \vec{x}$$

Parámetros para
Translación

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \text{hom}^{-1} \left[\begin{pmatrix} p_1 & p_2 & p_3 \\ p_4 & p_5 & p_6 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \right] = \begin{pmatrix} p_1 & p_2 & p_3 \\ p_4 & p_5 & p_6 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Parámetros para
Escalado
Rotación
Cizallamiento

Una transf. Afín mapea líneas rectas a líneas rectas, triángulos a triángulos y rectángulos a paralelogramos



Burger, W., Burge, M. J. (2008). Digital Image Processing - An Algorithmic Introduction using Java. Springer. ISBN: 978-1-84628-968-2

ESTIMAR LOS PARAM. DEL MODELO AFÍN

- Dadas tres correspondencias de puntos entre la imagen de entrada y la imagen transformada: $\vec{x}_i = (x_i, y_i)$, $\vec{x}'_i = (x'_i, y'_i)$, $i = 0, 1, 2$

- Tenemos el siguiente sistema de ecuaciones lineales

$$\begin{aligned} x'_0 &= p_1 x_0 + p_2 y_0 + p_3, & y'_0 &= p_4 x_0 + p_5 y_0 + p_6, \\ x'_1 &= p_1 x_1 + p_2 y_1 + p_3, & y'_1 &= p_4 x_1 + p_5 y_1 + p_6, \\ x'_2 &= p_1 x_2 + p_2 y_2 + p_3, & y'_2 &= p_4 x_2 + p_5 y_2 + p_6. \end{aligned}$$

$$\begin{pmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \end{pmatrix} = \begin{pmatrix} x'_0 \\ x'_1 \\ x'_2 \end{pmatrix}, \quad \begin{pmatrix} x_0 & y_0 & 1 \\ x_1 & y_1 & 1 \\ x_2 & y_2 & 1 \end{pmatrix} \begin{pmatrix} p_4 \\ p_5 \\ p_6 \end{pmatrix} = \begin{pmatrix} y'_0 \\ y'_1 \\ y'_2 \end{pmatrix}$$

- Considerando que los vectores son linealm. indep., podemos resolver los sistemas lineales usando la regla de Cramer:

$$\begin{aligned} p_1 &= [y_0(x'_2 - x'_1) + y_1(x'_0 - x'_2) + y_2(x'_1 - x'_0)]/d, \\ p_2 &= [x_0(x'_1 - x'_2) + x_1(x'_2 - x'_0) + x_2(x'_0 - x'_1)]/d, \\ p_3 &= [x_0(y_1 x'_2 - x'_1 y_2) + x_1(y_2 x'_0 - y_0 x'_2) + \\ &\quad x_2(y_0 x'_1 - y_1 x'_0)]/d, \end{aligned}$$

$$d = x_0(y_1 - y_2) + x_1(y_2 - y_0) + x_2(y_0 - y_1).$$

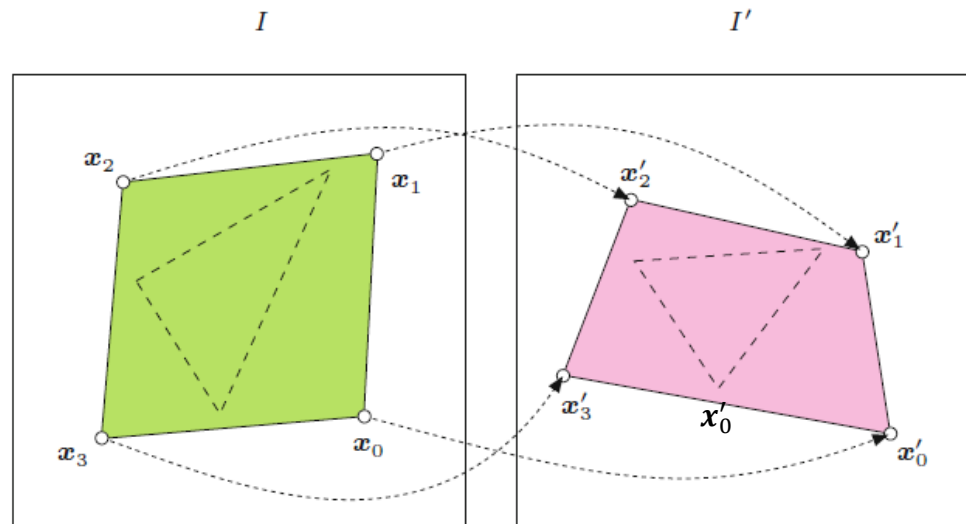
$$\begin{aligned} p_4 &= [y_0(y'_2 - y'_1) + y_1(y'_0 - y'_2) + y_2(y'_1 - y'_0)]/d, \\ p_5 &= [x_0(y'_1 - y'_2) + x_1(y'_2 - y'_0) + x_2(y'_0 - y'_1)]/d, \\ p_6 &= [x_0(y_1 y'_2 - y'_1 y_2) + x_1(y_2 y'_0 - y_0 y'_2) + \\ &\quad x_2(y_0 y'_1 - y_1 y'_0)]/d, \end{aligned}$$

TRANSE. PROYECTIVA (HOMOGRAFÍA)

$$\underline{\vec{x}'} = H \underline{\vec{x}}$$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \text{hom}^{-1} \left[\begin{pmatrix} p_1 & p_2 & p_3 \\ p_4 & p_5 & p_6 \\ p_7 & p_8 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \right] = \frac{1}{p_7x + p_8y + 1} \begin{pmatrix} p_1 & p_2 & p_3 \\ p_4 & p_5 & p_6 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Mapea líneas rectas a líneas rectas y rect. a cuadriláteros. No se conservan el paralelismo entre rectas ni la relación de distancias.



Burger, W., Burge, M. J. (2008). Digital Image Processing - An Algorithmic Introduction using Java. Springer. ISBN: 978-1-84628-968-2

ESTIMAR LOS PARAM. DE LA HOMOGRAFÍA

- Dadas cuatro correspondencias de puntos entre la imagen de entrada y la imagen transformada: $\vec{x}_i = (x_i, y_i)$, $\vec{x}'_i = (x'_i, y'_i)$, $i = 0, 1, 2, 3$
- Tenemos dos ecuaciones lineales por cada correspondencia

$$\begin{aligned}x'_i &= x_i p_1 + y_i p_2 + p_3 - x'_i x_i p_7 - x'_i y_i p_8 \\y'_i &= x_i p_4 + y_i p_5 + p_6 - y'_i x_i p_7 - y'_i y_i p_8\end{aligned}$$

$$\begin{pmatrix} x_0 & y_0 & 1 & 0 & 0 & 0 & -x'_0 x_0 & -x'_0 y_0 \\ 0 & 0 & 0 & x_0 & y_0 & 1 & -y'_0 x_0 & -y'_0 y_0 \\ x_1 & y_1 & 1 & 0 & 0 & 0 & -x'_1 x_1 & -x'_1 y_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -y'_1 x_1 & -y'_1 y_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x'_2 x_2 & -x'_2 y_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -y'_2 x_2 & -y'_2 y_2 \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -x'_3 x_3 & -x'_3 y_3 \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -y'_3 x_3 & -y'_3 y_3 \end{pmatrix} \begin{pmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \\ p_5 \\ p_6 \\ p_7 \\ p_8 \end{pmatrix} = \begin{pmatrix} x'_0 \\ y'_0 \\ x'_1 \\ y'_1 \\ x'_2 \\ y'_2 \\ x'_3 \\ y'_3 \end{pmatrix}$$

Este sistema $M\vec{p} = \vec{b}$, se puede resolver usando métodos numéricos como el algoritmo Gauss-Jordan, Cramer, etc.

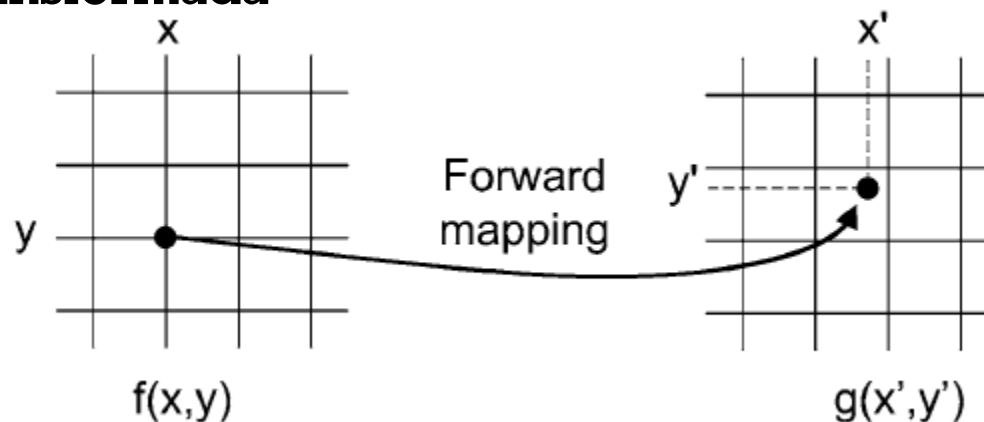
En MatLab:

$$\begin{aligned}\vec{p} &= \text{inv}(M)\vec{b} \\ \vec{p} &= M \backslash \vec{b}\end{aligned}$$

Burger, W., Burge, M. J. (2008). Digital Image Processing - An Algorithmic Introduction using Java. Springer. ISBN: 978-1-84628-968-2

MAPEO DIRECTO

- Consiste en dos pasos:
 1. Definir los coeficientes de la transformación
 2. Recorrer las posiciones de la imagen, para cada $\vec{x}$ calculamos su nueva posición $\vec{x}'$
- Con este tipo de mapeo podríamos observar huecos en la imagen transformada



Moeslund, T. B. (2012). Introduction to video and image processing: Building real systems and applications. Springer Science & Business Media.

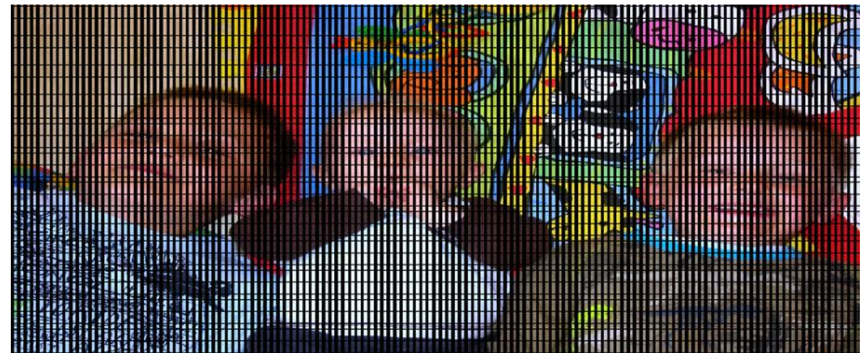
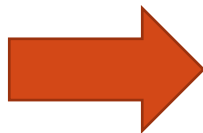
EJEMPLO

- Imagine que tenemos una imagen de 300×200 pixeles y queremos **escalarla** a una imagen de 510×200 pixeles
- Primero, definimos los factores $s_x = 510/300 = 1.7$ y $s_y = 1$
- Luego, aplicamos la transformación:

x	0	1	2	3	4	5	6	7	8	...	300
x'	0	1.7	3.4	5.1	6.8	8.5	10.2	11.9	13.6	...	510



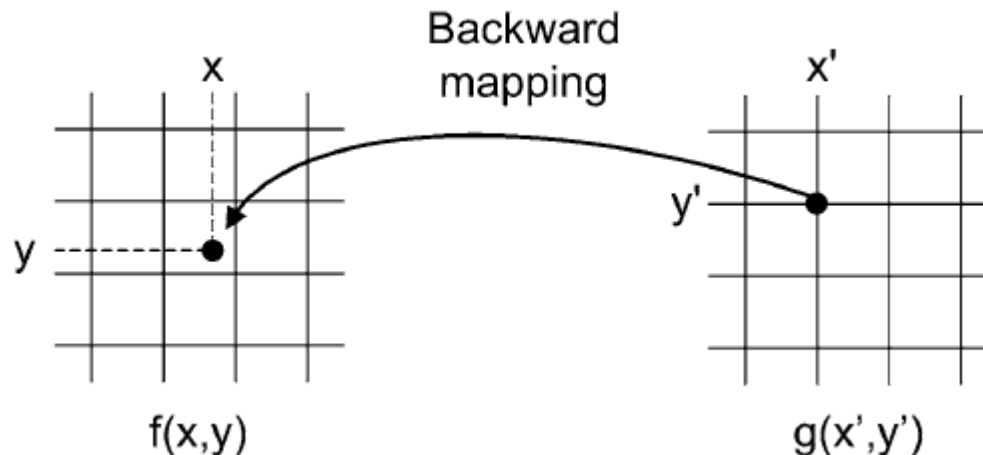
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$



Moeslund, T. B. (2012). Introduction to video and image processing: Building real systems and applications. Springer Science & Business Media.

MAPEO INVERSO

- La idea es recorrer los pixeles $\vec{x}'$ de la imagen transformada g y con la transformación inversa, estimar la posición $\vec{x}$ de la imagen de entrada f
- Con este tipo de mapeo ya no aparecen los huecos en la imagen transformada



Moeslund, T. B. (2012). Introduction to video and image processing: Building real systems and applications. Springer Science & Business Media.

TRANSFORMACIÓN INVERSA

- Transformación afín:

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} p_1 & p_2 & p_3 \\ p_4 & p_5 & p_6 \\ 0 & 0 & 1 \end{bmatrix}^{-1} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

Escalado:

$$\begin{bmatrix} 1/s_x & 0 & 0 \\ 0 & 1/s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Rotación:

$$\begin{bmatrix} \cos \theta & \sin \theta & 0 \\ -\sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Cizallamiento:

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

Moeslund, T. B. (2012). Introduction to video and image processing: Building real systems and applications. Springer Science & Business Media.

IMPLEMENTACIÓN

```
Image_Width_Output = Image_Width_Input * Sx;  
Image_Height_Output = Image_Height_Input * Sy;  
for (y_output = 0; y_output < Image_Height_Output;  
    y_output++)  
{  
    for (x_output = 0; x_output < Image_Width_Output;  
        x_output++)  
    {  
        x_input = 1/Sx * x_output;  
        y_input = 1/Sy * y_output;  
        g(x_output, y_output) = f(x_input, y_input);  
    }  
}
```

Si no son posiciones enteras,
podríamos necesitar una interpolación

Moeslund, T. B. (2012). Introduction to video and image processing: Building real systems and applications. Springer Science & Business Media.

Transf. Geom. Francisco J. Hernández-López

Ene-Jun 2025

GRACIAS POR SU ATENCIÓN

Francisco J. Hernandez-Lopez

fcoj23@cimat.mx

WebPage:

www.cimat.mx/~fcoj23

