Capítulo III: Transformaciones Geométricas

Procesamiento Digital de Imágenes

Contenido:

Contenido:

Transformaciones espaciales

Definición

Son operaciones que alteran la disposición espacial de los píxeles en una imagen.

- No cambian los valores de intensidad (color/brillo) de los píxeles directamente, sino dónde se ubican.
- El objetivo es modificar la forma, tamaño o posición de la imagen o de objetos dentro de ella.
- Es una operación bidireccional: se mapean puntos del espacio de entrada al espacio de salida, y viceversa.

Propósito

- Corrección de distorsiones.
- Registro de imágenes (alineación de múltiples imágenes).
- Normalización de tamaño y orientación.
- Creación de efectos visuales.

Transformaciones espaciales

Aplicaciones Cruciales en el Mundo Real

- Corrección de Distorsiones: Lentes de cámara, escaneos torcidos, perspectiva.
- Normalización: Estandarizar imágenes para análisis (reconocimiento de objetos, detección de rostros).
- Registro de Imágenes: Alineación de múltiples imágenes (médicas, satelitales, fotografía computacional).
- Composición de Imágenes: Creación de panoramas, incrustación de objetos, efectos visuales en cine/video.
- Realidad Aumentada/Virtual: Superposición de objetos virtuales en escenas reales.
- Visión por Computadora: Calibración de cámaras, reconstrucción 3D.

Transformaciones espaciales

Ejemplo prácticos

Transformaciones espaciales

Toda transformación geométrica se descompone en dos fases interdependientes:

Transformación Espacial de Coordenadas (Mapeo):

Define la relación matemática entre las coordenadas (x,y) en la imagen de entrada y sus nuevas coordenadas (x' , y') en la imagen de salida.
Puede ser un mapeo directo (de entrada a salida) o inverso (de salida a entrada).

Transformación Espacial de Coordenadas (Mapeo):

Dado que las coordenadas transformadas rara vez caen en ubicaciones exactas de píxeles enteros, este paso estima el valor de intensidad para cada píxel en la cuadrícula de salida.
Es esencial para rellenar los "agujeros" y crear una imagen continua.

Mapeo Directo vs. Mapeo Inverso

Mapeo Directo (Forward Mapping):

Para cada píxel (x,y) en la imagen de entrada, se calcula su nueva posición (x ' ,y ') en la imagen de salida.
Problema: Múltiples píxeles de entrada pueden mapear a la misma posición de salida, o algunas posiciones de salida pueden quedar sin asignar (agujeros).
Difícil de manejar la colisión de píxeles y los vacíos.

Mapeo Inverso (Inverse Mapping):

Para cada píxel (x ' ,y ') en la imagen de salida, se calcula la posición correspondiente (x,y) en la imagen de entrada utilizando la transformación inversa T^{-1} .
Se interpola el valor de intensidad en (x,y) y se asigna a (x ' ,y ').
Ventaja: Asegura que cada píxel de la imagen de salida obtenga un valor, evitando agujeros. Es el método estándar en bibliotecas como OpenCV.

Transformaciones espaciales

Traslación (Translation)

Desplaza cada píxel de la imagen por una cantidad fija en las direcciones horizontal (t_x) y vertical (t_y).

Fórmulas Cartesianas:

$$x' = x + t_x$$

$$y' = y + t_y$$

Parámetros: Se requieren dos parámetros: (t_x, t_y), que representan el vector de traslación.

Propiedades: Es una transformación rígida. Preserva la forma, el tamaño y la orientación de la imagen.

Transformaciones espaciales

Ejemplo Numérico - Traslación

Ejemplo: Trasladar un píxel en la posición original (10, 25) con un vector de traslación (t_x, t_y) = (5,−8).

Paso 1: Identificar las coordenadas originales y el vector de traslación.

$$(x, y) = (10, 25)$$

$$t_x = 5, t_y = -8$$

Paso 2: Aplicar las fórmulas de traslación.

$$x' = x + t_x = 10 + 5 = 15$$

$$y' = y + t_y = 25 + (-8) = 17$$

Resultado: El píxel se mueve a la nueva posición (15,17).

Rotación (Rotation)

Gira la imagen alrededor de un punto fijo (centro de rotación) por un ángulo θ.

Fórmulas Cartesianas (alrededor del origen (0,0) en sentido antihorario):

$$x' = x\cos\theta - y\sin\theta$$

$$y' = x\sin\theta + y\cos\theta$$

Parámetros: Se requiere el ángulo de rotación θ y el centro de rotación.

Propiedades: También es una transformación rígida. Preserva la forma y el tamaño.

Ejemplo Numérico - Traslación

Ejemplo: Rotar un píxel en la posición original (3,4) por 90° (o π/2 radianes) en sentido antihorario alrededor del origen.

Paso 1: Identificar las coordenadas originales y el ángulo.

$$(x, y) = (3, 4)$$

$$\theta = 90^\circ$$

Paso 2: Calcular sinθ y cosθ.

$$\cos(90^\circ) = 0$$

$$\sin(90^\circ) = 1$$

Paso 2: Aplicar las fórmulas de rotación.

$$x' = (3 \cdot 0) - (4 \cdot 1) = -4$$

$$y' = (3 \cdot 1) + (4 \cdot 0) = 3$$

Resultado: El píxel se mueve a la nueva posición (−4,3)

Transformaciones espaciales

Escalado (Scaling)

Modifica el tamaño de la imagen aplicando factores de escala s_x y s_y a lo largo de los ejes x e y.

Fórmulas Cartesianas (con respecto al origen):

$$x' = x \cdot s_x$$

$$y' = y \cdot s_y$$

Parámetros: Se requieren dos factores de escala: s_x (escala horizontal) y s_y (escala vertical).

Si $s_x = s_y$, es un escalado uniforme (preserva la relación de aspecto).

Si $s_x \neq s_y$, es un escalado no uniforme (distorsiona la relación de aspecto).

Propiedades: No es una transformación rígida. Cambia el tamaño y puede cambiar la forma.

Contenido:

Matrices de Transformación Homogéneas

Aplicación:

- Las operaciones de traslación en coordenadas cartesianas son aditivas $(x + t_x)$, mientras que la rotación y escalado son multiplicativas.
- Esto dificulta la combinación de múltiples transformaciones en una sola operación.
- Necesitamos un marco matemático unificado.
- Solución: Introducir una tercera coordenada (la coordenada homogénea w) para representar puntos 2D en un espacio 3D proyectivo.

Matriz de Traslación Homogénea

Traslación Unificada:

Fórmula:

$$\mathbf{M}_{Traslación} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

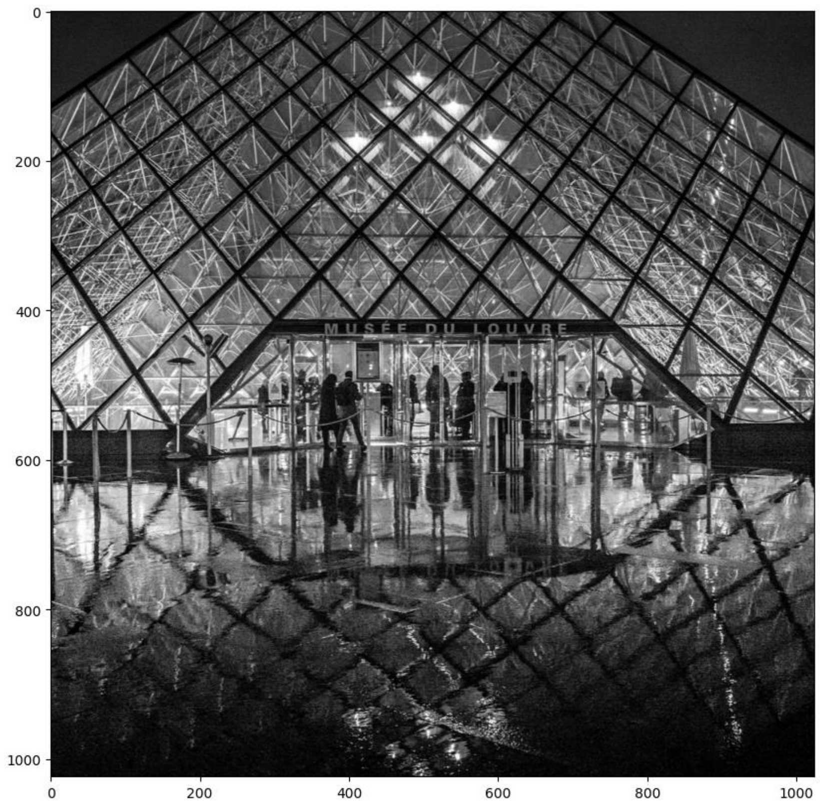
Operación:

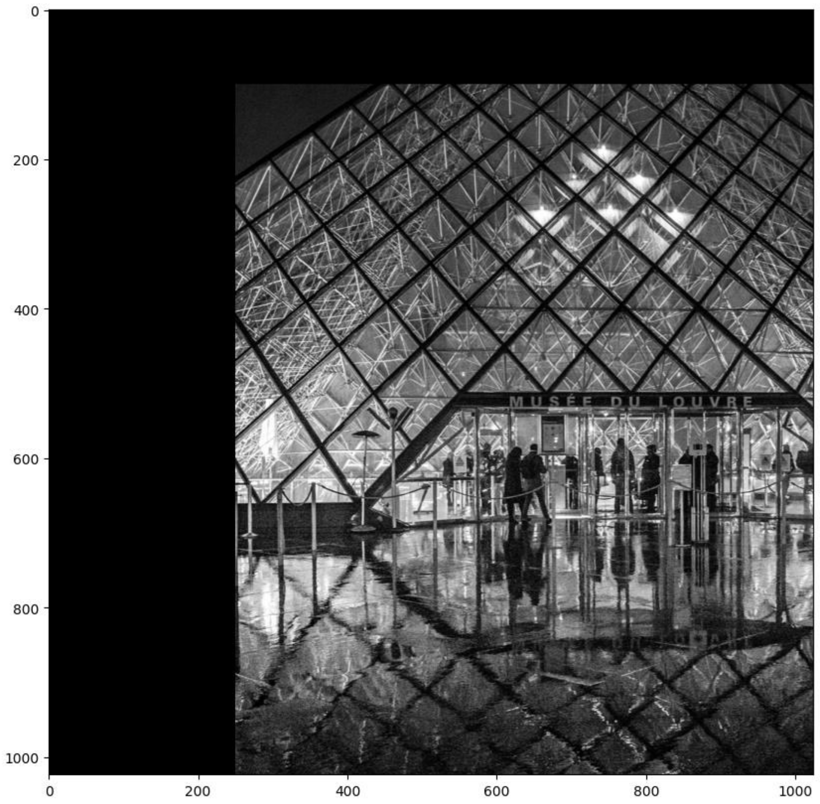
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x + t_x \\ y + t_y \\ 1 \end{bmatrix}$$

Ejemplo Numérico: Trasladar $(10, 25)$ por $(5, -8)$.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 5 \\ 0 & 1 & -8 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 10 \\ 25 \\ 1 \end{bmatrix} = \begin{bmatrix} 10 + 5 \\ 25 - 8 \\ 1 \end{bmatrix} = \begin{bmatrix} 15 \\ 17 \\ 1 \end{bmatrix}$$

Traslación


0
200
400
600
800
1000
0
200
400
600
800
1000
MUSÉE DU LOUVRE


0
200
400
600
800
1000
0
200
400
600
800
1000
MUSÉE DU LOUVRE

Matriz de Traslación Homogénea

Rotación Simplificada:

Fórmula (alrededor del origen):

$$\mathbf{M}_{Rotación} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

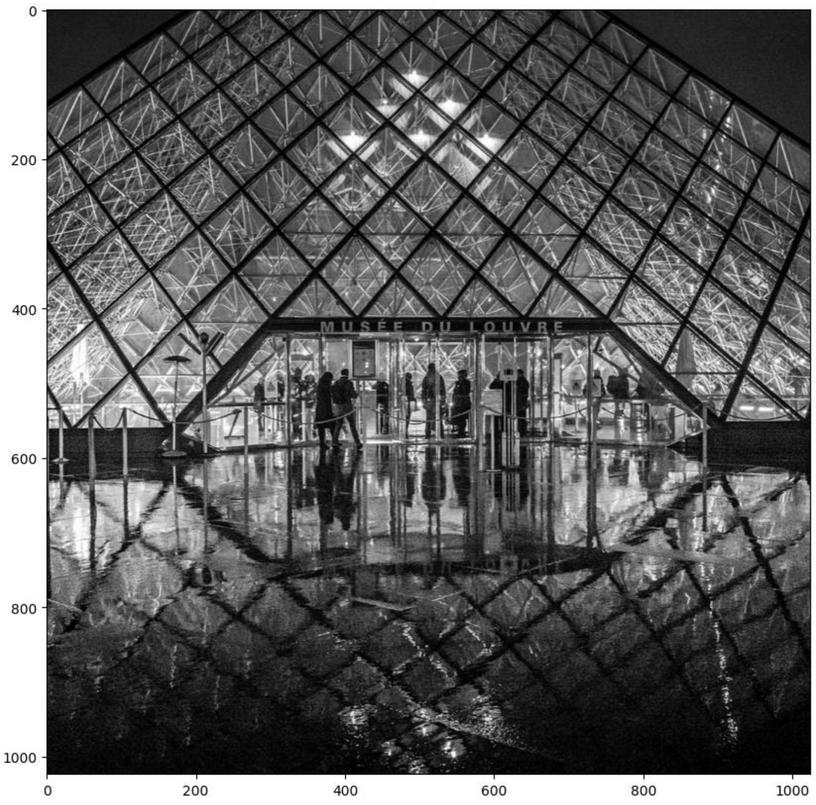
Ejemplo Numérico: Rotar $(3,4)$ por 90° antihorario

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 4 \\ 1 \end{bmatrix} = \begin{bmatrix} -4 \\ 3 \\ 1 \end{bmatrix}$$

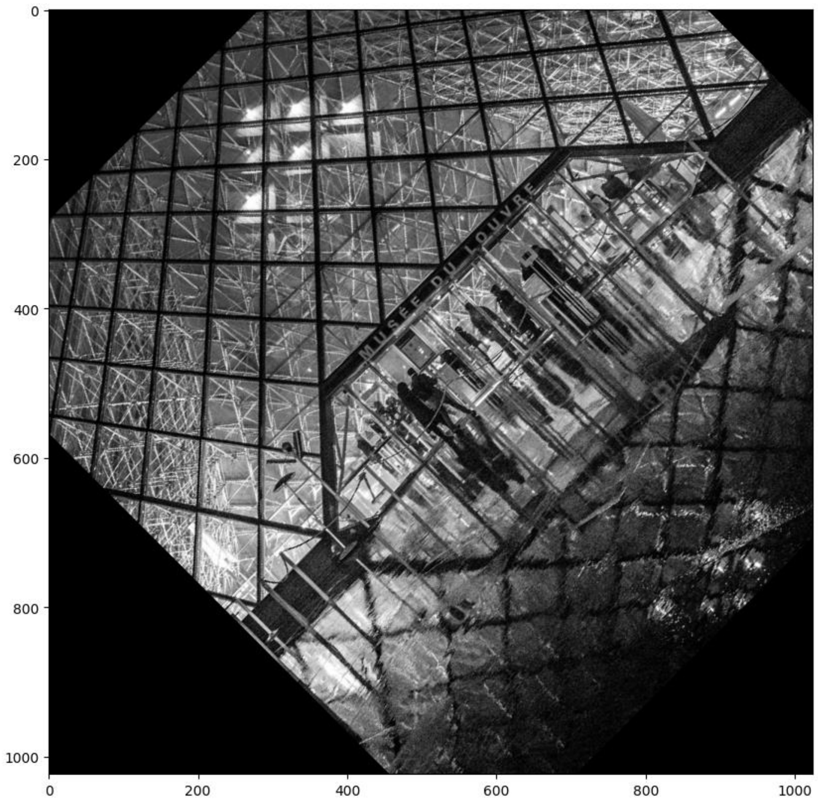
Operación:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x\cos\theta - y\sin\theta \\ x\sin\theta + y\cos\theta \\ 1 \end{bmatrix}$$

Rotación



0
200
400
600
800
1000
0
200
400
600
800
1000
MUSÉE DU LOUVRE



0
200
400
600
800
1000
0
200
400
600
800
1000
MUSÉE DU LOUVRE

Matriz de Traslación Homogénea

Escalado Unificado:

Fórmula (alrededor del origen):

$$\mathbf{M}_{Escalado} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

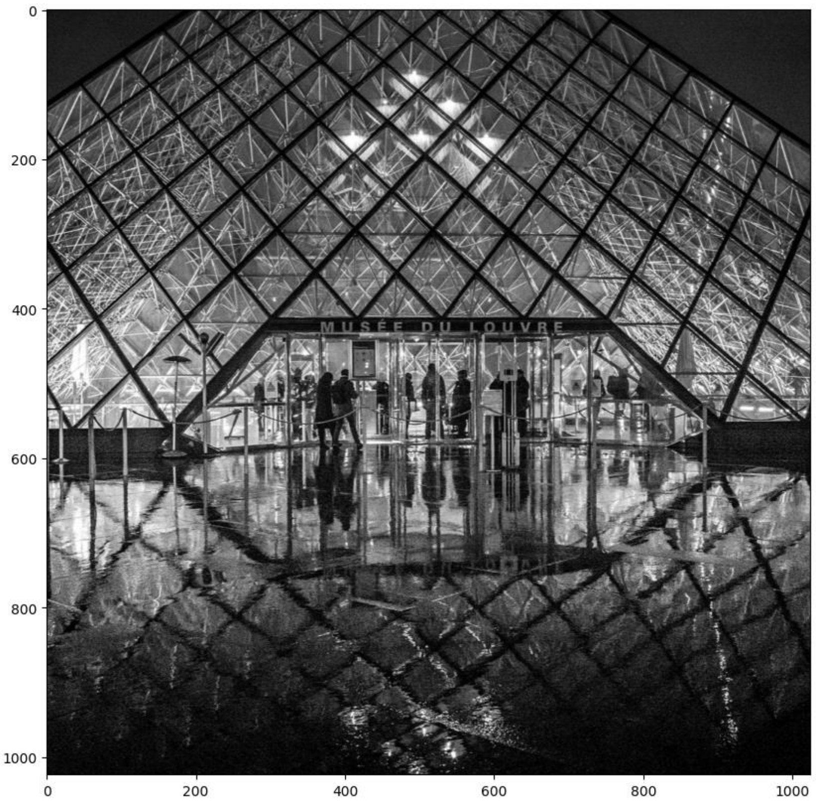
Ejemplo Numérico: Escalar $(5, 10)$ con $s_x = 2$, $s_y = 0.5$

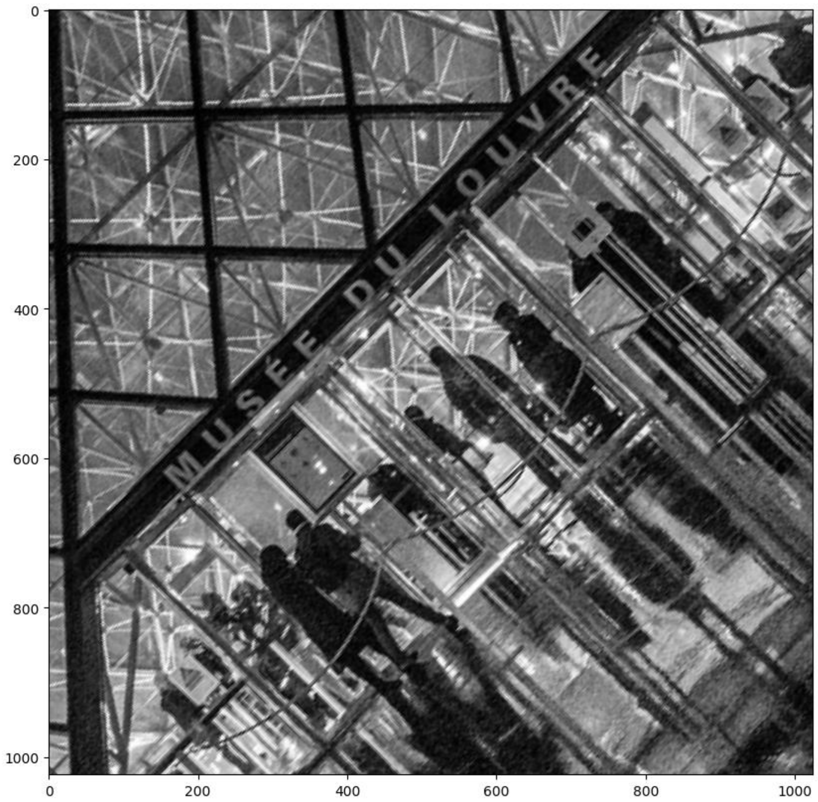
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 5 \\ 10 \\ 1 \end{bmatrix} = \begin{bmatrix} 10 \\ 5 \\ 1 \end{bmatrix}$$

Operación:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} x \cdot s_x \\ y \cdot s_y \\ 1 \end{bmatrix}$$

Escalado


MUSÉE DU LOUVRE
0
200
400
600
800
1000


MUSÉE DU LOUVRE
0
200
400
600
800
1000

Matriz de Traslación Homogénea

Concatenación Matricial:

- Concepto: La mayor ventaja de las coordenadas homogéneas es la capacidad de combinar múltiples transformaciones en una única matriz.
- Regla: Para aplicar una secuencia de transformaciones T_1, luego T_2, y finalmente T_3, la matriz compuesta se calcula multiplicando las matrices de transformación en orden inverso de aplicación: $\mathbf{M}_{Total} = \mathbf{M}_{T_3} \cdot \mathbf{M}_{T_2} \cdot \mathbf{M}_{T_1}$
- Propiedad Clave: La multiplicación de matrices no es conmutativa. El orden importa.
- Ejemplo: Rotar y luego trasladar **NO** es lo mismo que trasladar y luego rotar.

Ejemplo Compuesto: Rotar y Luego Trasladar

Problema: Rotar un punto P(1,0) por 90° antihorario alrededor del origen, y luego trasladarlo por (t_x, t_y)=(2,3).

Paso 1: Punto en homogéneas

$$\mathbf{P} = \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

Paso 2: Matriz de Rotación ($\mathbf{M}_R$)

$$\mathbf{M}_R = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Paso 3: Matriz de Traslación ($\mathbf{M}_T$)

$$\mathbf{M}_T = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix}$$

Paso 4: Matriz Compuesta ($\mathbf{M}_{Total} = \mathbf{M}_T \cdot \mathbf{M}_R$)

$$\mathbf{M}_{Total} = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Ejemplo Compuesto: Rotar y Luego Trasladar

Problema: Rotar un punto P(1,0) por 90° antihorario alrededor del origen, y luego trasladarlo por (t_x, t_y)=(2,3).

Paso 4: Matriz Compuesta ($\mathbf{M}_{Total} = \mathbf{M}_T \cdot \mathbf{M}_R$)

$$\mathbf{M}_{Total} = \begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} =$$

$$\begin{bmatrix} (1\cdot 0 + 0\cdot 1 + 2\cdot 0) & (1\cdot -1 + 0\cdot 0 + 2\cdot 0) & (1\cdot 0 + 0\cdot 0 + 2\cdot 1) \\ (0\cdot 0 + 1\cdot 1 + 3\cdot 0) & (0\cdot -1 + 1\cdot 0 + 3\cdot 0) & (0\cdot 0 + 1\cdot 0 + 3\cdot 1) \\ (0\cdot 0 + 0\cdot 1 + 1\cdot 0) & (0\cdot -1 + 0\cdot 0 + 1\cdot 0) & (0\cdot 0 + 0\cdot 0 + 1\cdot 1) \end{bmatrix}$$

$$\mathbf{M}_{Total} = \begin{bmatrix} 0 & -1 & 2 \\ 1 & 0 & 3 \\ 0 & 0 & 1 \end{bmatrix}$$

Paso 5: Aplicar la matriz total al punto.

$$\mathbf{P'} = \mathbf{M}_{Total} \cdot \mathbf{P} = \begin{bmatrix} 0 & -1 & 2 \\ 1 & 0 & 3 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 0\cdot 1 - 1\cdot 0 + 2\cdot 1 \\ 1\cdot 1 + 0\cdot 0 + 3\cdot 1 \\ 0\cdot 1 + 0\cdot 0 + 1\cdot 1 \end{bmatrix} = \begin{bmatrix} 2 \\ 4 \\ 1 \end{bmatrix}$$

Resultado: El punto final es $P'(2,4)$

Transformaciones Afines Generales

- Concepto: Las transformaciones afines son un tipo de transformación lineal que preserva las líneas paralelas y la relación de distancias en una línea.
- Incluye: Traslación, rotación, escalado (uniforme y no uniforme) y cizallamiento (shear).
- Matriz General (2D):

$$\mathbf{M}_{Afín} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

- Propiedad Clave: Una transformación afín puede definirse de forma única si conocemos las coordenadas de tres puntos no colineales en la imagen de entrada y sus correspondientes coordenadas en la imagen de salida.

Contenido:

El Problema del Re-muestreo

Después de aplicar una transformación espacial, las coordenadas calculadas (x' ,y') casi siempre serán valores de punto flotante (ej. 10.3, 25.7).
Una imagen digital solo puede almacenar información en ubicaciones de píxeles enteros (ej. (10,25) o (11,26)).
Se necesita una estrategia para determinar el valor de intensidad (color o gris) para cada píxel entero en la imagen de salida, a partir de los valores de los píxeles de la imagen de entrada.

El Método Más Simple: Vecino Más Cercano

- Para cada píxel en la imagen de salida (x' ,y'), se calcula su punto de origen (x,y)=T^{-1} (x' ,y') en la imagen de entrada.
- Luego, se redondean *x* e *y* a los enteros más cercanos (round(x), round(y)).
- El valor de intensidad del píxel de salida (x' ,y') se toma directamente del píxel más cercano en la imagen de entrada.

Interpolación

Ventajas:

- Rapidez: Computacionalmente muy eficiente.
- No introduce nuevos valores: Los valores de intensidad en la imagen de salida son siempre los mismos que en la entrada.

Desventajas:

- Artefactos: Produce un efecto de "escalón" (jaggies) en líneas diagonales y bordes curvos. La imagen parece "pixelada".
- Puede llevar a la pérdida de información si la imagen se escala hacia abajo.

Fórmula:

$$I_{salida}(x', y') = I_{entrada}(\mathrm{round}(x), \mathrm{round}(y))$$

Ejemplo: Interpolación por Vecino Más Cercano

Problema: Obtener el valor de intensidad para el píxel de salida en (5.7, 8.2) de una imagen original.

Paso 1: Identificar las coordenadas flotantes

$$(x_f, y_f) = (5.7, 8.2)$$

Paso 2: Redondear a los enteros más cercanos

- $\text{round}(5.7) = 6$
- $\text{round}(8.2) = 8$

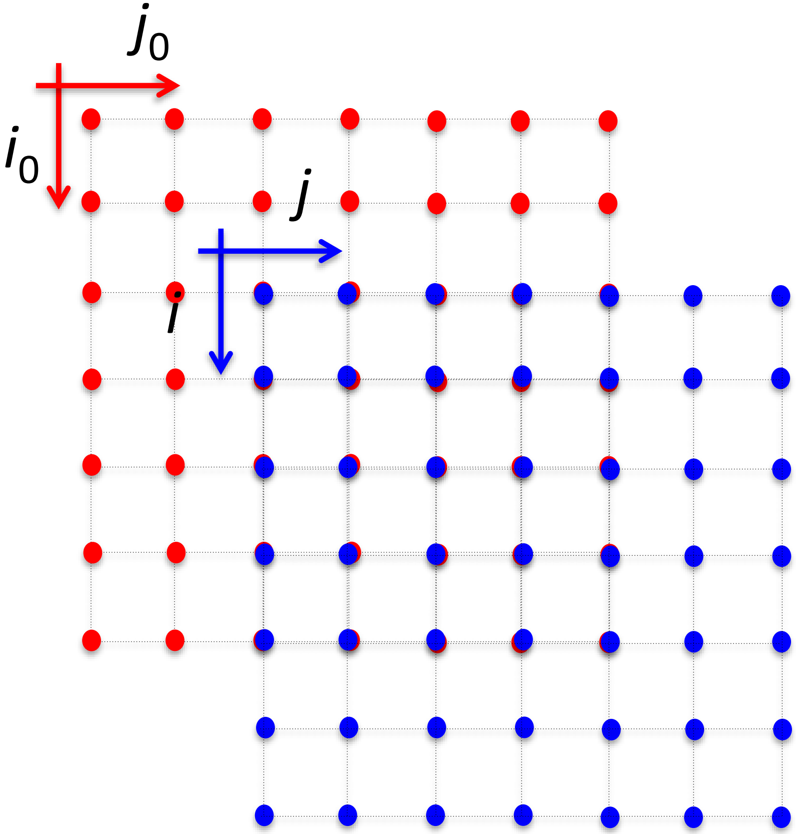
Paso 3: Tomar el valor del píxel en la imagen original

$$I_{salida}(5.7, 8.2) = I_{original}(6, 8)$$

Resultado: El valor de intensidad del píxel en $(5.7, 8.2)$ será igual al valor del píxel original en $(6, 8)$.

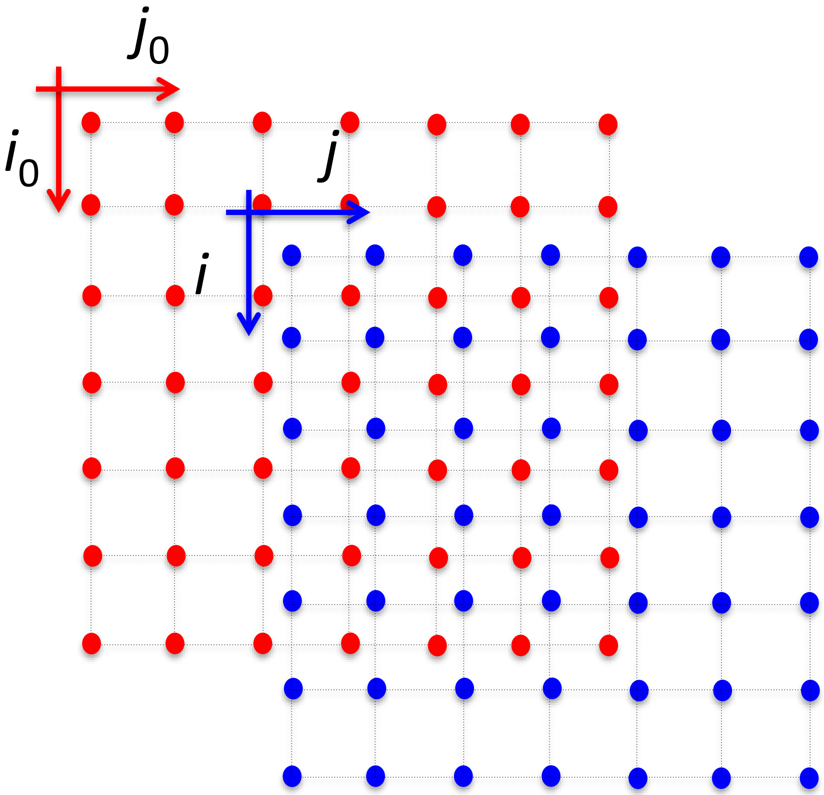
Interpolación

Interpolación (ideal)

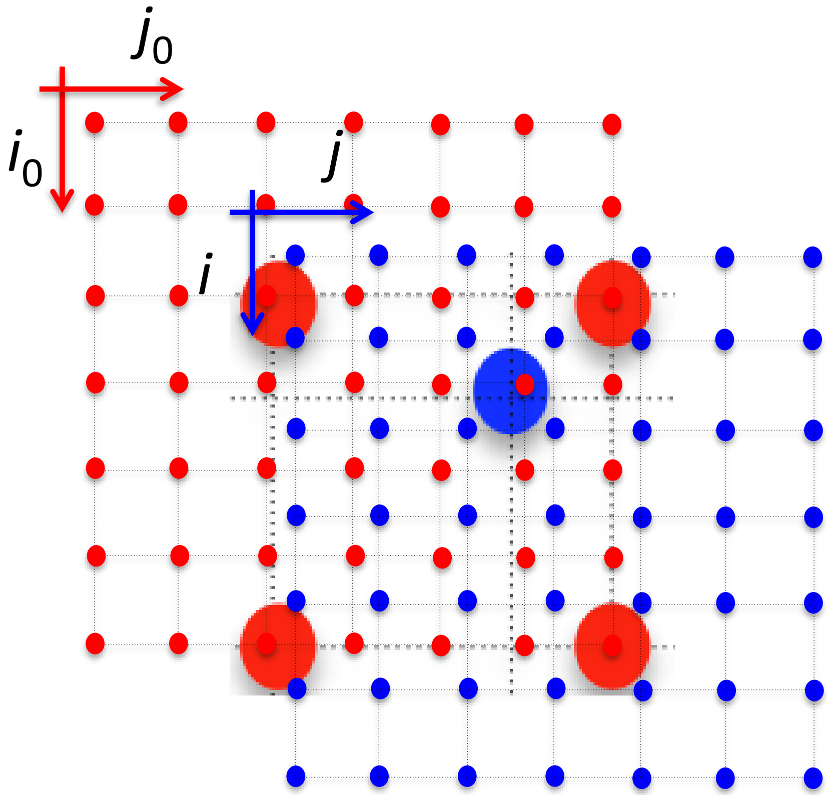


Interpolación

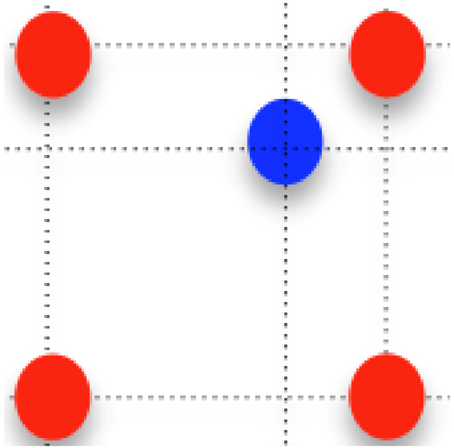
Interpolación (real)



Interpolación (técnica)



Interpolación (técnica)



Suavizado Intermedio: Bilineal

Para cada píxel de salida en (x' , y'), se calcula su origen (x, y) en la imagen de entrada.
En lugar de solo un píxel, se consideran los cuatro píxeles vecinos más cercanos al punto (x, y).
El valor de intensidad se calcula como un promedio ponderado de estos cuatro píxeles. Los pesos se basan en la distancia del punto (x, y) a cada uno de los vecinos.

Proceso (conceptual):

1. Interpolar linealmente en la dirección horizontal (eje X) entre los dos píxeles superiores.
2. Interpolar linealmente en la dirección horizontal (eje X) entre los dos píxeles inferiores.
3. Interpolar linealmente en la dirección vertical (eje Y) entre los dos resultados anteriores.

Ventajas:

- Resultados visualmente más suaves que el vecino más cercano.
- Reduce los artefactos de escalón.
- Buen equilibrio entre calidad y eficiencia.

Desventajas:

- Puede introducir cierto nivel de borrosidad, especialmente en los bordes.
- Más lento que el vecino más cercano.

Ejemplo Numérico - Interpolación Bilineal

Problema: Estimar el valor de intensidad del píxel en la coordenada original (2.3, 1.7) utilizando interpolación bilineal. Valores de píxeles vecinos:

- $I(2,1) = 100$
- $I(3,1) = 120$
- $I(2,2) = 110$
- $I(3,2) = 130$

Ejemplo Numérico - Interpolación Bilineal

Problema: Estimar el valor de intensidad del píxel en la coordenada original (2.3, 1.7) utilizando interpolación bilineal. Valores de píxeles vecinos:

- $I(2, 1) = 100$
- $I(3, 1) = 120$
- $I(2, 2) = 110$
- $I(3, 2) = 130$

Paso 1: Identificar la esquina superior-izquierda del bloque de píxeles y las fracciones.

- El punto de interés es $(x_{flotante}, y_{flotante}) = (2.3, 1.7)$.
- La coordenada entera de la **esquina superior-izquierda** del bloque de píxeles que contiene a $(2.3, 1.7)$ es $(\lfloor 2.3 \rfloor, \lfloor 1.7 \rfloor) = (2, 1)$.
- Usamos $(x_0, y_0) = (2, 1)$ como nuestra coordenada de referencia.
- Las fracciones (o "pesos") para la interpolación son:
  - $\Delta x = x_{flotante} - x_0 = 2.3 - 2 = 0.3$
  - $\Delta y = y_{flotante} - y_0 = 1.7 - 1 = 0.7$

Ejemplo Numérico - Interpolación Bilineal

Problema: Estimar el valor de intensidad del píxel en la coordenada original (2.3, 1.7) utilizando interpolación bilineal. Valores de píxeles vecinos:

- $I(2,1) = 100$
- $I(3,1) = 120$
- $I(2,2) = 110$
- $I(3,2) = 130$

Paso 2: Interpolar linealmente en la dirección x para la fila superior $\boxed{(y_0 = 1)}$.

- Esto se hace entre los píxeles $I(x_0, y_0) = I(2,1)$ y $I(x_0+1, y_0) = I(3,1)$.
- $R_{fila\_superior} = I(2,1) \cdot (1 - \Delta x) + I(3,1) \cdot \Delta x$
- $R_{fila\_superior} = 100 \cdot (1 - 0.3) + 120 \cdot (0.3)$
- $R_{fila\_superior} = 100 \cdot (0.7) + 120 \cdot (0.3) = 70 + 36 = 106$

Paso 3: Interpolar linealmente en la dirección x para la fila inferior $\boxed{(y_0 + 1 = 2)}$.

- Esto se hace entre los píxeles $I(x_0, y_0+1) = I(2,2)$ y $I(x_0+1, y_0+1) = I(3,2)$.
- $R_{fila\_inferior} = I(2,2) \cdot (1 - \Delta x) + I(3,2) \cdot \Delta x$
- $R_{fila\_inferior} = 110 \cdot (1 - 0.3) + 130 \cdot (0.3)$
- $R_{fila\_inferior} = 110 \cdot (0.7) + 130 \cdot (0.3) = 77 + 39 = 116$

x

y

Ejemplo Numérico - Interpolación Bilineal

Problema: Estimar el valor de intensidad del píxel en la coordenada original (2.3, 1.7) utilizando interpolación bilineal. Valores de píxeles vecinos:

- $I(2,1) = 100$
- $I(3,1) = 120$
- $I(2,2) = 110$
- $I(3,2) = 130$

Paso 4: Interpolar linealmente en la dirección y usando los resultados de las filas.

- Ahora interpolamos verticalmente entre $R_{fila\_superior}$ y $R_{fila\_inferior}$ usando Δy.
- $I_{interpolado}(2.3, 1.7) = R_{fila\_superior} \cdot (1 - \Delta y) + R_{fila\_inferior} \cdot \Delta y$
- $I_{interpolado}(2.3, 1.7) = 106 \cdot (1 - 0.7) + 116 \cdot (0.7)$
- $I_{interpolado}(2.3, 1.7) = 106 \cdot (0.3) + 116 \cdot (0.7) = 31.8 + 81.2 = 113$

Resultado: El valor de intensidad interpolado para la coordenada $(2.3, 1.7)$ es $\mathbf{113}$.

Ejemplo Numérico - Interpolación Bilineal

Problema: Se tiene una imagen y se necesita encontrar el valor de un píxel en la posición flotante (X=5.8, Y=3.2) utilizando interpolación bilineal. Valores de píxeles vecinos (niveles de gris) :

- Píxel superior-izquierda: $I(5,3) = 200$
- Píxel superior-derecha: $I(6,3) = 210$
- Píxel inferior-izquierda: $I(5,4) = 180$
- Píxel inferior-derecha: $I(6,4) = 190$

Ejemplo Numérico - Interpolación Bilineal

Problema: Se tiene una imagen y se necesita encontrar el valor de un píxel en la posición flotante (X=5.8, Y=3.2) utilizando interpolación bilineal. Valores de píxeles vecinos (niveles de gris) :

Paso 1: Identificar la Esquina Superior-Izquierda de Referencia y las Fracciones

Primero, encontramos las coordenadas enteras del píxel en la **esquina superior-izquierda** del cuadrado que contiene nuestro punto flotante $(5.8, 3.2)$.

- Coordenada X de referencia (x_0): $\lfloor 5.8 \rfloor = 5$
- Coordenada Y de referencia (y_0): $\lfloor 3.2 \rfloor = 3$

Entonces, nuestro píxel de referencia es $(x_0, y_0) = (5, 3)$.

Ahora, calculamos las **fracciones** que nos dicen qué tan lejos está nuestro punto de las coordenadas enteras en cada dirección:

- Fracción X (Δx): $x_{flotante} - x_0 = 5.8 - 5 = 0.8$
- Fracción Y (Δy): $y_{flotante} - y_0 = 3.2 - 3 = 0.2$

Ejemplo Numérico - Interpolación Bilineal

Problema: Se tiene una imagen y se necesita encontrar el valor de un píxel en la posición flotante (X=5.8, Y=3.2) utilizando interpolación bilineal. Valores de píxeles vecinos (niveles de gris) :

Paso 2: Interpolar Linealmente en la Fila Superior (Horizontalmente)

Usamos los valores de los píxeles en la fila superior (donde $Y = 3$): $I(5,3) = 200$ y $I(6,3) = 210$.

Aplicamos la interpolación lineal con $\Delta x = 0.8$:

$$R_{fila\_superior} = I(5,3) \cdot (1 - \Delta x) + I(6,3) \cdot \Delta x$$
$$R_{fila\_superior} = 200 \cdot (1 - 0.8) + 210 \cdot (0.8)$$
$$R_{fila\_superior} = 200 \cdot (0.2) + 210 \cdot (0.8)$$
$$R_{fila\_superior} = 40 + 168 = \mathbf{208}$$

Ejemplo Numérico - Interpolación Bilineal

Problema: Se tiene una imagen y se necesita encontrar el valor de un píxel en la posición flotante (X=5.8, Y=3.2) utilizando interpolación bilineal. Valores de píxeles vecinos (niveles de gris) :

Paso 3: Interpolar Linealmente en la Fila Inferior (Horizontalmente)

Ahora hacemos lo mismo para la fila inferior (donde $Y = 4$): $I(5,4) = 180$ y $I(6,4) = 190$.

Aplicamos la interpolación lineal también con $\Delta x = 0.8$:

$$R_{fila\_inferior} = I(5,4) \cdot (1 - \Delta x) + I(6,4) \cdot \Delta x$$
$$R_{fila\_inferior} = 180 \cdot (1 - 0.8) + 190 \cdot (0.8)$$
$$R_{fila\_inferior} = 180 \cdot (0.2) + 190 \cdot (0.8)$$
$$R_{fila\_inferior} = 36 + 152 = \mathbf{188}$$

Ejemplo Numérico - Interpolación Bilineal

Problema: Se tiene una imagen y se necesita encontrar el valor de un píxel en la posición flotante (X=5.8, Y=3.2) utilizando interpolación bilineal. Valores de píxeles vecinos (niveles de gris) :

Paso 4: Interpolar Linealmente Verticalmente

Finalmente, interpolamos entre los dos resultados que obtuvimos de las filas: $R_{fila\_superior} = 208$ y $R_{fila\_inferior} = 188$.

Usamos nuestra fracción $\Delta y = 0.2$:

$$I_{interpolado}(5.8, 3.2) = R_{fila\_superior} \cdot (1 - \Delta y) + R_{fila\_inferior} \cdot \Delta y$$
$$I_{interpolado}(5.8, 3.2) = 208 \cdot (1 - 0.2) + 188 \cdot (0.2)$$
$$I_{interpolado}(5.8, 3.2) = 208 \cdot (0.8) + 188 \cdot (0.2)$$
$$I_{interpolado}(5.8, 3.2) = 166.4 + 37.6 = \mathbf{204}$$

Resultado Final:

El valor de intensidad interpolado para la coordenada $(5.8, 3.2)$ en la imagen es **204**.

Interpolación Bicúbica

Considera los dieciséis píxeles vecinos (4×4 ventana) al punto (x,y) en la imagen de entrada.
Utiliza una función de interpolación cúbica (como un spline cúbico) para estimar los pesos. Esto permite una estimación de la intensidad que no solo considera los valores de los píxeles, sino también las pendientes y curvaturas de la superficie de intensidad.

Ventajas:

- Produce los resultados más suaves y de mayor calidad visual.
- Mantiene mejor la nitidez de los bordes que la interpolación bilineal.
- Minimiza el aliasing de forma efectiva.

Desventajas:

- Alta complejidad computacional: Mucho más lenta que los otros métodos.
- Puede introducir ligeros sobrepasos (overshoots) o artefactos de "anillo" alrededor de cambios de intensidad muy abruptos.

Contenido:

El Fenómeno del Aliasing

Se produce cuando la frecuencia de muestreo de una señal es insuficiente para capturar sus componentes de alta frecuencia. En imágenes, las altas frecuencias corresponden a cambios rápidos en intensidad (bordes afilados, texturas finas, patrones repetitivos).
Cuando se reduce el tamaño de una imagen o se rota, si el nuevo muestreo no es lo suficientemente denso, estas altas frecuencias se interpretan incorrectamente.

Consecuencias en PDI

- **Jaggies (dientes de sierra):** Líneas diagonales y curvas aparecen escalonadas.
- **Patrones Moiré:** Patrones de interferencia extraños y no deseados en texturas repetitivas (ej. un patrón de rejilla se convierte en ondas).
- **Pérdida de Detalles:** Información visual de alta frecuencia se pierde o se "alias".

¿Por Qué Ocurre el Aliasing?

- **Submuestreo (Downsampling):** Al reducir el tamaño de una imagen, se descartan píxeles, lo que puede llevar a una pérdida de información de alta frecuencia.
- **Rotación de Líneas Finas/Diagonales:** Una línea perfectamente delgada puede desaparecer o aparecer discontinua cuando se rota y sus píxeles no caen en la cuadrícula de salida.
- **Interpolación del Vecino Más Cercano:** Este método no realiza ningún tipo de promedio o suavizado, lo que lo hace muy propenso al aliasing.
- **Distorsión de Perspectiva:** Los objetos lejanos pueden comprimirse en el espacio de la imagen, aumentando la frecuencia espacial percibida y creando aliasing.

Suavizando el Anti-aliasing

- **Concepto:** Son técnicas diseñadas para mitigar los efectos visuales no deseados del aliasing.

- **Principio:** Antes o durante el muestreo, se aplica un filtro de paso bajo para eliminar o atenuar las altas frecuencias que podrían causar aliasing.

  - ➢ Esto convierte las transiciones abruptas (bordes afilados) en transiciones más suaves (bordes difusos), lo que la cuadrícula de muestreo de salida puede representar mejor.

- **Compromiso:** El anti-aliasing siempre implica un compromiso entre la nitidez de la imagen y la reducción de artefactos. Un exceso de anti-aliasing puede llevar a una imagen borrosa.

Estrategias Comunes de Anti-aliasing

Pre-filtrado (Pre-filtering):

- Aplicar un filtro de suavizado (ej. filtro Gaussiano) a la imagen de entrada antes de la transformación.
- Suaviza los bordes y las texturas finas, reduciendo las altas frecuencias que podrían "aliarse".
- Desventaja: Puede causar una pérdida de nitidez general en toda la imagen.

Sobremuestreo (Supersampling) y Downsampling:

- Realizar la transformación a una resolución significativamente más alta que la deseada.
- Luego, promediar los valores de los múltiples sub-píxeles resultantes para calcular el valor de cada píxel final en la resolución objetivo.
- Ventaja: Resultados de muy alta calidad.
- Desventaja: Computacionalmente muy intensivo.

Interpolación Avanzada:

- Métodos como la interpolación bilineal y bicúbica actúan como formas intrínsecas de anti-aliasing.
- Al promediar los valores de los píxeles vecinos, suavizan las transiciones y atenúan las altas frecuencias.

Estrategias Comunes de Anti-aliasing

Pre-filtrado (Pre-filtering):

- Aplicar un filtro de suavizado (ej. filtro Gaussiano) a la imagen de entrada antes de la transformación.
- Suaviza los bordes y las texturas finas, reduciendo las altas frecuencias que podrían "aliarse".
- Desventaja: Puede causar una pérdida de nitidez general en toda la imagen.

Sobremuestreo (Supersampling) y Downsampling:

- Realizar la transformación a una resolución significativamente más alta que la deseada.
- Luego, promediar los valores de los múltiples sub-píxeles resultantes para calcular el valor de cada píxel final en la resolución objetivo.
- Ventaja: Resultados de muy alta calidad.
- Desventaja: Computacionalmente muy intensivo.

Interpolación Avanzada:

- Métodos como la interpolación bilineal y bicúbica actúan como formas intrínsecas de anti-aliasing.
- Al promediar los valores de los píxeles vecinos, suavizan las transiciones y atenúan las altas frecuencias.

Aliasing vs. Anti-aliasing (La Diferencia Visual: Antes y Después)

Capítulo III:
FIN

Procesamiento Digital de Imágenes