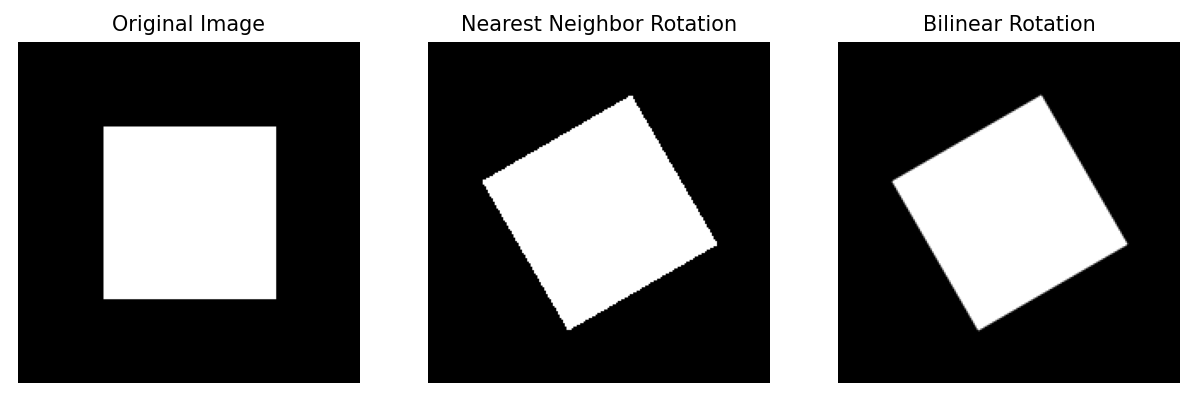
**Comment**

最近鄰插值（Nearest Neighbor Interpolation）由於僅根據最接近的像素值進行映射，不進行任何平滑處理，因此旋轉後的邊緣可能會出現明顯的**鋸齒效應（aliasing）或像素塊狀化（pixelation）**，導致邊界呈現不規則的突起。而**雙線性插值（Bilinear Interpolation）**則透過對鄰近的四個像素進行加權平均計算，使得輸出影像的過渡更加平滑，從而減少鋸齒現象，使旋轉後的結果更接近於理論上的連續變換效果，視覺上更自然且細膩。

**Photo**



**Program**

import cv2

import numpy as np

import matplotlib.pyplot as plt

size = 200

image = np.zeros((size, size), dtype=np.uint8)

cv2.rectangle(image, (50, 50), (150, 150), 255, -1)

angle = 30

center = (size // 2, size // 2)

M = cv2.getRotationMatrix2D(center, angle, 1.0)

# nearest neighbor interpolation

rotated\_nearest = cv2.warpAffine(image, M, (size, size), flags=cv2.INTER\_NEAREST)

# bilinear interpolation

rotated\_bilinear = cv2.warpAffine(image, M, (size, size), flags=cv2.INTER\_LINEAR)

cv2.imwrite("original.png", image)

cv2.imwrite("rotated\_nearest.png", rotated\_nearest)

cv2.imwrite("rotated\_bilinear.png", rotated\_bilinear)

# 顯示結果

fig, axs = plt.subplots(1, 3, figsize=(12, 4))

axs[0].imshow(image, cmap='gray')

axs[0].set\_title("Original Image")

axs[1].imshow(rotated\_nearest, cmap='gray')

axs[1].set\_title("Nearest Neighbor Rotation")

axs[2].imshow(rotated\_bilinear, cmap='gray')

axs[2].set\_title("Bilinear Rotation")

for ax in axs:

    ax.axis('off')

plt.show()