


(1) Assignment statement

Homework 5 (Due: 4/4)

(1) Create an image consisting of a white square with a black background, e.g.,

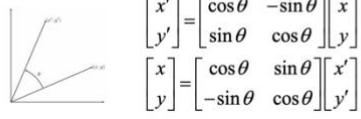


(2) Rotate the image by 30 degrees. Use (a) rotation with neighbor interpolation, and (b) rotation with bilinear interpolation.

(3) Compare the two results.

2-0

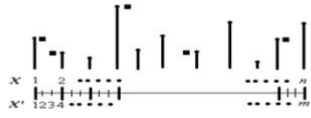
• Image Rotation



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

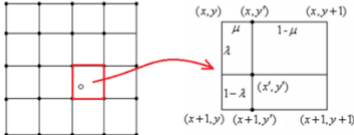
$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix} \begin{bmatrix} x' \\ y' \end{bmatrix}$$

• Nearest Neighbor Interpolation



2-1

• Bilinear Interpolation



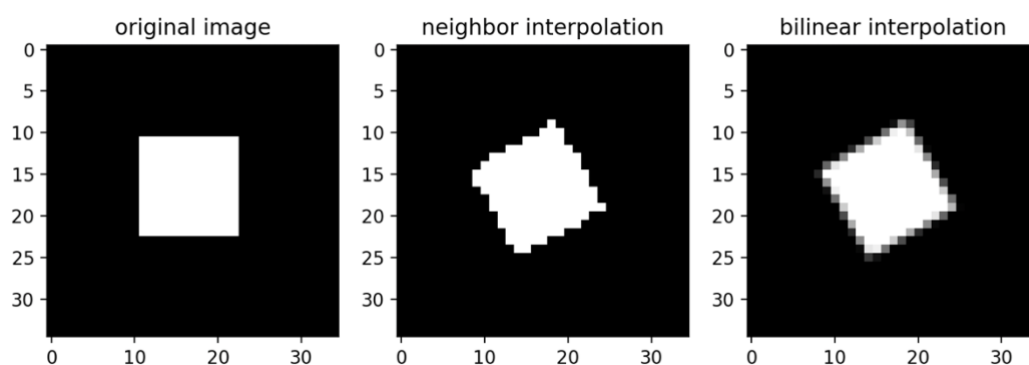
2-2

(2)

(a) Input/output images

Input: 此份作業程式碼沒有 user input。

Output:



(b) Source code

```
hw05.py > ...
1  import sys
2  import numpy as np
3  import matplotlib.pyplot as plt
4  import matplotlib.image as mpimg
5  import skimage.io as io
6  import skimage.transform as trans
7  side = 35
8  img = np.zeros([side, side])
9
10 for y in range(side//3, side*2//3):
11     for x in range(side//3, side*2//3):
12         img[y][x] = 255
13
14 neighbor_interp = trans.rotate(img, 30, order=0)
15 bilinear_interp = trans.rotate(img, 30, order=1)
16
17 plt.subplot(1, 3, 1)
18 plt.imshow(img, cmap='gray')
19 plt.title("original image")
20
21 plt.subplot(1, 3, 2)
22 plt.imshow(neighbor_interp, cmap='gray')
23 plt.title("neighbor interpolation")
24
25 plt.subplot(1, 3, 3)
26 plt.imshow(bilinear_interp, cmap='gray')
27 plt.title("bilinear interpolation")
28
29 plt.show()
```

(c) Comments

這次作業首先創造一個 35*35 的黑白圖片（黑底、正中央有約 1/9 的區域是白色正方形），再使用 scikit-image 的 `transform.rotate()` 函式來旋轉圖片，`order=0` 代表使用 `neighbor interpolation`，`order=1` 則是表示使用 `bilinear interpolation` 旋轉圖片（下圖中包含 `order` 的值分別代表的含義，來源自 scikit-image 官方 document）。

儘管目前已有現成函式庫可以用來處理圖像，就算不知道原理也能順利處理圖片，並且使用上相當簡潔方便，但若是能了解這些轉換背後的含義及原理，使用別人寫好的函式時心裡會比較踏實，因為知道這個函式實際做了什麼，另外，可能會遇到某些情況下需要自行撰寫處理、旋轉圖片的程式，這時就必須了解圖片旋轉投射的原理。

order : int, optional
The order of interpolation. The order has to be in the range 0-5:

- 0: Nearest-neighbor
- 1: Bi-linear (default)
- 2: Bi-quadratic
- 3: Bi-cubic
- 4: Bi-quartic
- 5: Bi-quintic

Default is 0 if image.dtype is bool and 1 otherwise.