作業一 影像全景圖

學號:7105056122  
姓名:陳奕翔

bilinear interpolation 和 spatial transforms C#

點擊顯示座標 和 計算spatial transforms矩陣 python

left image



right image



點擊圖顯示座標

import os, cv2, numpy

import sys

def l\_callbackFunc(e,x,y,f,p):

if e==cv2.EVENT\_LBUTTONDOWN:

print(x,y)

point["left"][0] = x

point["left"][1] = y

cv2.namedWindow("Image")

cv2.setMouseCallback("Image", l\_callbackFunc, None)

point = {"left": [-1,-1]}

img = cv2.imread("C:/Users/USER/Desktop/q2.jpg")

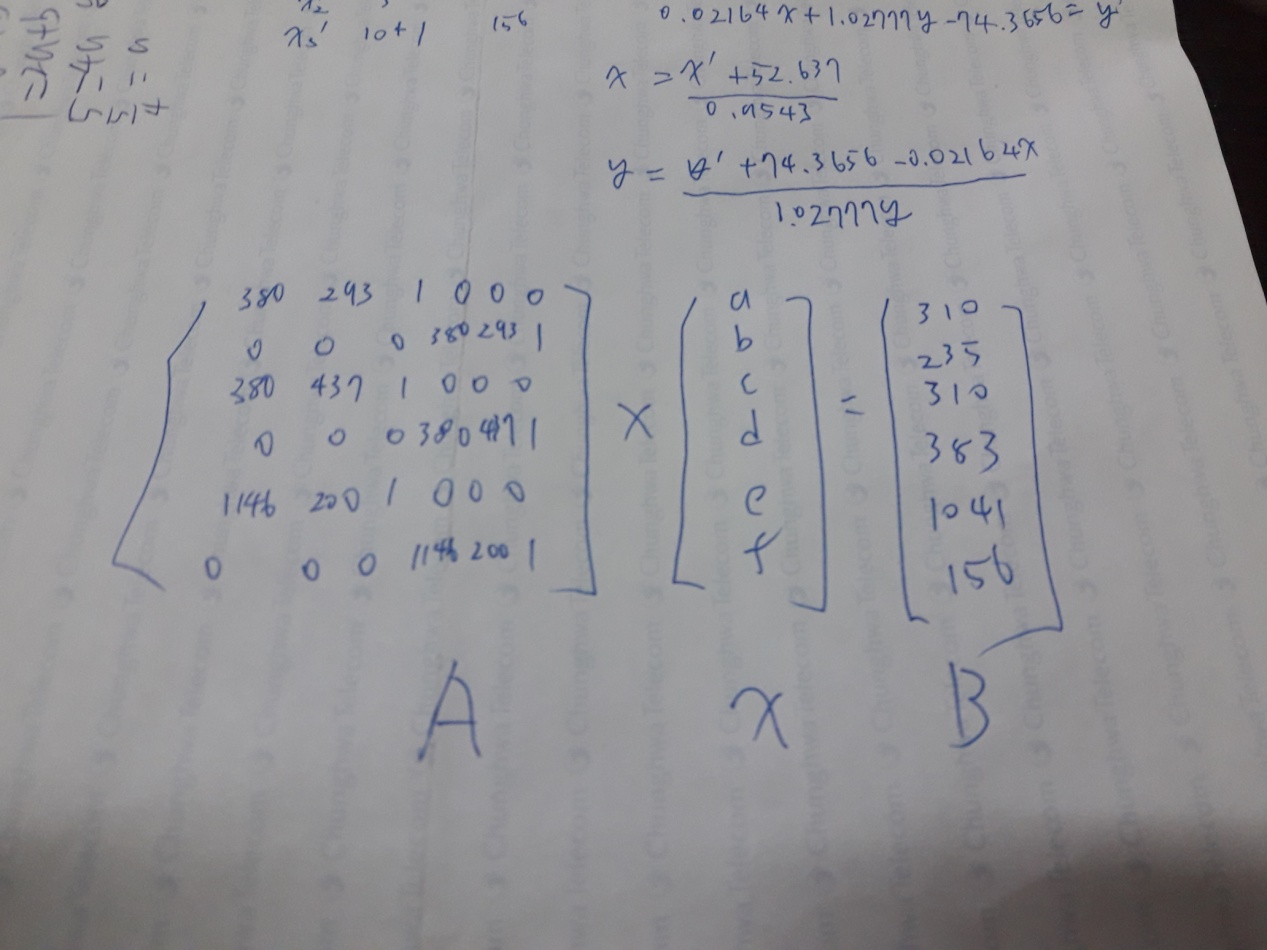
cv2.imshow("Image", img)

cv2.waitKey (0)

cv2.destroyAllWindows()

-----------------------------------------------------------------------------------

計算spatial transforms矩陣



left point1 (380,293) right point1 (310,235)

left point2 (380,437) right point2 (310,383)

left point3 (1146,200) right point3 (1041,156)

import numpy as np

matrix=np.array([[380,293,1,0,0,0],

[0,0,0,380,293,1],

[380,437,1,0,0,0],

[0,0,0,380,437,1],

[1146,200,1,0,0,0],

[0,0,0,1146,200,1],],dtype=float)

b=np.array([[310],

[235],

[310],

[383],

[1041],

[156],],dtype=float)a=np.linalg.inv(matrix)

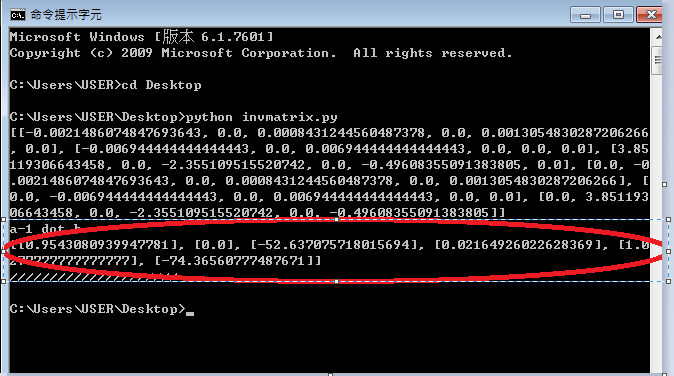
print (a.tolist())

print ("a-1 dot b")

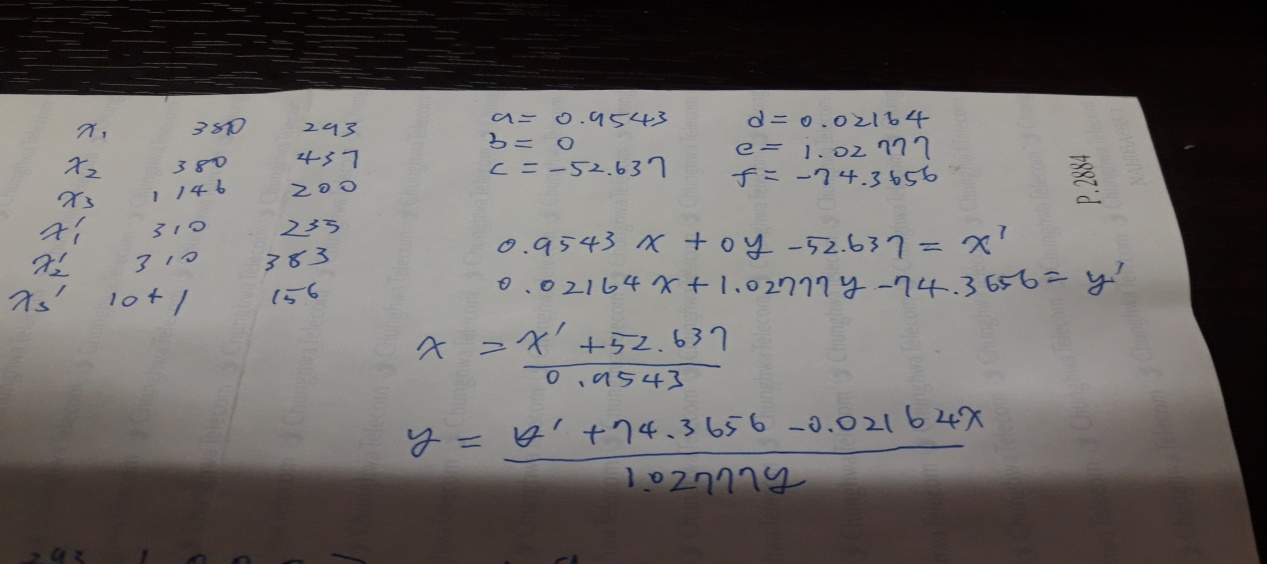
print(np.matmul(a,b).tolist())

-----------------------------------------------------------------------------------------

a~f



bilinear interpolation C#

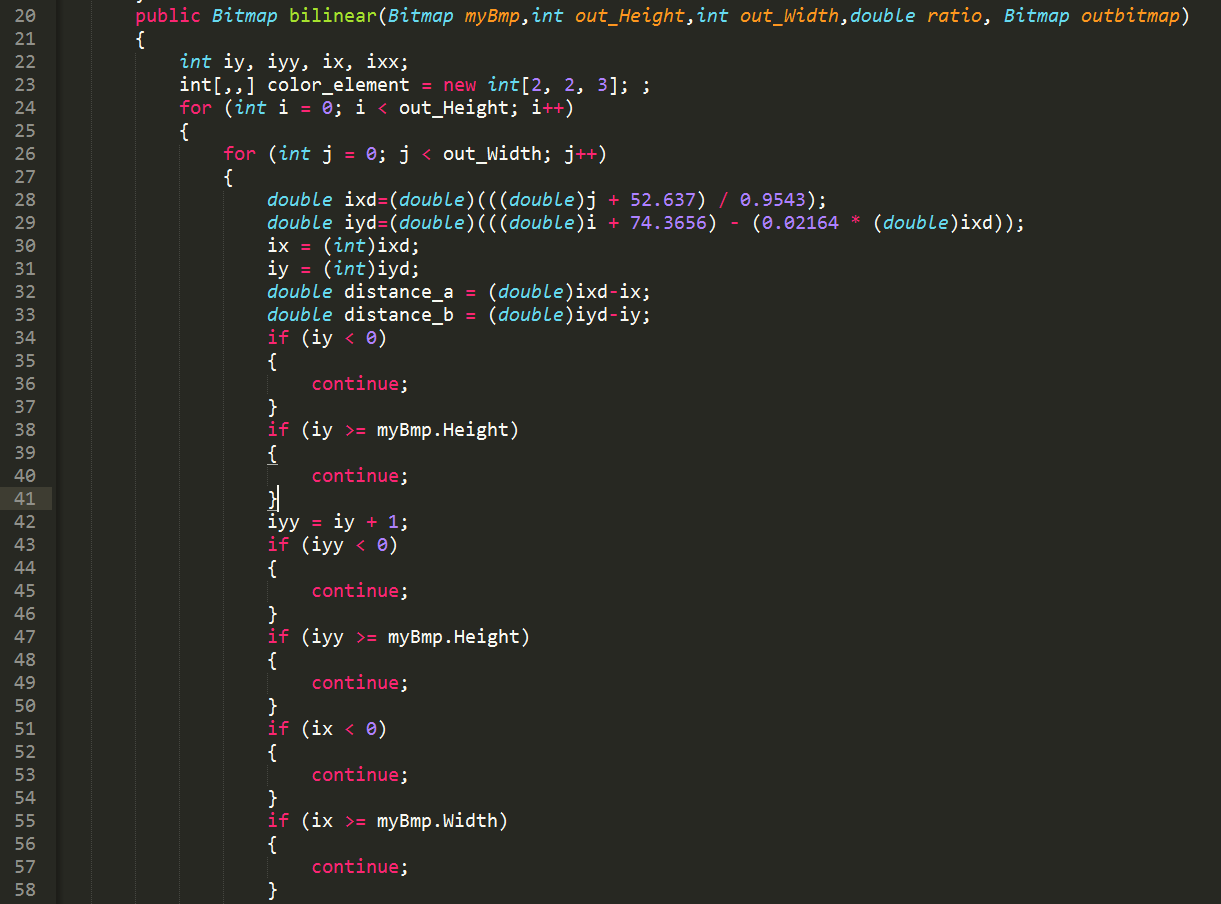


28,29行如上公式，j,i為input point ixd,iyd為out point

跑out loop對應到input point

ix ixx iy iyy為input point

計算distance=>ixd-ix,iyd-iy



68~83 提取4點的pixel

84~95 bilinear interpolation 公式

109 setpixel

