**Computer Vision HW3 Report**

Student ID: R12921107

Name: 王冠傑

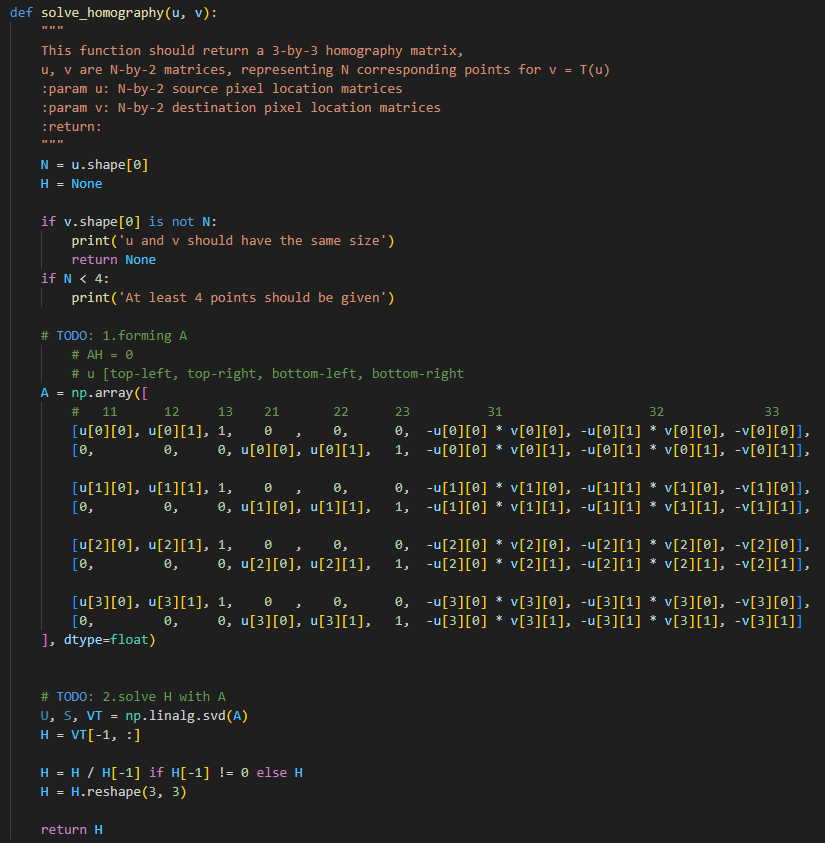
**Part 1.**

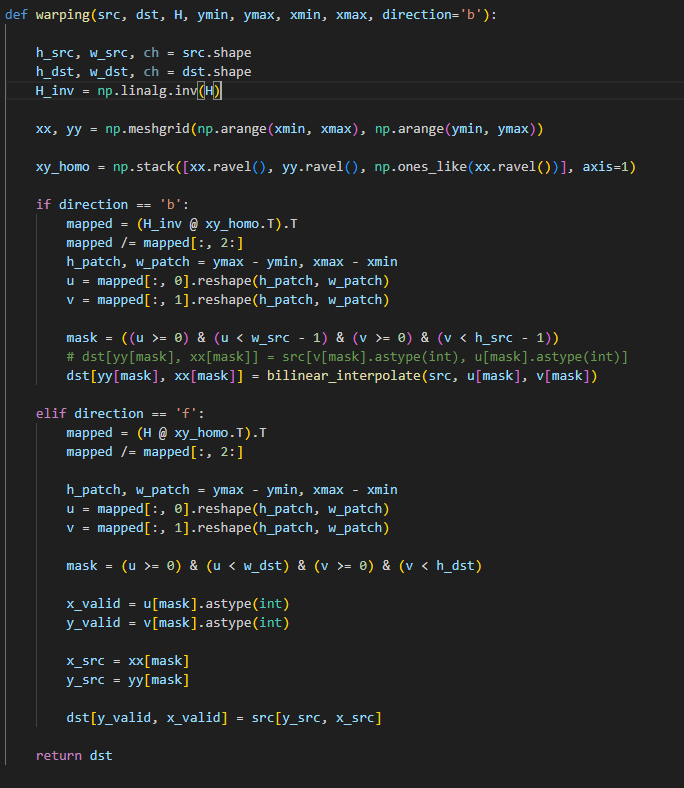
**• Paste your warped canvas**

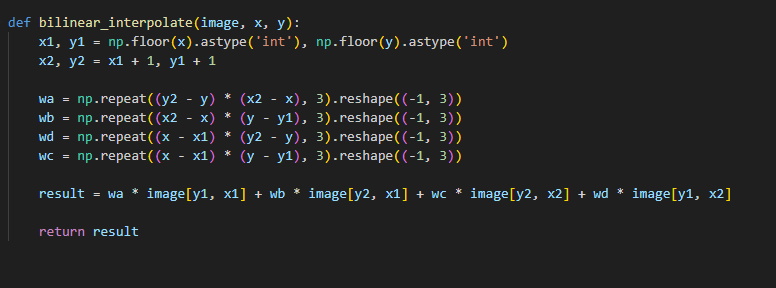


**Part 2.**

**• Paste the function code *solve\_homography(u, v)* & *warping( )* (both forward & backward)**

****

****

****

**• Briefly introduce the interpolation method you use**

**這裡使用bilinear interpolation計算，畫質相對nearest neighbor(直接取int)更加smooth，有嘗試使用bicubic算法，但效果與bilinear 差距不遠。**

**Part 3.**

**• Paste the 2 warped images and the link you find**



**output3\_1.png output3\_2.png**

**• Discuss the difference between 2 source images, are the warped results the same or different?**

**BL\_secret1相對於BL\_secret2是平整的照片，而BL\_secret2有很明顯地扭曲變形，推測是使用廣角鏡頭進行攝影的效果。造成原圖已經不是單純的affine transform，多增加了些彎折現象。兩張圖片都得到**[**http://media.ee.ntu.edu.tw/courses/cv/25S/**](http://media.ee.ntu.edu.tw/courses/cv/25S/) **的連結。**

**• If the results are the same, explain why. If the results are different, explain why?**

**output3\_2.png 的鋸齒程度比output3\_1.png還嚴重許多，推測原因如上題提及。但透過Homography 轉換後依然得到相同的QRcode連結。**

**Part 4.**

**• Paste your stitched panorama**



**• Can all consecutive images be stitched into a panorama?**

**沒有，因匹配特徵點只有在圖片部分區域。像是在路燈的部分就有被切割，建築物業右邊的場景被切掉。**

**• If yes, explain your reason. If not, explain under what conditions will result in a failure?**

**主要是篩選到的Homography feature 位置，以及圖片的重疊程度，通常越靠近邊緣的圖片會變形的越嚴重，且我們在warping也有做clip的手段，所以在經過Homography後投射在template外部的部分都會被捨棄掉。另外像是圖片中那棟建築物就發生拉伸變形。**

**Reference:**

[**https://pyimagesearch.com/2020/12/21/detecting-aruco-markers-with-opencv-and-python/**](https://pyimagesearch.com/2020/12/21/detecting-aruco-markers-with-opencv-and-python/)

[**https://blog.csdn.net/weixin\_43229348/article/details/120565635**](https://blog.csdn.net/weixin_43229348/article/details/120565635)

[**https://github.com/bettyteng21/1122-CV/tree/master**](https://github.com/bettyteng21/1122-CV/tree/master)

[**GitHub - Louislar/NTU\_CV\_HW**](https://github.com/Louislar/NTU_CV_HW)

[**Image-Stitching-OpenCV/Image\_Stitching.py at new · linrl3/Image-Stitching-OpenCV · GitHub**](https://github.com/linrl3/Image-Stitching-OpenCV/blob/new/Image_Stitching.py)