Computer Vision – Programming Project 4

王順興　0210814

Introduction

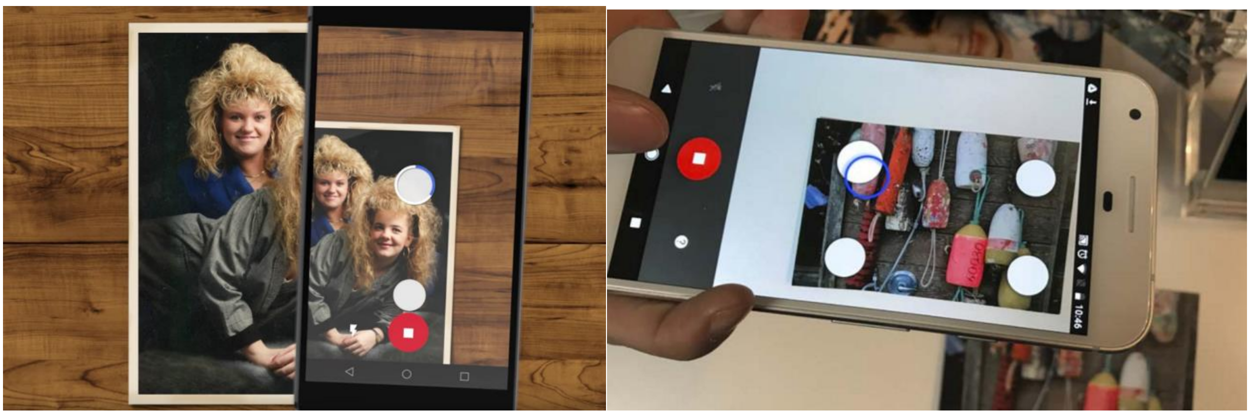
In this project, I will work on Image Stitching and, used a method that I've thought of, to address the issue of brightness difference between stitched images. Image stitching is quite simple, so I'd spent more time dealing with the latter.

i. The Dataset

All of the samples are photos taken by me.

ii. Motivation

Image Stitching has been quite mature, but recently the most impressive (in quality and speed) application of such is published by Google. The Google PhotoScan can produce a high quality image of a subject by taking multiple photos of different parts of the subject:

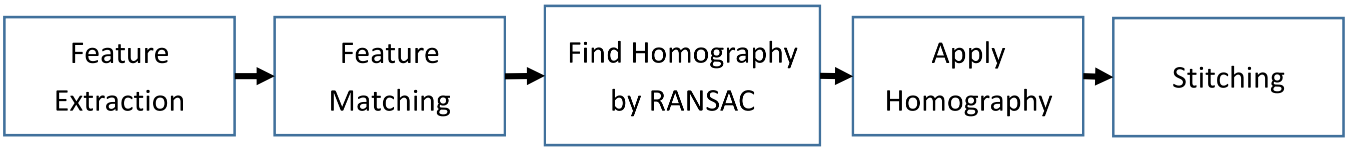


So, I wish to discover the problems that they might have encountered when developing the app, and address it my way. I will most likely fail, but as Thomas Edison once said:

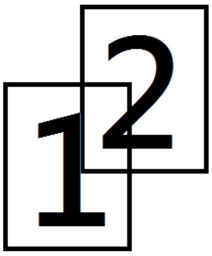
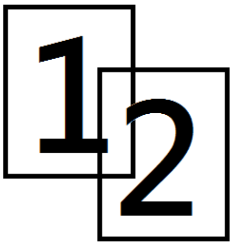
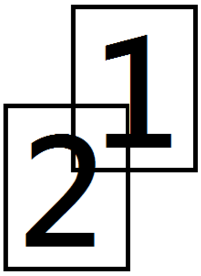
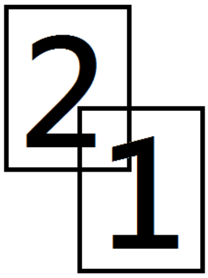
"I have not failed. I've just found 10,000 ways that won't work."

Method and Implementation

For **Image Stitching**, the method is almost identical to the last project, so here I explain it very briefly.



For Stitching, I simply put one image on top of the other. The tedious part is to deal with four configurations of the images:

				
X-axis	Pivots image 1	Pivots image 1	Pivots image 2	Pivots image 2
Y-axis	Pivots image 2	Pivots image 1	Pivots image 1	Pivots image 2

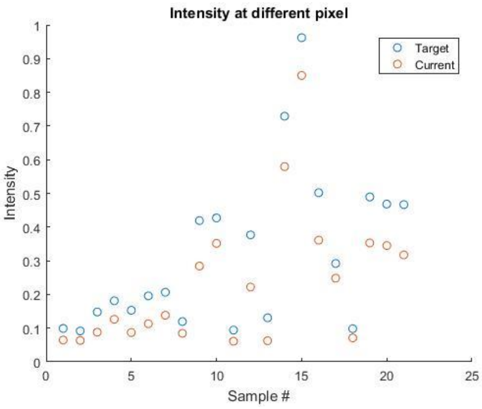
After the stitching you'll see something like this:

which looks horrible; not just because there is no blending, but also because the huge difference in brightness between the two images. This is why I decided to deal with it.

For **Brightness Difference between two images**, from my limited knowledge of Image Processing, I do not know of a way to fix it. So I came up with my own.

The concept is quite simple: We need to know the Homography to project the images to the same plane. Knowing Homography requires inliers, which is found at the RANSAC step. Inliers are the matched points between two images; and in a stitched image, the matched points should have the same brightness.

So by matching the brightness of the Inliers, we should be able to match the brightness between two images. Using inliers to do the job kept the additional computation at minimum.



← The difference of brightness between two images at matched points.

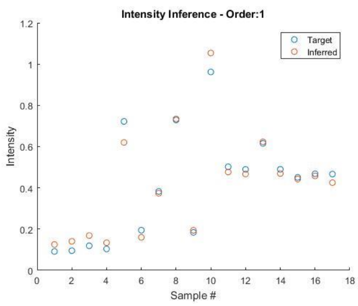
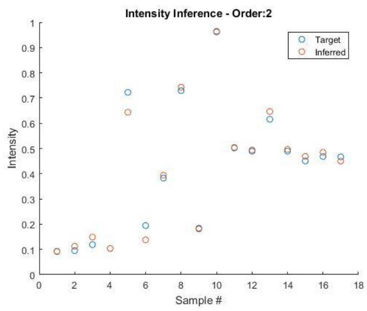
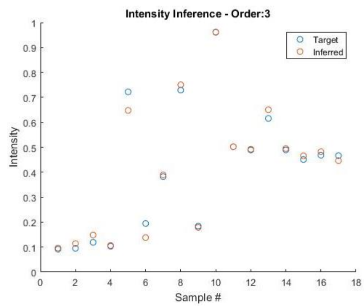
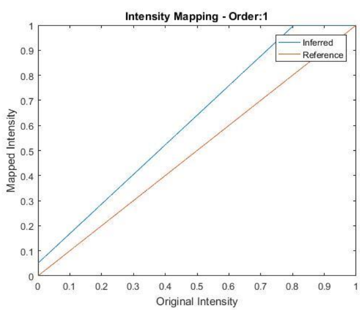
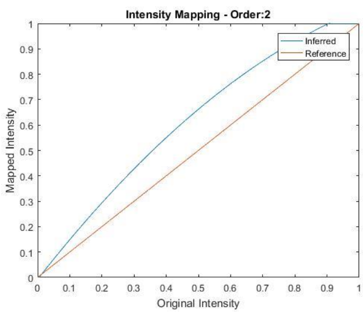
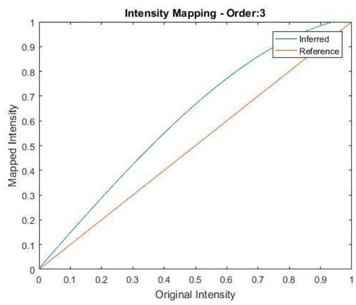
The brightness of an image is the V value in HSV, which is normalized to [0 1].

To find the equation to relate one image's brightness to another, polynomial regression is used:

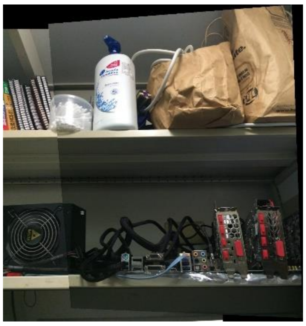
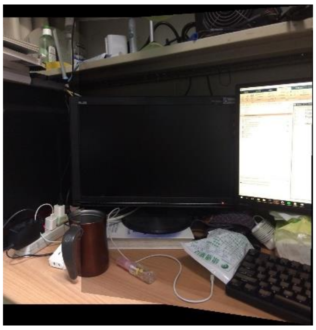
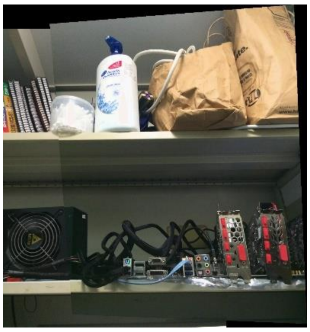
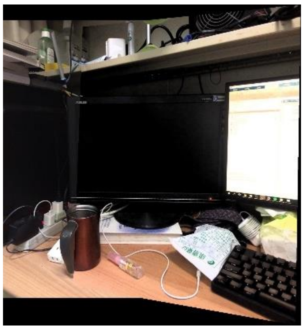
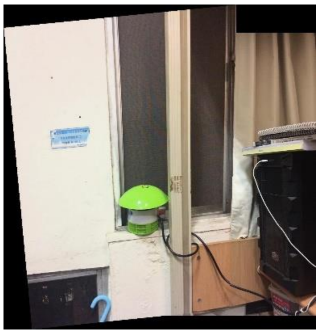
$$y_i = a_0 + a_1 x_i + a_2 x_i^2 + \cdots + a_m x_i^m + \varepsilon_i \ (i = 1, 2, \ldots, n)$$

where y_i is the Target brightness, and x_i is the Current brightness.

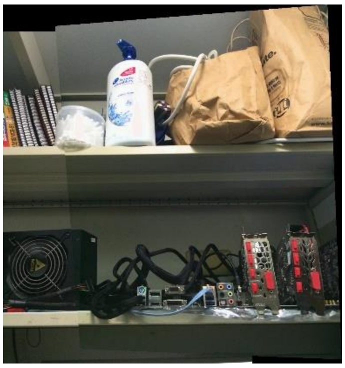
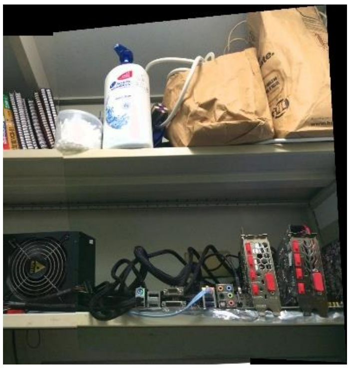
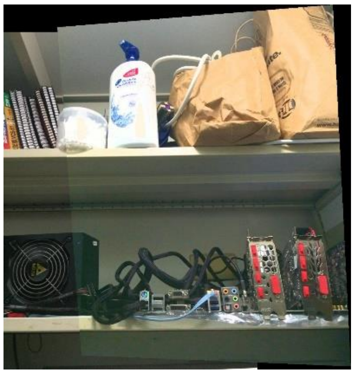
I implemented polynomial regression of order 1～3:

Order	1	2	3
Target Inference	 	 	
Intensity Mapping	 	 	

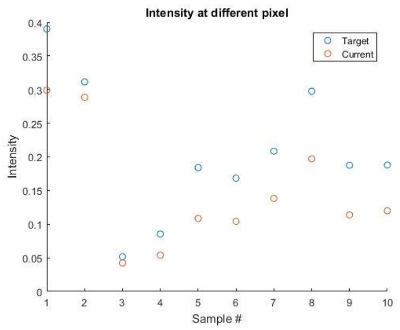
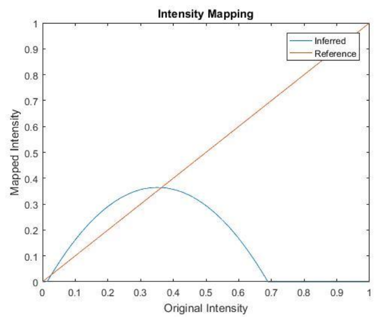
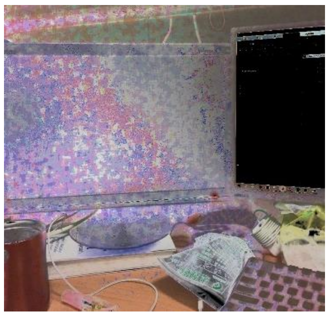
Result

Before			
After			

And instead of using only the brightness of the inlier, we can also consider its neighbor:

Inlier Only	3x3 Region	5x5 Region
		

Seems good, but this method has a fundamental problem: What if the inliers does not span the whole brightness range?

		
The brightness span from 0.05 to 0.4	Brightness map became weird	The resulting image is interesting, though

So, either the polynomial parameters have to be constrained, or the introduction of priors are required.

The method can be much more robust but also uneconomic, by using RANSAC over points in the overlapping area of the two images.