Introduction:

We have an image with three channels representing r, g, b respectively. The goal is to divide this image into three parts and combine them into a colored image. But combining them directly will generate an image with blur and ghosting artifacts because each part doesn’t align to each other. We need to align them effectively to generate correct result.

Implementation Procedure:

First, we divide the input image into three parts. Second, we manually cut a patch as the feature. Note that the more unique the patch is, the better it is. Third, we use normalized cross-correlation method to detect where the featured image located. And then, we can get three position that featured image locate at. Forth, align three image and combine them into one colored image.

Experimental Results:





Test Image:





Discussion:

When detecting the position of featured image, I use the method of sum of squared differences. But the intensity of the image in three channels are different, it failed to detect it correctly. Later I use normalized cross-correlation to solve it, because it doesn’t be affected by mean intensity. By the way, the value after normalized cross-correlation is between -1 and 1. I know it is more similar when the value is larger but I still don’t know the true meaning of the value.

Conclusion:

By using normalized cross-correlation, we can detect the featured image precisely but it cost much time. Not to mention we need to process three channels. To align the image, it took about 3 minutes to process it. Although internal image is good, the surrounding is noisy because of border effects.