VFX HW02 Panoramas

P07922001 李哲安
R07922003 劉濬慶

1. Introduction

By requirements, the homework includes four parts, feature detection, feature matching, image matching and blending. In the following chapters, we will explain our ideas and methods used, also share the experience during the homework.

2. Part 1: Feature Detection

There're many solutions to find out the feature in picture, to complete the homework without too many mistake, we choose Harris Conner Detection for feature detection. According to the slide, we need to find out matrix M and get determinant and trace of M.

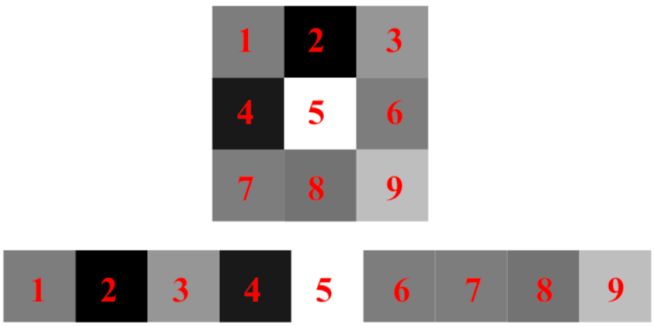
$$M = \sum_{x,y} w(x,y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

$R = detM - k(traceM)^2$, k=[0.04 : 0.06]

The candidates found by Harris are too many to compare, especially lots of pixel close to corner will be detected as corner. (This is related to the window size we used in Harris Corner Detection). To reduce the candidates, we use Local Maximum to keep only one within the window size. In the other word, we only keep one pixel as corner within the window size by its' value R.

Besides only the points in the first 20% will be took as feature points and send to the stage doing comparison.

To describe the features, we use the method in slide, the 8-neighbor which surround it.



For example, if pixel 5 is the feature we found, then we write down the 8 neighbor surround it in line for feature comparison.

Here are two images, the left-side image is the original picture and the right-side one is marked with features with orange color.

3. Part 2: Feature Matching

After part 1, we have lots of features from the image. In this stage, we are going to compare features between two images. Here we have a limitation, image input only with correct order to generate the panorama. If the image coming with random order, the feature detection might be failed as using easy description in feature detection.

To reduce the number of comparison, the pixel to be checked only within +- 10 pixel in height. And the pick up oder is rely on the distance of two pixel. Check the pixel earlier if pixels close to each other. This is because if the camera doesn't have vertical movement, the offset of image only in horizon. Every pixel in two image has same offset, so the distance of two pixels might be the offset the camera movement.

By the feature description in the part 1, here we will calculate the difference of the values in description and reorder by the summation. And using RANSAC to pick up the best offset of two image for the next stage to stitching them together. The RANSAC is an algorithm which random pick up a sample in collection and keep the best into the next round. Because the high resolution picture will take too many time on comparison especially when features are more than expected. We set 500 rounds in RANSAC and return the minimum offset in x-axis and y-axis.

4. Part 3: Image Matching

After getting the offsets of two images, we will merge them into one in this stage. Image image1 and image2 are two pictures just beside each other, and image1 is the left one and image2 is the right one. Two images are overlapping in image3 which is both part of image1 and image2. What we want to do is based on offset and access the overlay part and paste image2 on the right side of the image1.

$$\begin{cases} choose\ image1,\ x < \frac{offset}{2} + threshold \\ choose\ image2,\ x > \frac{offset}{2} + threshold \\ (1-\alpha) * image1 + \alpha * image2,\ otherwise \end{cases}$$

When the pixel landing within threshold, the final output will using Alpha as weighting, and trying to make a fair decision with mixing the two pixels.

5. Result

When doing the homework, we find out that tree is very difficult to handle. The leafs will return many corners with Harris method and make huge loading in comparison. And also if wind blows, corners in leafs will change immediately. To get better result, buildings or simple color and shape component will be easier to handler.

First time we didn't keep camera stable during taking picture, and result is not very well.

But after trying many times to keep the camera stable and keep almost equal movement, we get a better result. (The upper and lower block areas are removed)