

DigiVFX Project2 Report

b03902051 賴岳群 40347011S 張哲魁

1.Description of this project

This is a project combining a series of photos taken by a camera with only rotation under the scene into a panorama. Our implementation is written in C++ under OpenCV library on Windows 8.1.

2.Learned from this project

We learned the idea and implemented several algorithms which are very important to image processing such as feature detecting, feature matching, image transformation, RANSAC and more. Not only panorama stitching but also many applications we can do in the future after learning these algorithms.

3.Description of implemented algorithms and implementation details

(1) Feature detection

We use Harris corner detector, computing intensity, gradient, Gaussian sum, before computing corner response R using $k=0.04$. Then, we collect feature points for those $R > 800000000000$ and R bigger than all 8 neighbors.

(2) Feature matching

First, we assign a feature's neighboring 5×5 color pixels as feature descriptor. Then, for every feature points on the left image, go through every feature points on the right image and find a feature point with minimum distant (sum of square of each pixel's RGB difference) to match. Finally, we label those matching distance within double of minimum distance or 60000 as a good match.

We also make sure a feature point on the right image will not be matched by two or more left image feature point. If so, we choose one match with smallest distance.

(3) Cylindrical projection

Since the focal length provided by Autostitch seems to be incorrect, we calculate focal length by our self using the photo's EXIF-tag and CCD width. Then we compute the projection as the course slides suggested. Also, we project the feature points' coordinates onto the cylinder.

(4) RANSAC

We pick four pairs of matches to estimate the transformation model. Assuming we have 50% of success rate in matching feature points, we randomly draw 72 sets of four matching pairs. We tried to find homography of the four points by implementing normalized DLT but failed. We use opencv's `estimateRigidTransform()` with only the four matching pairs we picked rather than input the whole image which gives the answer directly. We still implement the RANSAC algorithm by counting the inliers within all matching pairs and reserve the model with most inliers.

(5) Image stitching

After we have the transformations between each two neighboring images, we create a long matrix and starts to paste the cylindrical image one by one from left to right using the transformations. If the target pixel being pasted is already a color pixel, we blend the two pixel using the ratio of the original pixel to the original image's center line.

Finally, we crop the extra black rectangle areas away and perform an affine transformation then output the panorama.

4.Results



The total of 17 portrait images ($1000 * 1496$ each) generates a $7927 * 1494$ panorama within about 1000 seconds. There are several noise (black dots) we tried to reduce but couldn't eliminate all due to the transformation is not a one-pixel-to-one-pixel function.

5.Extensions implemented

(none)