CS172 Computer Vision I: Project1: Local Feature Extraction

Yang Hongdi
2019533234
yanghd@shanghaitech.edu.cn

Abstract

This project mainly stitch 4 images into one panorama. The algorithm firstly use SIFT feature extraction to extract the key feature of the two images. Then the euclidean distance of the desciptors is computed and used to match the keypoints. After matching, RANSAC algorithm is used to find the best Homography matrix. As the best Homography matrix is found, we then use it to warp the target image. Finally, we stitch the images and get the panorama.

1. Introduction

In this project, traditional methods are used for image stitching. Python is selected to be the language used in the project. For SIFT desciptors extraction, opencv-python library is used. Scipy.spatial.distance is used to compute the euclidean distance. For Homography matrix computation and RANSAC algorithm, it is implemented by the author.

2. Algorithm Description

2.1. SIFT feature extraction

SIFT(scale-invariant feature transform) is a feature detection algorithm used in computer vision. For desciptor computation, SIFT alogorithm would divide patch into 4x4 sub-patches. Then the histogram of gradient orientations(8 reference angles) is computed inside each sub-patch. So for each desciptor, it will be a 4x4x8 = 128-dimension vector.

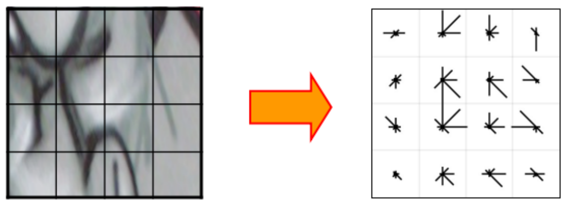


Figure 1. SIFT 128-d descriptor

2.2. Match Keypoints

To describe the similarity of the two keypoints, we use the euclidean distance of the descriptors. As the desciptor is a 128-d vector, the formula of euclidean distance will be

$$\sqrt{\sum_{i=0}^{127}(xi - yi)^2}$$

For each point in the resource image, we find the point which has smallest euclidean distance to it in the destination image. If the samllest distance is smaller than *ratio* * second smallest distance and smallest distance is smaller than the threshold, we match the two points.

2.3. Homography matrix

When we take the photo, we only rotate the camera but not translate the camera. So for any keypoint (x,y), if it has a matching keypoint $(x^{'}, y^{'})$in another image, then we can use a matrix to represent the transformation, that is

$$\lambda \begin{bmatrix} x^{'} \\ y^{'} \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

the matrix is the Homography matrix we need to find. As $\lambda \boldsymbol{x_i^{'}} = \boldsymbol{Hx_i}$, then $\boldsymbol{x_i^{'}} \times \boldsymbol{Hx_i} = 0$, we can get a equation

$$\begin{bmatrix} 0^T & -\boldsymbol{x_i^T} & y_i^{'}\boldsymbol{x_i^T} \\ \boldsymbol{x_i^T} & 0^T & -x_i^{'}\boldsymbol{x_i^T} \\ -y_i^{'}\boldsymbol{x_i^T} & x_i^{'}\boldsymbol{x_i^T} & 0^T \end{bmatrix} \begin{bmatrix} \boldsymbol{h_1} \\ \boldsymbol{h_2} \\ \boldsymbol{h_3} \end{bmatrix} = 0$$

For the 3 equations, only 2 are linearly independent. And as the sacle of matrix H can be aribtrary, so H has 8 degrees of freedom. So finding four points then we can calculate the H matrix.
When we have n points, each point has two equations, then we have a $\boldsymbol{Ah} = 0$ equation.

$$A = \begin{bmatrix} 0^T & -\boldsymbol{x_1^T} & y_1^{'}\boldsymbol{x_1^T} \\ \boldsymbol{x_1^T} & 0^T & -x_1^{'}\boldsymbol{x_1^T} \\ \dots & \dots & \dots \\ 0^T & -\boldsymbol{x_n^T} & y_n^{'}\boldsymbol{x_n^T} \\ \boldsymbol{x_n^T} & 0^T & -x_n^{'}\boldsymbol{x_n^T} \end{bmatrix}$$

A is a 2nx9 matrix. To find the Homogeneous least squares, we can find h minimizing $||Ah||^2$ by using Singular Value Decomposition.
For A, we do SVD.

$$\boldsymbol{A} = \boldsymbol{U} \sum \boldsymbol{V^T}$$

,

$$\boldsymbol{V^T} = [\boldsymbol{v_1}, \boldsymbol{v_2}, ..., \boldsymbol{v_9}]^T$$

then $\boldsymbol{v_9}$ is the eigenvector of $A^T A$ corresponding to smallest eigenvalue. Reshaping $\boldsymbol{v_9}$ into a 3x3 matrix, it will be the Homography matrix we need.

2.4. RANSAC

As randomly choose 4 point to calculate the Homography matrix can lead to bad result, we use RANSAC to find the best Homography matrix. Here is how the RANSAC algorithm works.

- Randomly choose four point to calculate the Homography matrix.
- Using the Homography matrix to transform the keypoints in resource image and compare its coordinate with the matched keypoints in destination image. If the Euclidean distance between them is smaller than threshold, we count the point as a inlier.
- If the number of inliers is the highest we have found so far, we use all inliers to recompute the Homography matrix, and we record the Homography matrix as the best Homography matrix we have found.
- Repeat above steps for N times, and return the best Homography matrix at last.

2.5. Image Stitching

For this part, we stitch the image by using cv2.warpperspective(). To stitch the two image and show the whole image, the corners of the image after warping should be computed to calculate the offset. For stitching the image from the left, we need to calculate the coordinate of its topleft corner, then form a new matrix

$$\boldsymbol{H'} = \begin{bmatrix} 1 & 0 & widthOffset \\ 0 & 1 & heightOffset \\ 0 & 0 & 1 \end{bmatrix} \boldsymbol{H}$$

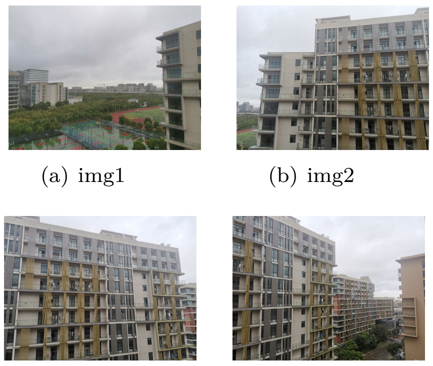
Then we can stitch the two image together and show the whole stitched image.

3. Code Implementation

3.1. Run code

python imageStitch.py

3.2. The original images


(a) img1 (b) img2

(c) img3 (d) img4

Figure 2. original images

3.3. Feature extraction

We use img1 and img2 as example.
After feature extraciton, we can see the keypoints on the image.

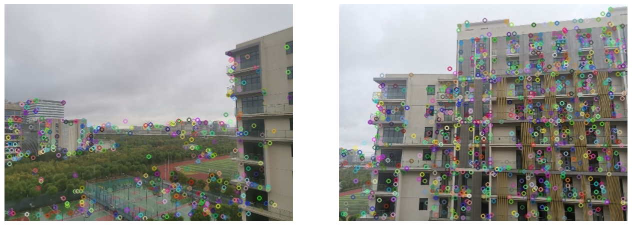
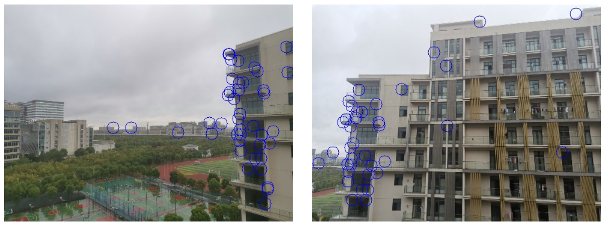

(a) img1 (b) img2

Figure 3. Keypoints

Each keypoint has a 128-d desciptor, which will be used to do matching in next section.

3.4. Keypoints Matching

We use img1 and img2 as example.
Apply the keypoints matching algorithm. We can see the matched keypoints on the image.



(a) img1 (b) img2

Figure 4. Keypoints matched

As to reduce run time, the images are resized so there might not be so many matched keypoints, but it is shown that most of the keypoints are matched.

3.5. Apply Homography matrix

We use img1 and img2 as example.
After getting the Homography matrix, we Apply it on img1. Then we stitch img1 and img2.

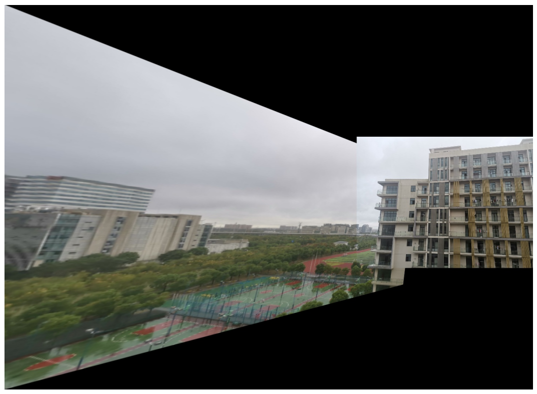


Figure 5. stitch img1 and img2

We can see the two images are properly stitched.

4. Result

We first stitch img1 and img2 from the left to get img12, then stitch img12 and img3 from the left to get img123, then stitch img123 and img4 from the right to get the final result.

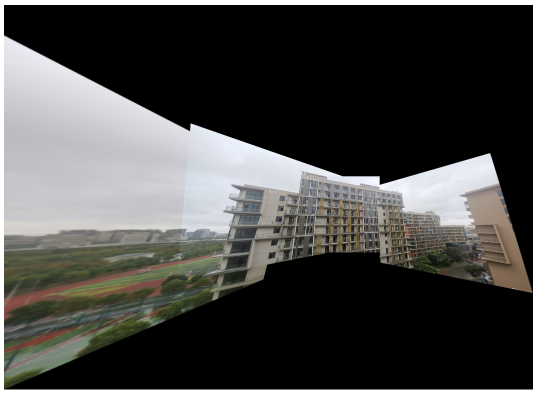


Figure 6. Final result

5. References

1. Singular Value Decomposition (SVD) tutorial http://web.mit.edu/be.400/www/SVD/Singular_Value_Decomposition.htm