Computer Vision E1 216 Assignment 2

Bazil Ahmed

March 17, 2020

1 Introduction and Overview

We are given a bunch of images taken by rotating the camera after each click. We are required to find the homography corresponding to each rotation between the clicks and then eventually stitch the images together to form mosaic.

1.1 Computing Homography

To compute homography between any pair of overlapping images, we are required to find at least 8 pair of coordinates in both the images, which are projection of the same 3D points(i.e., every pair corresponds to different 3D point).

1.2 SIFT Features

To find pixel pair from two images corresponding to a 3D point we need SIFT feature extraction, an algorithm to extract image features. In this assignment, I am using the given Raw features provided as part of the assignment.

2 Theoretical Background

Homography is a 3x3 linear transform having 8 degree of freedoms. If a 3D point O has projection P_1 in image1 and P_2 in image2 (P_1 and P_2 are in homogeneous form), then these two points can be related by the homography(H_{12}) corresponding to the rotation of a particular camera between the two clicks as:

$$P_2 = H_{12}P_1$$

let's say,

$$P_1^T = [x_1, y_1, 1], P_2^T = [x_2, y_2, 1] \;\;\&\;\; H_{12} = \begin{bmatrix} H_{11} & H_{12} & H_{13} \\ H_{21} & H_{22} & H_{23} \\ H_{31} & H_{32} & H_{33} \end{bmatrix}$$

$$\begin{aligned} &then, \\ x_2 &= \frac{H_{11}x_1 + H_{12}y_1 + H_{13}}{H_{31}x_1 + H_{32}y_1 + H_{33}} \\ y_2 &= \frac{H_{21}x_1 + H_{22}y_1 + H_{23}}{H_{31}x_1 + H_{32}y_1 + H_{33}} \end{aligned}$$

which on cross multiplication gives linear equations as follows,

$$a_1^T h = 0$$
$$b_1^T h = 0$$

where,

$$
\begin{aligned}
a_1^T &= [-x_1, -y_1, -1, 0, 0, 0, x_2x_1, x_2y_1, x_2] \\
b_1^T &= [0, 0, 0, -x_1, -y_1, -1, y_2x_1, y_2y_1, y_2] \\
h^T &= [H_{11}, H_{12}, H_{13}, H_{21}, H_{22}, H_{23}, H_{31}, H_{32}, H_{33}]
\end{aligned}
$$

Now, to compute as above, we require at least 8 matching pairs(8 degree of freedom). We use SIFT feature extraction for feature matching. But as SIFT might produce outliers, we need to filter outliers using RANSAC, which is an iterative algorithm. We randomly pick 4 pairs out of all pairs given by the SIFT feature extraction and then compute homography by forming a matrix A consisting of above linear equations corresponding to each picked pair. Then solving $Ah = 0$, using SVD. Once we get the homography, we test this against every matched pair provided by the SIFT feature extraction. We count the number of inliers as the number of matched pairs under some threshold error. Then select the best homography as the one which gives maximum number of inliers.
Then a particular image, out of all images given for stitching(preferably central image),is used as a universal frame of reference and all images are transformed into that reference frame by suitable transformation of homography matrices.

3 Algorithm Implementation and Complications

For computing homography, we require matrix A to be well conditioned as it might range from order 0 to 10^5 as images may be of large dimensions. For this I have first translated every matched pair by translating their origin to their respective centroid. Now after this I have scaled the coordinates such that there mean euclidean distance from origin is $\sqrt{2}$.
Then feeded these normalized pairs to RANSAC loop(set to 200 iterations, chose after several experiments). For testing for inliers I used:

$$
\begin{aligned}
a_1^T h &< \epsilon \\
b_1^T h &< \epsilon
\end{aligned}
$$

where $\epsilon = .0005$ again experiment based value.

NOTE: *The best homography obtained from RANSAC using the randomly selected 4 pairs is being used directly for further stitching process as fitting it to the selected inliers was giving poor results.*

Once we got the homography, I transformed it to original coordinate system by reversing the transformations done during normalization. Then I calculated the dimension of the mosaic by first converting the coordinates of the corners of the images to the reference frame coordinate system. And then finding minimum and maximum of x and y coordinate.
Once I got the dimensions for the mosaic, I iteratively check each pixel coordinate(which is in central reference frame) if it lies in any image which has been transformed by suitable homography to the reference frame of central image. I take the bi-cubic interpolation of the transformed coordinate as it might not be integer coordinate after transformation. If it lies in any overlapping region then i take the average of the interpolated pixel values of the images involved in forming the overlapping region. Here I am not using any linear blending hence the results shows some blocky effect(image boundaries are visible).
It takes about half an hour to produce the mosaic(implementation has been done in python).

4 Computational Results

Here I am attaching the results obtained.

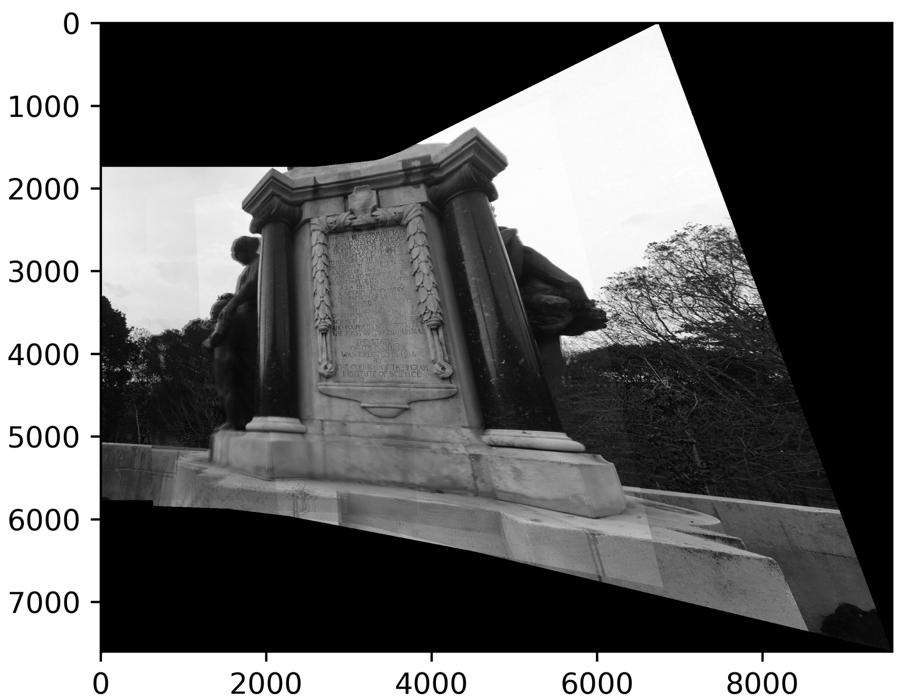

Figure 1: Gray scale Mosaic when image1 was set as the reference frame

Figure 2: Gray scale Mosaic when image3(central image) was set as the reference frame

Figure 3: Colored Mosaic when image3(central image) was set as the reference frame

5 Summary And Conclusions

It's important to choose reference frame carefully as you can see from the above results that choosing image1 as the reference frame might result in a stretched mosaic. While with central image as the reference gives better mosaic.
Averaging for blending might not give neat result, hence proper blending should be done as it can remove blocky effect significantly.
Sequential stitching(create mosaic with 1st two images and then use the resultant mosaic for stitching with the next image in sequence) is not a good idea. As it is against the principles of homography. And give very poor results.
Conditioning is also an important aspect for proper homography computation.