

CSE 573
Homework 3
Enze Qian
50091378

Part I:

(a) Describe my solution:

Step 1: Load images by `imread()`. Convert them by `rgb2gray()` and `im2double()`.

Step 2: Detect feature by function `harris.m`, use parameter `sigma=3` `thresh=0.01` `radius=3` `disp=0` as recommended in its description.

Step 3: Form descriptor by using `find_sift.m` which is provided in code folder. The function `find_sift` takes `img`, `circle` and `enlarge_factor` as parameters. I use `enlarge_factor=1.5` as it recommends. The parameter `circle` is what returns from `harris.m` function. (`x,y, radius=3`).

Step 4: Compute distances between every descriptor by using `dist2.m` function.

Step 5: Select putative matches based on the matrix of pairwise descriptor distances obtained above. For here, I choose select the top three hundreds descriptor pairs with the smallest distances. The reason I'm not using a specified threshold is that we have no information about the distance. In order to determine the threshold, I have to check the distance matrix to get a reasonable threshold value. Thus, getting top few hundred is more convenient.

Here is an interesting implementation choice I made, in order to get few hundred pairs with the smallest distance, a simple implementation is sorting the distance matrix. However, the run time could be long if the matrix is very large. Hence, I use another implementation as below:

```
temp=distance(:);
for i=1:N
    [~,indx(i)]=min(temp);
    temp(indx(i))=inf;
end
[idx1,idx2]=ind2sub(size(distance),indx);
```

It gives me a shorter running time compared to `sort(distance(:))`

Step 6: Run RANSAC

My `step6.m` function takes the matching pairs in two images `x1, y1, x2, y2` as parameters and returns estimate homography, inlier number and average residual.

My RANSAC function takes 30000 iterations and I choose `threshold=5` which considers those squared distances less than 5 as inliers. Then the function `H_fitting` is used for calculated homography fitting, in which I construct matrix `A` as follow:

From $n \geq 4$ corresponding points:

$$A h = 0$$

$$\begin{pmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -x_1x'_1 & -y_1x'_1 & -x'_1 \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -x_1y'_1 & -y_1y'_1 & -y'_1 \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -x_2x'_2 & -y_2x'_2 & -x'_2 \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -x_2y'_2 & -y_2y'_2 & -y'_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_n & y_n & 1 & 0 & 0 & 0 & -x_nx'_n & -y_nx'_n & -x'_n \\ 0 & 0 & 0 & x_n & y_n & 1 & -x_ny'_n & -y_ny'_n & -y'_n \end{pmatrix} \begin{bmatrix} h_{1,1} \\ h_{1,2} \\ \vdots \\ h_{3,3} \end{bmatrix} = 0$$

then use function `svd()` for solving H .

In each iteration, it selects random sample and calculated its inliers number and homography estimation. I finally choose the sample which can get most numbers of inliers. Then the average residual is calculated by the total sum distance from the sample I chosen divided by the number of inliers.

Step 7 8 9: I finished these three steps in one function: `stemp78.m`. First, I create a multidimensional spatial transformation structure T for the left image by function `maketform` and the homography estimation I got from step6. Then I wrap the left image by function `imtransform` and determine the size of the new image(first and last row/column). After that, I project the left and right images to a image whose size is same as the new image. Finally, I combine them together and take the average where the two image are overlap.

(b). Report the number of inliers and average residual and display the matches:

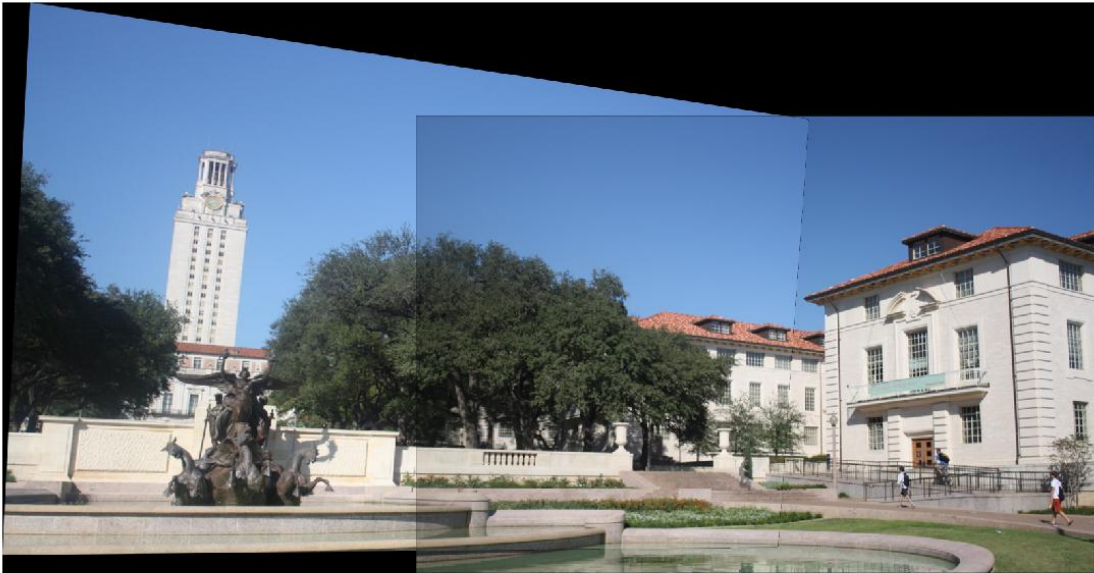
InlierNum=152

AvgResidual=1.2864

The location of inlier matches in both images:



(c). Display final result:



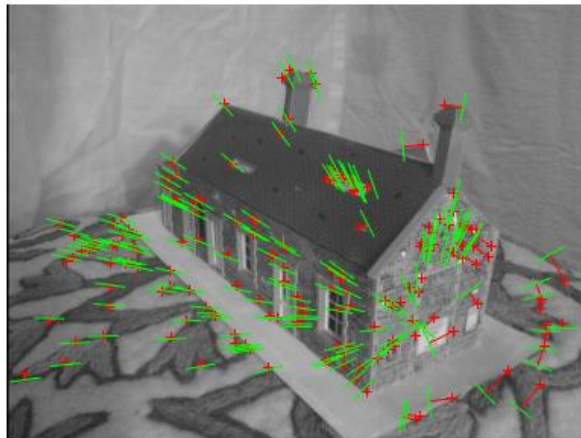
Part II:

(a) part2.m set step=2

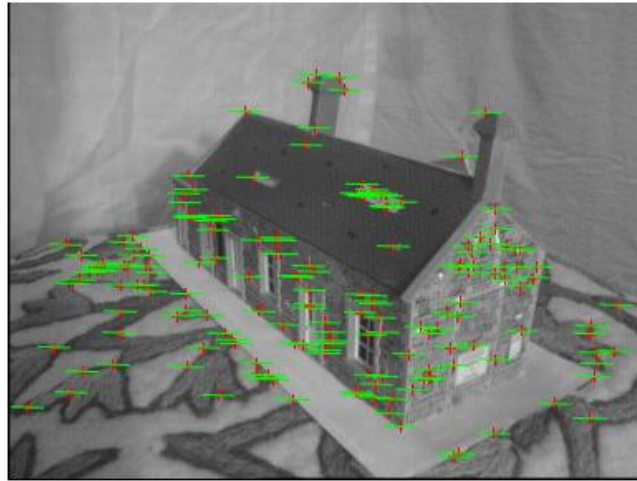
Unnormalized estimation (In fit_fundamental.m, set normal=0.)

Normalized estimation (In fit_fundamental.m, set normal=1.)

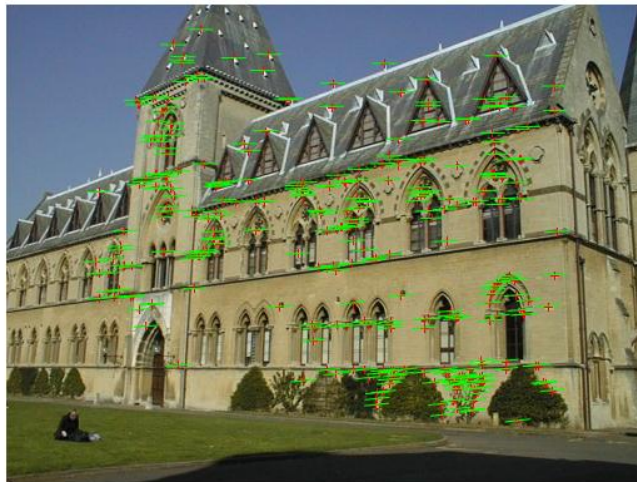
Average residual is the value of the variable *distance* in line 37, part2.m



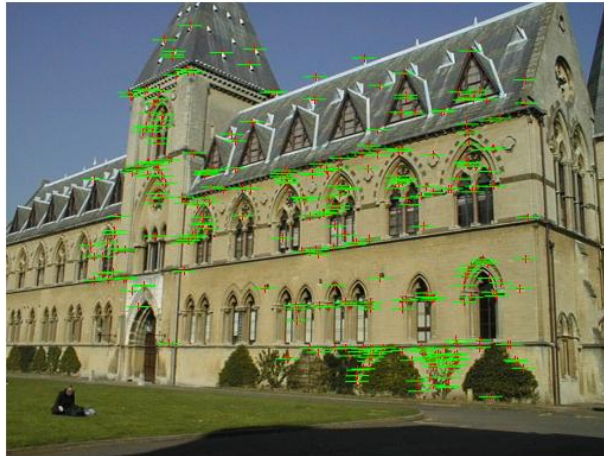
House_unnormal.jpg
Average residual=3.0342



House_normal.jpg
Average residual=0.2103

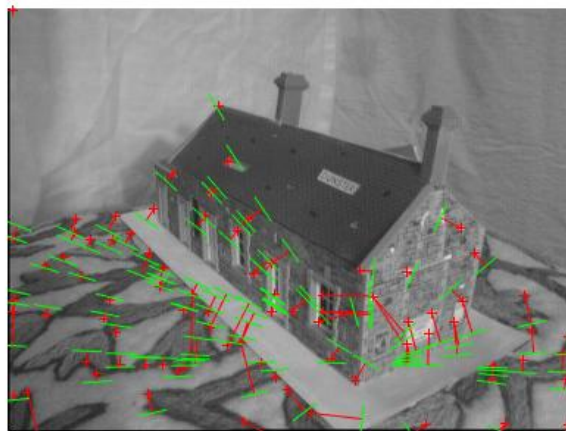


Library_unnormal.jpg
Average residual=0.3385

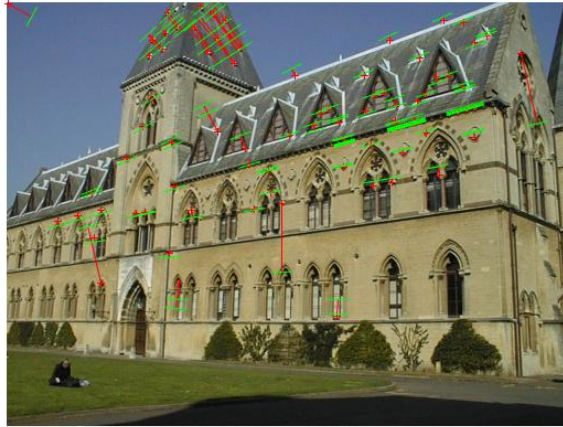


Library_normal.jpg
Average residual=0.1835

(b). Use putative match generation and RANSAC: In part2.m, set step=3.
Number of inliers and average residual are printed by RANSAC_part2.m and part2.m.



House_RANSAC.jpg
Number of inliers=124
Average residual=11.3488

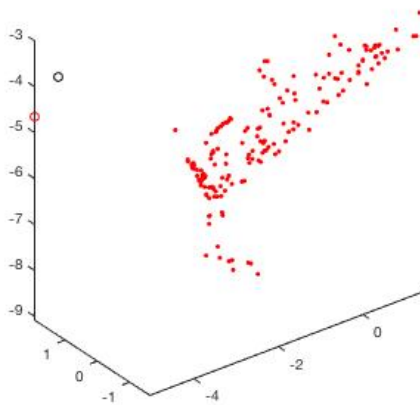


Library_RANSAC.jpg
Number of inliers=155
Average residual=7.8186

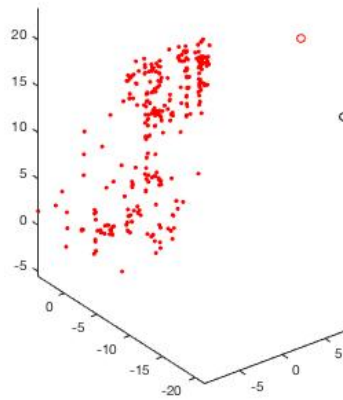
Compare with step2: The result of RANSAC estimation is more like the result of unnormalized method in step2.

(c). 3D camera centers and triangulated 3D points

Run camera.m, centers can be generated by the function get_center and residuals between the 2D points and projected 3D points are generated by the function calc_residuals.



House_camera.jpg
Residuals of image1=6.4742
Residuals of image2=50.4092
Center of image1: Black circle
Center of image2: Red circle



Library_camera.jpg
Residuals of image1=51.5323
Residuals of image2=81.2813
Center of image1: Black circle
Center of image2: Red circle