Problem Set 7

Resources with corresponding images and code are on Stellar under `andrewmo@mit.edu`. The files are in the `pset7.zip` folder. To reproduce the below figures, run `pset7_main.m`.

Problem 7.1

For this problem, we stitch overlapping photos together to create a panorama. We will first utilize SIFT descrpitors in the photos and match them to obtain correspondences. RANSAC will be used to help account for errors and outliers so our algorithm is more robust.

(A) Lets do a simple example of using RANSAC to fit a 2D line. Originally, I sampled points on the line $y = 2x + 1$. To see the advantage of RANSAC, I sampled points from the line plus some minor random error. I sampled 30 points uniformly from $[0, 10]$ and introduced 3 outliers. Below are the results that get us fairly close to the correct line.
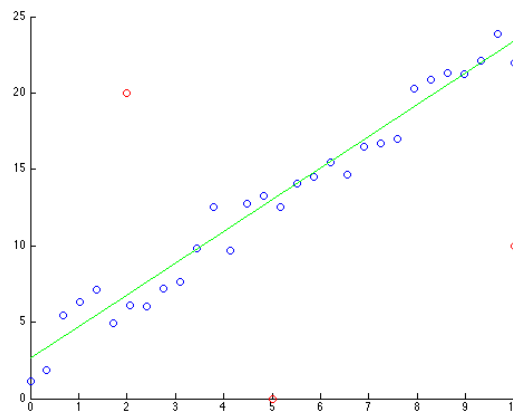

Figure 1: 2D line-fitting using RANSAC. The resulting line (green) after using RANSAC on 30 inliers (blue) and 3 outliers (red). Parameters include 100 iterations, 2 samples used per iteration, a distance $\epsilon = 2$ when determining if point is an inlier, and an acceptable probability of error of 1e-9. RANSAC will be described in more detail in later parts.

(B) Next we compute SIFT features for both images `seoul1.jpg` and `seoul2.jpg`. Figure 2 shows a plot of the matching features and correspondences.

(C) This question pertains to the function `H = FitHomograohy(x1, x2)`. Given a set of points from two images, we want to compute the homography from the set of points from the
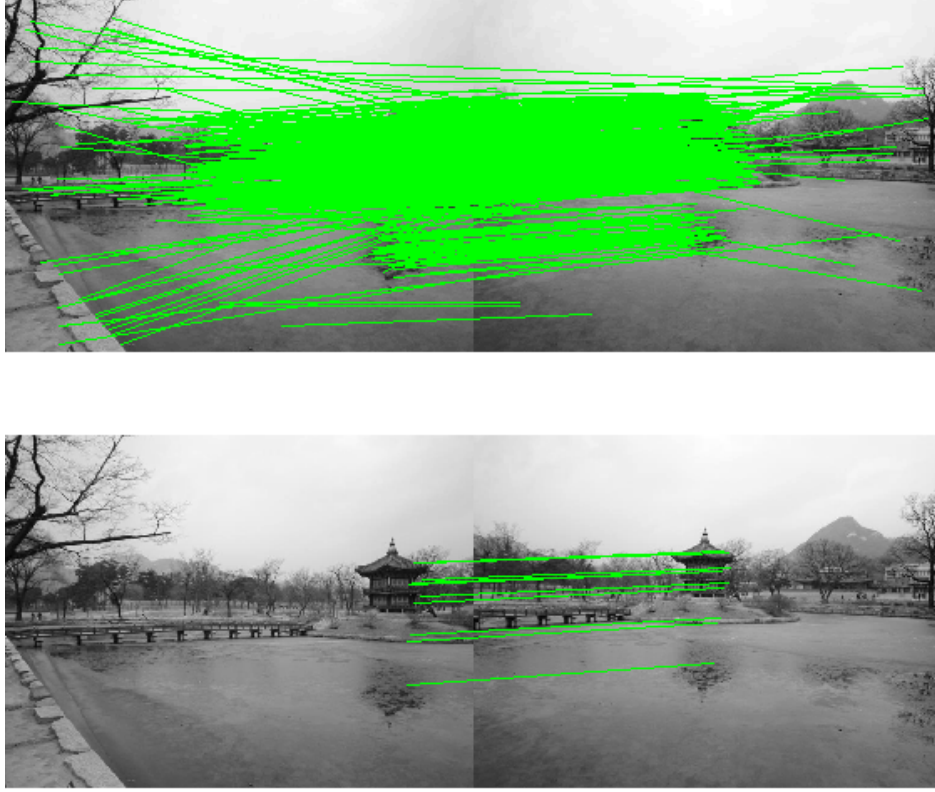
Figure 2: Resulting matches and correspondences after running SIFT on the overlapping images. 10 matches (bottom) is a more clearly viewed subset from the original 808 matches (top).

overlapped image to the original. This can be reduced to solving a least squares problem. We would like to minimize $||Ah = 0||^2$, where $Ah = 0$ is defined below.

$$\begin{bmatrix} x_n & y_n & 1 & 0 & 0 & 0 & -x'_n x_n & -x'_n y_n & -x'_n \\ 0 & 0 & 0 & x_n & y_n & 1 & -y'_n x_n & -y'_n y_n & -y'_n \\ ... & & & & & & & & \end{bmatrix} \begin{bmatrix} h_{00} \\ ... \\ h_{22} \end{bmatrix} = \begin{bmatrix} 0 \\ ... \\ 0 \end{bmatrix}$$

A is a $2n$x9 matrix and h is a 9x1 matrix. The solution is the unit vector of h, which is the eigenvector of $A^T A$ with the smallest eigenvalue. Normalization is necessary to account for scaling (ensures third component of 3D point is 1).

(D) This question pertains to H = TransformRANSAC(x1, x2). To make the algorithm more robust from outliers, the goal is to perform RANSAC to get the best homography H with the least probablity of error. In addition to the input and output points for the corresponding images,

tunable parameters to the algorithm are necessary and shown below.

```
Niter = 10000; %Number of iterations
epsilon = 10; %Distance threshold when determining if inlier
P = 4; %Number of samples to use when fitting H to best seen so far
acceptableProbFailure = 1e-9; %Acceptable probability of failure
```

Best results were seen with 10000 iterations and an ϵ of 10, however, for reasonably quick results, 1000 iterations and an ϵ of 20 was used (Please refer to below figures for results). For P, At least 4 sample points is needed to solve the least squares equation. The acceptable probability of failure was arbitrary. When running RANSAC, we want to limit the failure rate. This can be accounted for by performing $P(failure) = (1 - G^P)^i$, where G is the percentage of inliers to total points and i is the number of iterations seen thus far. Please refer to `TransformRANSAC.m` for a better description/overview of the algorithm.

(E) This question pertains to `im = MakePanorama(im1, im2, H)`. The goal is now to produce a panoramic stitch of a H transformed version of im2 onto im1. To get padding working correctly, I utilized the built-in MATLAB function `imtransform`. Therefore, a transformed version of an image can be easily implemented as seen below.

```
T=maketform('projective',H);
im2t=imtransform(im2,T,'XYScale',1);
```


Figure 3: Transformed image of `seoul2.jpg` after applying H optimized by RANSAC.

(F) This question pertains to the function `im = Photomerge(im1, im2)`. This combines the previous parts where an overall panorama is produced given two images. The first image is used as reference. The summarized steps to produce the panorama is as follows:

- Produce matches/correspondecies by running SIFT

- Extract points `x1,x2` from matches.

- Get H: `H = TransformRANSAC(x2,x1)` (want to transform im2)

- Get stitched image: `im = MakePanorama(im1, im2, H)`

Figure 4: Results after running `PhotoMerge.m` on the first two seoul images. This used $N = 1000, \epsilon = 20$.

(G) This question pertains to the function `im = PanoramaN(im1, im2, ..., imN)`. This function creates a panorama from multiple images and sets the middle image in the list as reference. Refer to `PanoramaN.m` for implementation. Basically, I perform `PhotoMerge` N-1 times, keeping track which image (left onto right, right onto left) needs to be performed. Refer to Figure 5 for images $n > 2$ resulting panoramas. It should be noted that when stitching images together, it was necessary to blend pixels as best we could. It is important that both images are of the same dimension, with added padding if necessary. When images intersect, take the average of both images. If they don't intersect, take the value of the valid image. A summary of choosing/blending pixels is shown below, taking advantage of masking.

$$I_{out} = I_1(M_1 \& \sim M_2) + I_2(M_2 \& \sim M_1) + (I_1/2 + I_2/2)(M_1 \& M_2)$$

Figure 5: Results after running `PanoramaN.m` on the 3 seoul images (top) and the 4 riverbed images (bottom). They both used used $N = 10000, \epsilon = 10$. Note: with more images used, stitching and finding inliers relies on the parameters used, therefore, some artifacts may exist.

(H)(i) This question pertains to why a homography suffices for aligning two images. Let $P_1 = K[I|t_1]$ and $P_2 = K[R|t_2]$ be the corresponding camera matrices. Let $x_1 = P_1X$ and $x_2 = P_2X$. Show that there exists a homography H such that $x_2 = Hx_1$.

$$x_2 = Hx_1$$
$$P_2X = HP_1X$$
$$K[I|t_1]X = HK[R|t_2]X, \text{apply substitutions}$$
$$K[R|t_2]^{-1}K[I|t_1]X = HX, \text{apply inverse rotation matrix}$$
$$K[R|t_2]^{-1}K[I|t_1]XX' = H, \text{where } X' = X^{-1} \text{ (inverse 3D world point)}$$

This shows that there exists a homography H that does the transformation from x_1 to x_2.

(H)(ii) Optional.