Assignment 4: 3D Reconstruction

Michael Shepherd, 19059019

September 2019

Question 1

Figure 1: Image 1

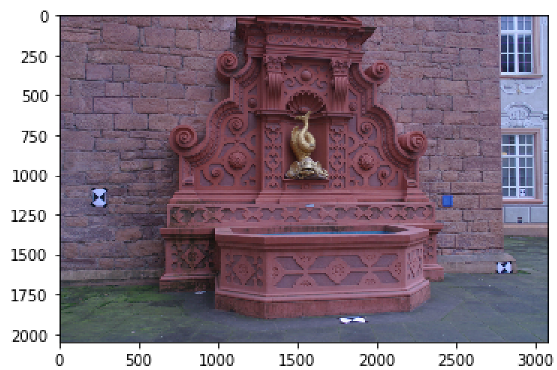


Figure 2: Image 2

1a

Below, I have plotted all the matches and all the matches that have a euclidean distance of less than 600 between them on image 1. The Blue points denote markers on image 1 and the red point denotes the match of that marker from image 2. The threshold of 600 was chose, as it gave the widest variety of points at different depths while still removing the most obvious false matches. We can see that some of the matches from figure 4 are definitely wrong, but we trust that the later algorithms will handle those issues. The rest

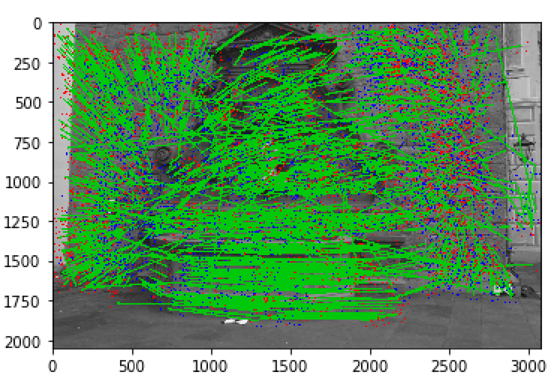


Figure 3: All Matches

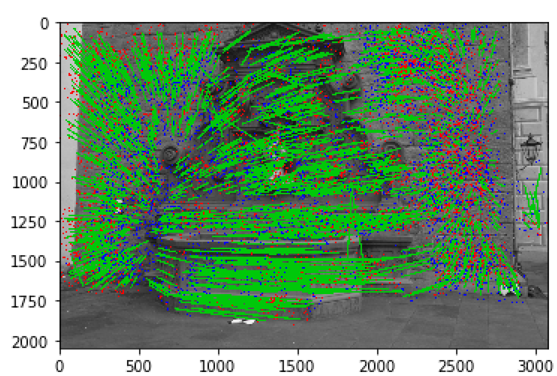


Figure 4: Matches with 600 Threshold

of this question uses a RANSAC based procedure to find the inliers in a set of matches. This means that the outliers will be removed later, so it is okay to have a few of them as long as we have enough inliers. A lenient threshold then allows for us to have more inliers, at the expense of more outliers, but this will work well for the rest of the pipeline.

1b

Using the following algorithm, we can narrow the matches down even further into a set of inliers by randomly estimating F. This allows us to find the largest consensus set and re-estimate F from that consensus set.

The data we have: matches $(x_i, y_i, 1) \leftrightarrow (x'_i, y'_i, 1)$

1. repeat many times:
   - 1.1 choose 8 matches randomly
   - 1.2 calculate F (as we did in Lecture 16)
   - 1.3 for every match i, find the Sampson distance $d_F(\underline{x}_i, \underline{x}'_i)$; those with a sufficiently small distance form the consensus set
2. pick the largest consensus set found: this is our set of inliers
3. re-estimate F in a least-squares sense, using the entire set of inliers

Figure 5: RANSAC-based computation of F

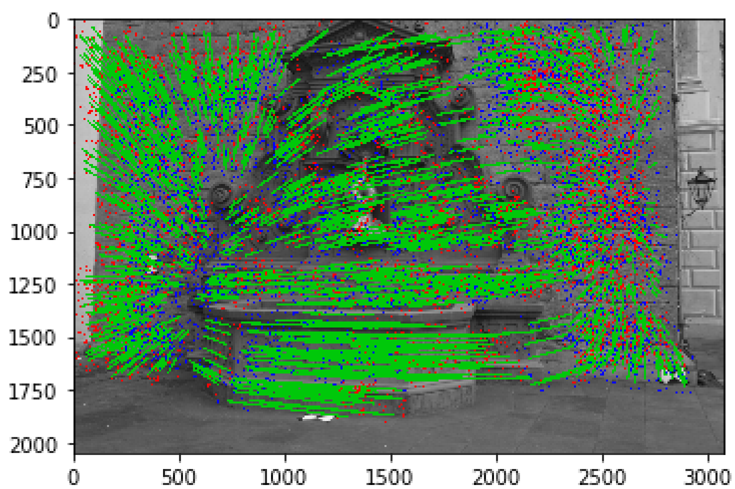


Figure 6: RANSAC Matches with Sampson distance threshold 10 after 1000 iterations

$$F = \begin{bmatrix} 0.11425088069384726 & -1.0976443853235909 & 1216.8066064325246 \\ 2.92293845935525 & 1.0147679529035127 & 37606.20381954895 \\ 1075.4820032551422 & -46003.982510832626 & 3514754.5090216585 \end{bmatrix}$$

1c

Using the formula $E = K'^T F K$, the given K matrices and the F matrix from the previous question: (The following fix was also included to prevent unwanted reflections)

```
[U,S,V] := svd(E)
if (det(U) > 0) and (det(V) < 0) then  E := -E, V := -V
elseif (det(U) < 0) and (det(V) > 0) then  E := -E, U := -U
```

$$E = \begin{bmatrix} 0.08660498 & -1.06546641 & 0.24365151 \\ -2.45507609 & -0.13768798 & -9.22912999 \\ -0.13950599 & 9.58493463 & -0.11437703 \end{bmatrix}$$

With singular values:

$$9.65$$

$$9.55$$

$$0.00924$$

$$U = \begin{bmatrix} 0.11352911 & 0.00459973 & -0.99352402 \\ -0.17993405 & -0.98335805 & -0.02511357 \\ -0.97710536 & 0.18161992 & -0.11081212 \end{bmatrix}$$

$$V = \begin{bmatrix} 0.06092623 & 0.25016302 & -0.96628487 \\ -0.98055495 & 0.19593046 & -0.01110124 \\ 0.18654752 & 0.94817178 & 0.25723589 \end{bmatrix}$$

We can see that the first two singular values are approximately equal to each other and the third singular value is approximately equal to zero.

1d

We use the following equations to find the camera matrices:

$$P = K[I|\underline{0}]$$
$$\text{and} P' \text{ is one of the following:}$$
$$P' = K'[UWV^T|\underline{u}_3] \text{ or}$$
$$P' = K'[UWV^T|-\underline{u}_3] \text{ or}$$
$$P' = K'[UW^TV^T|\underline{u}_3] \text{ or}$$
$$P' = K'[UW^TV^T|-\underline{u}_3]$$
$$\text{With:}$$
$$W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

We then use triangulation to pick the correct P' from the four options by using a point $\underline{X}$ which is a point in real world coordinates that is in the frame of both cameras:

$$\underline{n}_1 = R_1^T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\underline{n}_2 = R_2^T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$if\ \underline{n}_1(\underline{X} - \underline{C}_1) > 0\ and\ \underline{n}_2(\underline{X} - \underline{C}_2):$$
$$\underline{X} \text{ is in front of both cameras}$$

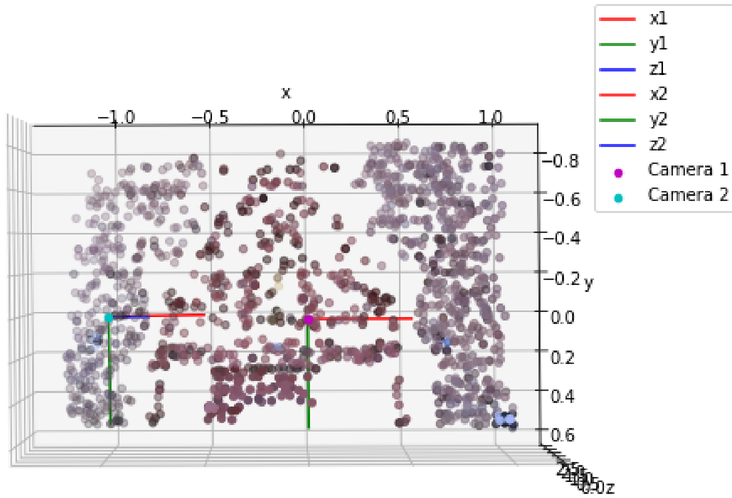
We then find that:

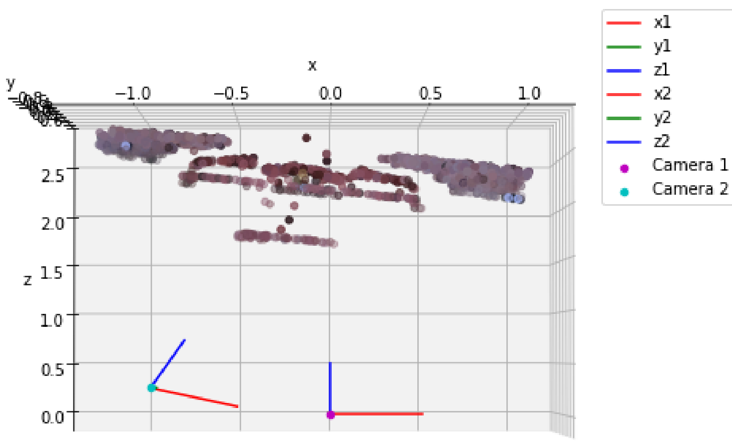
$$P = \begin{bmatrix} 2759.48 & 0.0006 & 1520.69 & 0 \\ 0 & 2764.17 & 1006.81 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}$$

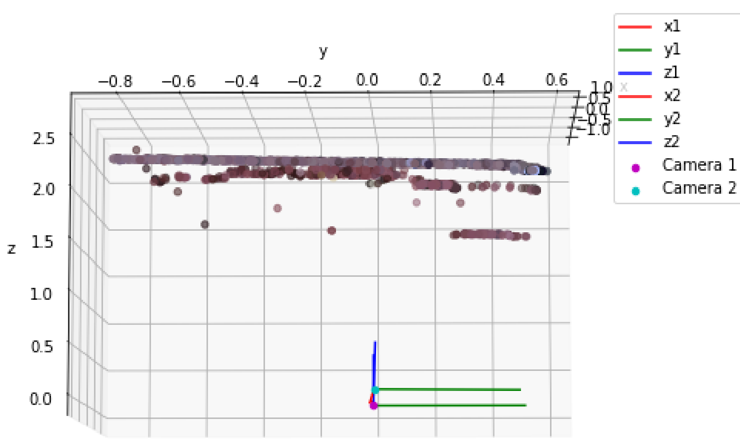
$$P' = \begin{bmatrix} 3122.95 & -21.21 & 417.13 & 2910.12 \\ 390.94 & 2778.21 & 884.85 & 180.98 \\ 0.36 & 0.01 & 0.93 & 0.11 \end{bmatrix}$$

1e

I used the camera matrices and the RANSAC inliers to plot the object in 3D. It was clear that some of the points that RANSAC had chosen did not fit in with the data set, so I implemented a distance threshold to remove the points that were clearly too far away from the cameras.







Question 2

2a

The idea of image rectification is to projectively transform each of the two images, so that the epipolar lines become parallel and perfectly horizontal, to ease the search for correspondences.

To do this, we find homographies to rectify each image as follows:

$$
\begin{aligned}
P_1 &= K_1 R_1 [I| - \underline{C}_1] \\
P_2 &= K_2 R_2 [I| - \underline{C}_2] \\
\text{let } K_n &= \frac{1}{2}(K_1 + K_2) \\
\text{let } \underline{r}_1 &= \frac{\underline{C}_2 - \underline{C}_2}{||\underline{C}_2 - \underline{C}_2||} \\
\text{let } \underline{r}_2 &= \frac{\underline{k} \times \underline{r}_1}{||\underline{k} \times \underline{r}_1||} \\
\text{let } \underline{r}_3 &= \underline{r}_1 \times \underline{r}_2 \\
\text{let } R_n &= \begin{bmatrix} \underline{r}_1^T \\ \underline{r}_2^T \\ \underline{r}_3^T \end{bmatrix} \\
\text{with } \underline{k} &= R_1^T \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}
\end{aligned}
$$

We can then define the two homographies:

$$T1 = K_n R_n R_1^T K_1^{-1}$$

$$T1 = \begin{bmatrix} -0.8082241105259916 & -0.004098120205764376 & 2585.7131284583616 \\ 0.10614522310428495 & -0.9994729671392577 & 2002.2857781365892 \\ 0.0001003738066990551 & 5.089393083788065e-07 & 0.8077243399613508 \end{bmatrix}$$

$$T2 = K_n R_n R_2^T K_2^{-1}$$

$$T2 = \begin{bmatrix} -1.044975745143705 & -0.026070465735595436 & 3276.9968797030497 \\ -0.012656268074460079 & -1.004613076385177 & 2103.173297896742 \\ -3.2327449697251866e-05 & -4.8435481698941085e-06 & 1.0499599585651216 \end{bmatrix}$$

After applying these homographies to their respective images, using the minx and miny to shift the origin correctly, we get these rectified images:

Figure 7: Image 1 Rectified

Figure 8: Image 2 Rectified

2b

Once the images have been rectified, we can see that the epipolar lines in the images are perfectly horizontal. This shows us that any feature in the one rectified image will match with a point in the other at the same vertical coordinate of that feature. This is due to the camera matrices sharing the same intrinsic parameters, K_n and R_n.

It is clear that the top left hand corner of the image may not be the origin anymore. This is due to the fact that to o rectify the images, we warp them, and in some cases, the new warped shape means that we must increase or decrease the bounds of the image, which has the chance of changing to origin to a pixel that is no longer a part of the original image.

To find the epipolar lines, we first needed to find the camera matrices for the new rectified images:

$$P_1' = K_n R_n [I| - \underline{C}_1]$$
$$P_2' = K_n R_n [I| - \underline{C}_1]$$

With these new camera matrices, we can now draw the epipolar lines, as we did in the previous assignment, using the min_y values from applying the homographies to shift the points so that we ensure that we are drawing correct corresponding lines:

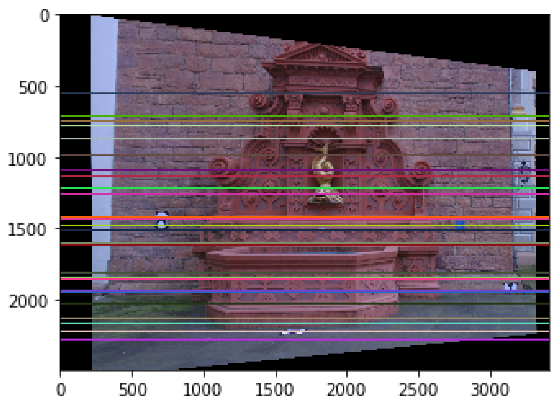


Figure 9: Image 1 rectified with epipolar lines

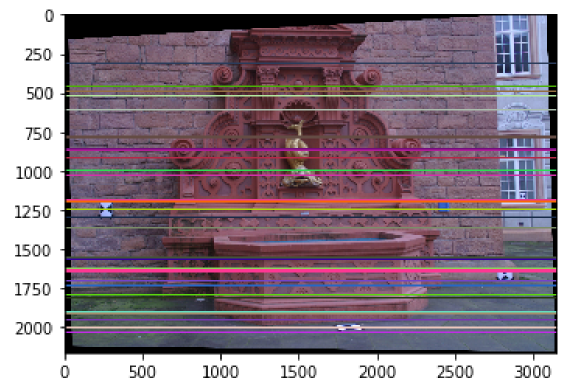


Figure 10: Image 2 rectified with epipolar lines