Hi TA's! I'm using 2 late day points on this assignment! Best, Kevin

Q1.1

Using the fundamental matrix equation, we know that

$$\tilde{\mathbf{x}}_2^{\mathrm{T}}\mathbf{F}\tilde{\mathbf{x}}_1 = 0$$

In homogeneous coordinates, we know that

$$\tilde{\mathbf{x}}_1 = \tilde{\mathbf{x}}_2 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} f_{11} & f_{12} & f_{13} \\ f_{21} & f_{22} & f_{23} \\ f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = 0$$

$$\begin{bmatrix} f_{31} & f_{32} & f_{33} \end{bmatrix} \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = 0$$

$$f_{33} = 0$$

Q1.2

If the there is only translation parallel to the x-axis, we can define the rotation and translation matrices as

$$\mathbf{R} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{t} = \begin{bmatrix} t_x \\ 0 \\ 0 \end{bmatrix}$$

We can use these to define the essential matrix

$$\mathbf{t}_\times = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -t_x \\ 0 & t_x & 0 \end{bmatrix}$$

$$\mathbf{E} = \mathbf{t}_\times \mathbf{R} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -t_x \\ 0 & t_x & 0 \end{bmatrix}$$

We can then get the equations for the epipolar lines

$$\mathbf{l}_1 = \tilde{\mathbf{x}}_2{}^{\mathrm{T}}\mathbf{E} = [x_2 \quad y_2 \quad 1] \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -t_x \\ 0 & t_x & 0 \end{bmatrix} = [0 \quad t_x \quad -y_2 t_x]$$

$$\mathbf{l}_2 = \tilde{\mathbf{x}}_1{}^{\mathrm{T}}\mathbf{E}^{\mathrm{T}} = [x_1 \quad y_1 \quad 1] \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & t_x \\ 0 & -t_x & 0 \end{bmatrix} = [0 \quad -t_x \quad y_1 t_x]$$

The equations of the two lines can be written as

$$(1)\ t_x \mathrm{y} = y_2 t_x$$

$$(2)\ t_x \mathrm{y} = y_1 t_x$$

Since neither of these equations depend on x, they both describe lines that are parallel to the x axis.

Q1.3

Let's start with a 3D point in the real world, $\mathbf{x}$. At time 1 and time 2 we will see x in two different camera frames

$$\mathbf{x}_1 = \mathbf{R}_1\mathbf{x} + \mathbf{t}_1$$

$$\mathbf{x}_2 = \mathbf{R}_2\mathbf{x} + \mathbf{t}_2$$

We can use the first equation to solve for x, and plug this result into the second equation

$$\mathbf{x} = \mathbf{R}_1{}^{-\mathbf{1}}(\mathbf{x}_1 - \mathbf{t}_1)$$

$$\mathbf{x}_2 = \mathbf{R}_2\mathbf{R}_1{}^{-\mathbf{1}}(\mathbf{x}_1 - \mathbf{t}_1) + \mathbf{t}_2$$

$$\mathbf{x}_2 = \mathbf{R}_2\mathbf{R}_1{}^{-\mathbf{1}}\mathbf{x}_1 - \mathbf{R}_2\mathbf{R}_1{}^{-\mathbf{1}}\mathbf{t}_1 + \mathbf{t}_2$$

Now we can see a relative translation between frame 1 and frame 2

$$\mathbf{R}_{\text{rel}} = \mathbf{R}_2\mathbf{R}_{\mathbf{1}}{}^{-1}$$

$$\mathbf{t}_{\text{rel}} = -\mathbf{R}_2\mathbf{R}_1{}^{-\mathbf{1}}\mathbf{t}_1 + \mathbf{t}_2$$

We can now define the essential matrix and the fundamental matrix

$$\mathbf{E} = \mathbf{t}_{\text{rel}\times}\mathbf{R}_{\text{rel}}$$

$$\mathbf{E} = \mathbf{K}^{\mathrm{T}}\mathbf{F}\mathbf{K}$$

$$\mathbf{F} = \mathbf{K}^{-\mathrm{T}}\mathbf{E}\mathbf{K}^{-\mathbf{1}}$$

$$\mathbf{F} = \mathbf{K}^{-\mathrm{T}}\mathbf{t}_{\text{rel}\times}\mathbf{R}_{\text{rel}}\mathbf{K}^{-\mathbf{1}}$$

Q1.4

Let's start with the following notation: **P** is the object point in 3D coordinates, **P'** is the reflected object point in 3D coordinates, $\tilde{\mathbf{p}}$ is the object point in 2D camera coordinates, $\widetilde{\mathbf{p}'}$ is the reflected object point in 2D camera coordinates, and **K** is the intrinsics matrix of the camera. We can start by defining the projections of the 3D points:

$$\lambda_1\tilde{\mathbf{p}} = \mathbf{KP}, \qquad \lambda_2\widetilde{\mathbf{p}'} = \mathbf{KP}'$$

Now we define the transformation between **P** and **P'** and substitute the above values of the 3D points:

$$\mathbf{P}' = \mathbf{RP} + \mathbf{t}$$

$$\mathbf{R} = \mathbb{I}_3, \mathbf{t} = \begin{bmatrix} t_x \\ t_y \\ t_z \end{bmatrix}$$

$$\lambda_2\mathbf{K}^{-1}\widetilde{\mathbf{p}'} = \lambda_1\mathbf{K}^{-1}\tilde{\mathbf{p}} + \mathbf{t}$$

Now we take the cross product of both sides with respect to $\mathbf{t}$, which is the same as premultiplying by:

$$\mathbf{t}_\times = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}$$

$$\lambda_2\mathbf{t}_\times\mathbf{K}^{-1}\widetilde{\mathbf{p}'} = \lambda_1\mathbf{t}_\times\mathbf{K}^{-1}\tilde{\mathbf{p}}$$

Now, taking the dot product of both sides with respect to $\widetilde{\mathbf{p}'}$. This causes the left-hand side to be zero because $\mathbf{t}_\times\widetilde{\mathbf{p}'}$ is perpendicular to $\widetilde{\mathbf{p}'}$.

$$\lambda_1\widetilde{\mathbf{p}'}^T\mathbf{K}^{-T}\mathbf{t}_\times\mathbf{K}^{-1}\tilde{\mathbf{p}} = 0$$

The fundamental matrix can now be expressed as

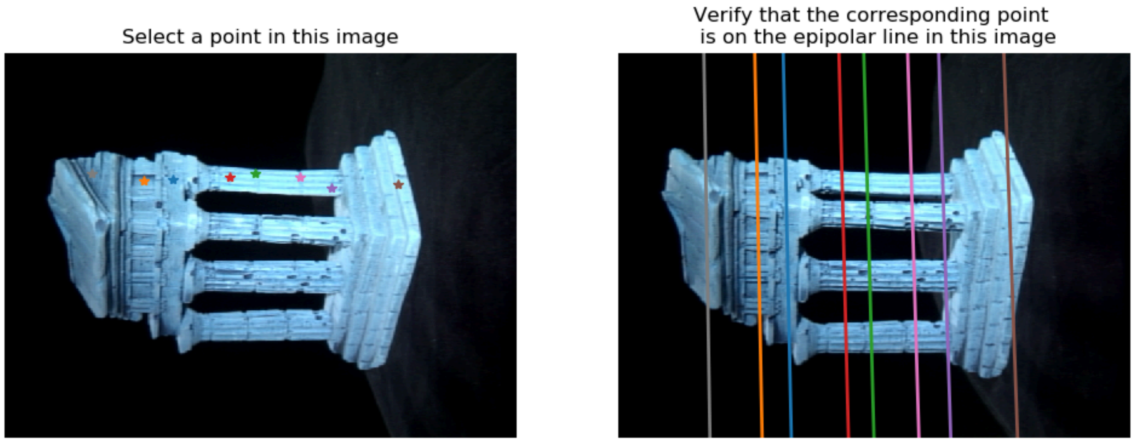
$$\mathbf{F} = \mathbf{K}^{-T}\mathbf{t}_\times\mathbf{K}^{-1}$$

Because $\mathbf{t}_\times$ is skew-symmetric: $-\mathbf{t}_\times = \mathbf{t}_\times{}^T$

$$\mathbf{F}^T = \mathbf{K}^{-T}\mathbf{t}_\times{}^T\mathbf{K}^{-1} = -\mathbf{K}^{-T}\mathbf{t}_\times\mathbf{K}^{-1} = -\mathbf{F}^T$$

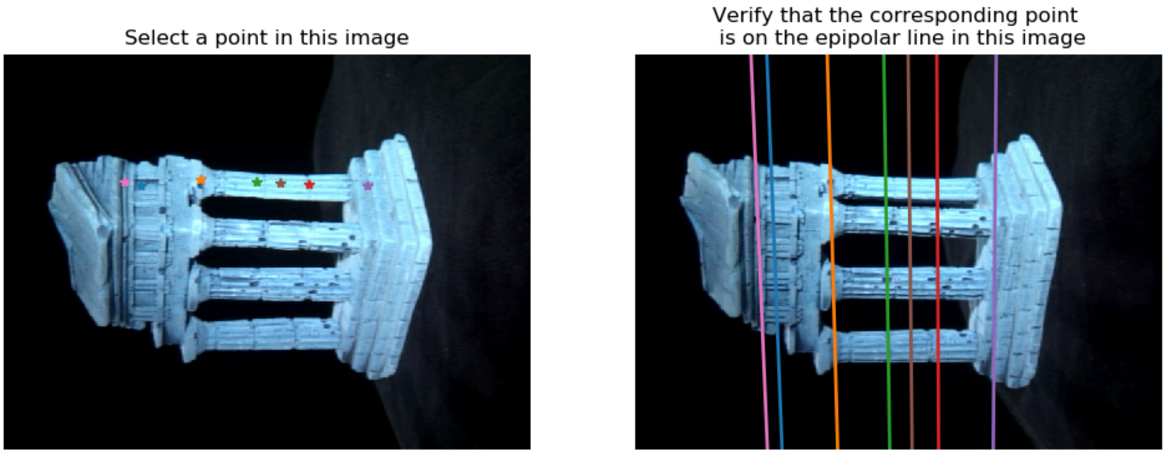
So, F is also skew-symmetric.

Q2.1 8 Point Algorithm



$$\mathbf{F} = \begin{bmatrix} 9.78833282e-10 & -1.32135929e-07 & 1.12585666e-03 \\ -5.73843315e-08 & 2.96800276e-09 & -1.17611996e-05 \\ -1.08269003e-03 & 3.04846703e-05 & -4.47032655e-03 \end{bmatrix}$$

Q2.2 7 Point Algorithm



$$\mathbf{F} = \begin{bmatrix} 1.55556901e-08 & 4.41605611e-07 & -1.53948843e-03 \\ -2.44876595e-07 & -1.13314839e-08 & 9.88640680e-05 \\ 1.47710494e-03 & -1.04658477e-04 & 5.30282941e-03 \end{bmatrix}$$

Q3.1 Essential Matrix calculated using results of 8 Point Algorithm

$$\mathbf{E} = \begin{bmatrix} 2.26268683e-03 & -3.06552495e-01 & 1.66260633e+00 \\ -1.33130407e-01 & 6.91061098e-03 & -4.33003420e-02 \\ -1.66721070e+00 & -1.33210351e-02 & -6.72186431e-04 \end{bmatrix}$$

Q3.2 Triangulation of a 3D point

We start with two corresponding image points and the two associated camera matrices

$$\tilde{\mathbf{x}}_{\mathbf{1}\boldsymbol{i}} = \begin{bmatrix} x_{1i} \\ y_{1i} \\ 1 \end{bmatrix}, \qquad \tilde{\mathbf{x}}_{\mathbf{2}\boldsymbol{i}} = \begin{bmatrix} x_{2i} \\ y_{2i} \\ 1 \end{bmatrix}$$

$$\mathbf{C1} = \begin{bmatrix} C1_{11} & C1_{12} & C1_{13} & C1_{14} \\ C1_{21} & C1_{22} & C1_{23} & C1_{24} \\ C1_{31} & C1_{32} & C1_{33} & C1_{34} \end{bmatrix}, \qquad \mathbf{C2} = \begin{bmatrix} C2_{11} & C2_{12} & C2_{13} & C2_{14} \\ C2_{21} & C2_{22} & C2_{23} & C2_{24} \\ C2_{31} & C2_{32} & C2_{33} & C2_{34} \end{bmatrix}$$

We can now set up the relation to find the 3D projected point $\widetilde{\mathbf{w}}_{\boldsymbol{i}}$

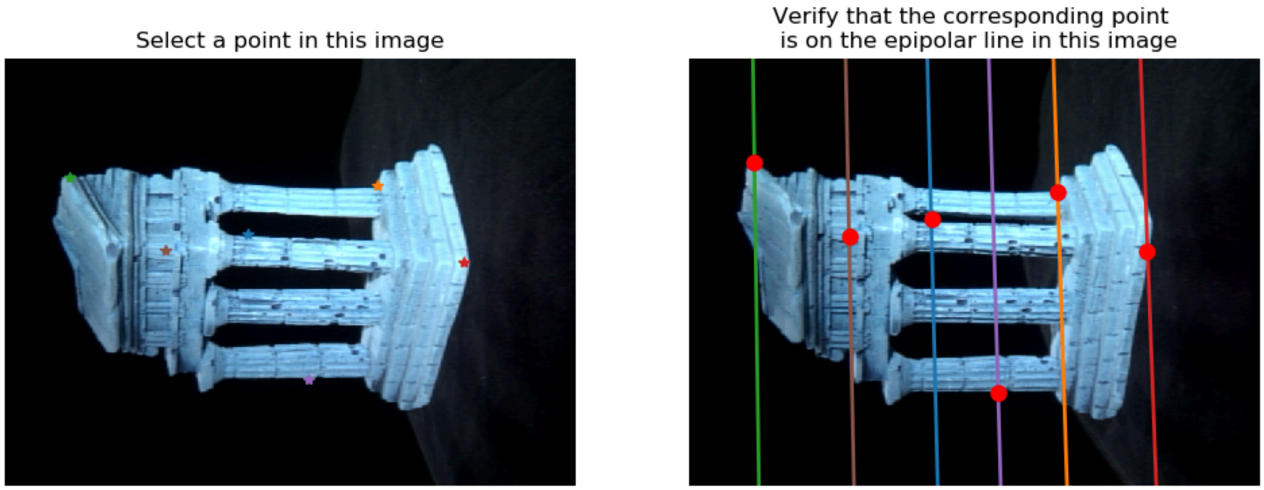
$$\lambda\tilde{\mathbf{x}}_{\mathbf{1}\boldsymbol{i}} = \mathbf{C1}\widetilde{\mathbf{w}}_{\boldsymbol{i}}, \qquad \lambda\tilde{\mathbf{x}}_{\mathbf{2}\boldsymbol{i}} = \mathbf{C2}\widetilde{\mathbf{w}}_{\boldsymbol{i}}$$

We can then set up the following equation and solve for $\widetilde{\mathbf{w}}_{\boldsymbol{i}}$ using least squares

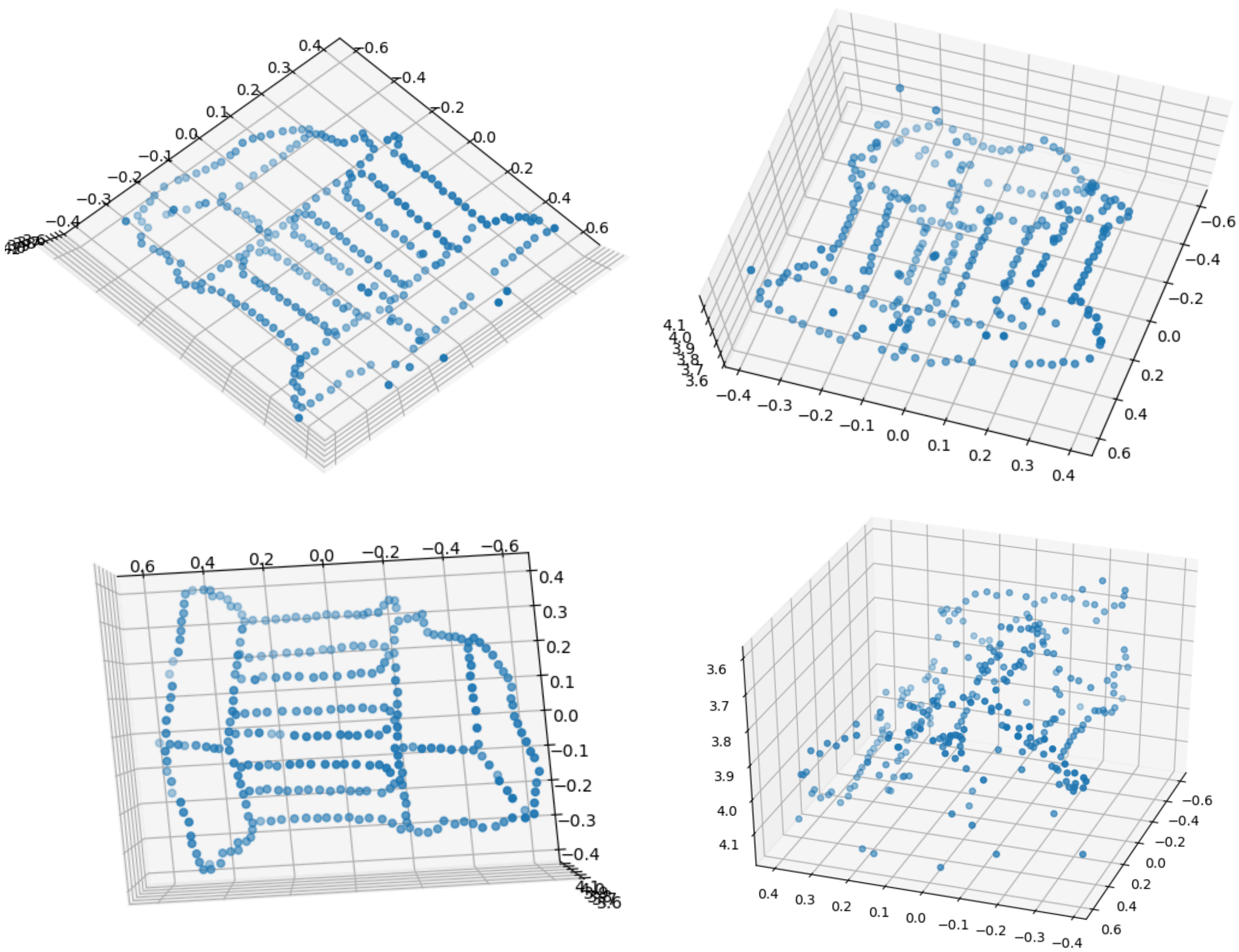
$$\mathbf{A}_{\boldsymbol{i}}\widetilde{\mathbf{w}}_{\boldsymbol{i}} = \mathbf{0}$$

$$\mathbf{A}_{\boldsymbol{i}} = \begin{bmatrix} C1_{31}x_{1i} - C1_{11} & C1_{32}x_{1i} - C1_{12} & C1_{33}x_{1i} - C1_{13} & C1_{34}x_{1i} - C1_{14} \\ C1_{31}y_{1i} - C1_{21} & C1_{32}y_{1i} - C1_{22} & C1_{33}y_{1i} - C1_{23} & C1_{34}y_{1i} - C1_{24} \\ C2_{31}x_{2i} - C2_{11} & C2_{32}x_{2i} - C2_{12} & C2_{33}x_{2i} - C2_{13} & C2_{34}x_{2i} - C2_{14} \\ C2_{31}y_{2i} - C2_{21} & C2_{32}y_{2i} - C2_{22} & C2_{33}y_{2i} - C2_{23} & C2_{33}y_{2i} - C2_{24} \end{bmatrix}$$

Q4.1 epipolarMatchGUI Results



Q4.2 Point Cloud from visualize.py



Q5.1 ransacF Results

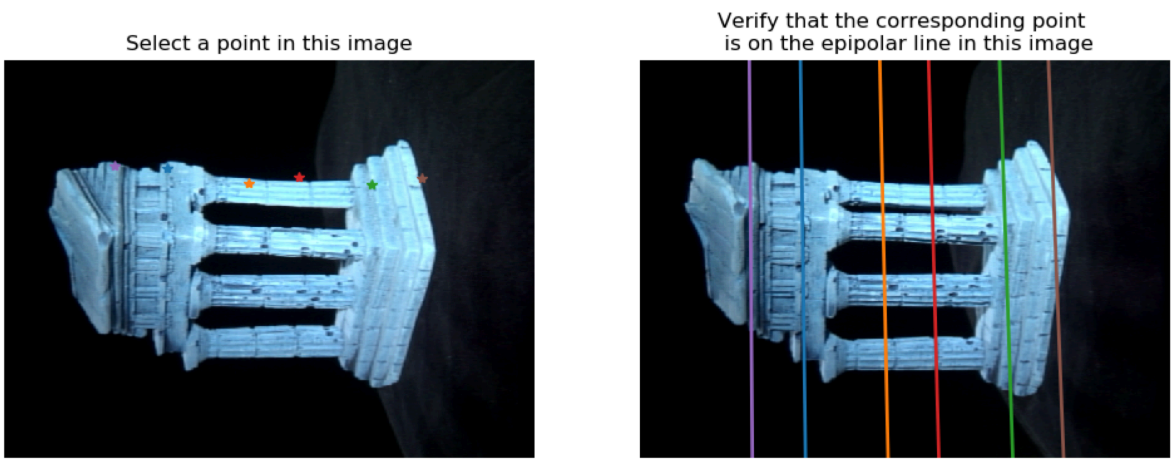
For my RANSAC implementation, I used the following error metric, discussed in lecture:

$$\hat{\mathbf{F}} = \underset{\mathbf{F}}{\operatorname{argmin}} \left[\sum_{i=1}^{I} \left((\text{dist}[\mathbf{x}_{i1}, \mathbf{l}_{i1}])^2 + (\text{dist}[\mathbf{x}_{i2}, \mathbf{l}_{i2}])^2 \right) \right]$$

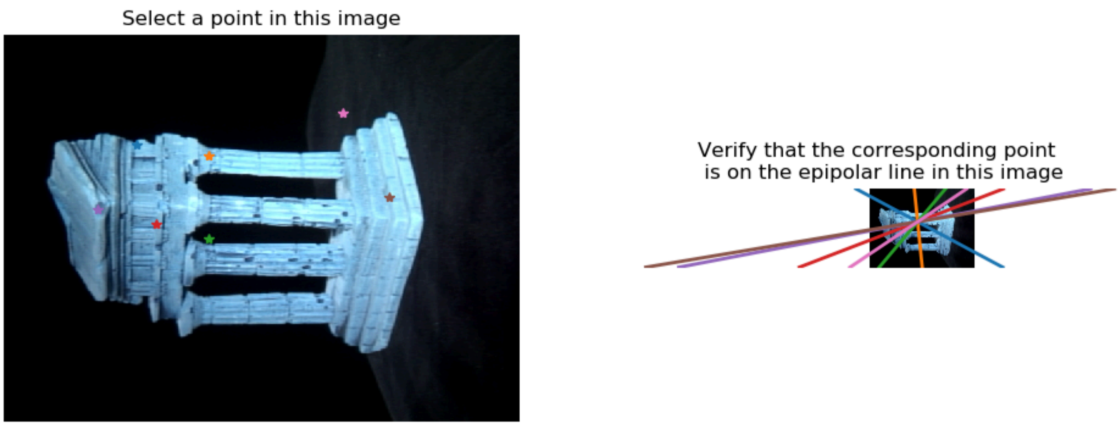
If $\mathbf{l} = [a, b, c]^T$ and $\mathbf{x} = [x, y]^T$ $\quad \text{dist}[\mathbf{x}, \mathbf{l}] = \dfrac{ax + by + c}{\sqrt{a^2 + b^2}}$

This computes the distance between the points in image 1 and the corresponding epipolar lines on image 1 created by the points in image 2, and adds it to the distance between the points in image 2 and the corresponding epipolar lines on image 2 created by the points in image 1. An inlier is found if this distance metric is less than one, proving that the accuracy of the computed epipolar lines is very high.

Here is the result of the displayEpipolarF GUI using my implementation of the 7 point algorithm with RANSAC on the noisy correspondence data:



For comparison, here is the result of the displayEpipolarF GUI using my implementation of the 8 point algorithm on the noisy correspondence data:



Clearly, the 8 point algorithm does not perform well when there is noise in the data, proving the fact that RANSAC is the preferred method when there is uncertainty in the correspondences.

Q5.3 Results of Bundle Adjustment

The images below show one test of bundle adjustment. I used test_M2_solution to get an initial set of 3D points **P_init** and an initial extrinsic matrix **M2_init**. **P_init** and **M2_init** were then input into bundleAdjustment. The resulting **M2** from bundleAdjustment was then used to triangulate the 2D points into 3D space. These reprojected points are shown in red. The initial points before bundle adjustment (**P_init**) are shown in blue.

The error results vary greatly from run to run due to the fact that RANSAC is random, but I consistently see 3D points that are close together, and reprojection errors that are much smaller after bundle adjustment. In the given examples, here are the reprojection errors before and after bundle adjustment:

Error BEFORE bundle adjustment: 5511.71664

Error AFTER bundle adjustment: 6.501739