Introduction

We have learned the concept of fundamental matrix and reconstruction in two-view geometry in the class. This purpose of this project is to test your knowledge in real problems.

1.Use the images in the project 1. Select a set of corresponding wall and floor points (>20) in both first image and the second image. Please classify the points into two groups: those belong to wall plane and those belong to floor plane.

2. Compute fundamental matrix F using any algorithm that we learned from the class. However, it has to use all points that you have. Note that we do have a lot noise in the system.

3. Identify plane at infinity and reconstruct the wall plane and the floor plane up to affinity in 3D.

You need to submit a project report that includes the following items,

- The original images with marked corresponding points (I do not care whether they are color or grayscale images.) [10pts]
- The methods you used to compute F and to reconstruct the two planes. [30pts]
- The plane at infinity and the homography matrix used for transformation to affinity. [10pts]
- The camera matrices P and P'. [10pts]
- The 3D floor plane and the wall plane in both numerical and graphical illustration (no rendering is required.) [20pts]
- Discussions about pros and cons of your approach.[10pts]
- Source code. [20pts]

Input Image:

Theory and Implementation

According to the textbook and lecture notes, I am going to use the Linear Least Squares estimation of the Fundamental Matrix. The Fundamental Matrix F is estimated using the manual correspondences (x_i, x'_i) that are selected by the user. (x_i, x'_i) are the pixel locations in homogeneous coordinates in the left and right image of the same world point selected by the user manually. The Fundamental Matrix F gives us the algebraic representation of the epipolar geometry which finds the relation between the corresponding points (x_i, x'_i). The algebraic equation is given by:

$$x'_i F x_i = 0$$

The above equation can be expanded in details such as:

$$[u'_i u_i \quad u'_i v_i \quad u'_i \; v'_i \quad v'_i u_i \quad v'_i v_i \quad v'_i \; u_i \quad v_i \quad 1\,] f \; = \; 0$$

From a set of n point matches, we obtain a set of linear equations of the form:

$$\begin{bmatrix} u'_1 u_1 & u'_1 v_1 & u'_1 & v'_1 u_1 & v'_1 v_1 & v'_1 & u_1 & v_1 & 1 \\ . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . \\ u'_n u_n & u'_n v_n & u'_n & v'_n u_n & v'_n v_n & v'_n & u_n & v_n & 1 \end{bmatrix} \mathbf{f} = 0$$

Where

$$f \; = \; [f_{11} \; f_{12} \; f_{13} \; f_{21} \; f_{22} \; f_{23} \; f_{31} \; \; f_{32} \; f_{33}\,] \quad f_{33} \; = \; 1$$
$$x_i \; = \; [u_i \; v_i \; 1]^T, \; x'_i \; = \; [u'_i \; \; v'_i \; \; 1]^T$$

Each correspondence point gives one 1 equation. Since there are 8 unknowns, we need a minimum of 8 such correspondences to use the normalized 8 point algorithm, which normalizes the data to improve the estimate of the Fundamental Matrix F. The steps are listed below:

1. Normalization
   The image coordinates are transformed according to
   $$\hat{x}_i = T x_i \quad \hat{x'}_i = T' x'_i$$
   Here, T and T' are the normalization transformation consisting of a translation and scaling. The normalizing transformations T, for points xi and T' , for points x'i for the two images are found such that all the pixel correspondences have 0 mean and are at a

distance of $\sqrt{2}$ from the center C1 = (0,0). Here, the normalization process is like this, first we assume the normalization matrix T1 for data Xi, and we have

$$T_1 = \begin{bmatrix} s & 0 & t_x \\ 0 & s & t_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$\widehat{X_\iota} = T_1\vec{X} = \begin{bmatrix} sx_i + t_x \\ sy_i + t_y \\ 1 \end{bmatrix} = \begin{bmatrix} \hat{x} \\ \hat{y} \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} s\overline{x_i} + t_x \\ s\overline{y_i} + t_y \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

$$\frac{1}{n}\Sigma_i^m \sqrt{\hat{x}_i^2 + \hat{y}_i^2}$$
$$= \frac{1}{n}\Sigma_i^m \sqrt{(sx_i - s\overline{x})^2 + (sy_i - s\overline{y})^2}$$
$$= \frac{s}{n}\Sigma_i^m \sqrt{(x_i - \overline{x})^2 + (y_i - \overline{y})^2}$$

$$s = \frac{\sqrt{2}}{\frac{1}{m}\Sigma_i^m \sqrt{(x_i-\overline{x})^2+(y_i-\overline{y})^2}}$$

In this way, we are able to normalize the image data to be centered around (0,0) and with average distance $\sqrt{2}$.

2. Direct Linear Solution

   When we have normalized all the points, we stack up all of them and make a matrix vector equation of the form Af = 0.The matrix Aˆhas all the stacked rows similar to the matrix shown above and is composed of the matches between xi and x'i . This is solved using SVD to yield the matrix F from f corresponding to the smallest singular value of A, or we can say it is the null space of A. In this assignment, we have used 20 points and the matrix A is the size 20 x 20.

3. Constraint enforcement
   The rank of the fundamental matrix is 2, but the fundamental matrix we just solved has rank 3. So the way we did that is to SVD decomposition F again and set the smallest eigenvalue to be zero, and then recompute the F to get the initial estimation of F. More specifically, we assume the smallest component of D is σ_3

$$F = UDV$$
$$D = diag[\sigma_1\ \sigma_2\ \sigma_3]$$
$$D' = diag[\sigma_1\ \sigma_2\ 0]$$
$$F = UD'V$$

In the way, the rank of new F is 2 and det(F) = 0.

4. Denormalization

   After we obtained the F from step 3 ,we need to find the F for the original data points, so we have apply transformation again.

   $$new\ F = T'FT$$

5. Compute epipoles

   In this step, we calculate the epipoles in the two images. The epipoles are very important to the calculation of projective matrix P for step 7. We have learned that the epipoles are the left and right null space of fundamental matrix F. Here we denote the epipole in image 1 as e_1 and epipole in image 2 as e_2, then we can calculate the epipoles below:

   $$Fe_1 = 0$$
   $$e_2F = 0$$

6. Calculate the P1, P2

   In this step, we fixed the coordinate to the image 1, and I can set the projective matrix $P_1 = [I|0]$ and then I use the method proposed in the textbook and lecture notes to find estimation for P_2, there are many P_2 s that satisfy the projective requirement, so we can select one by utilization of the epipole. Here e1x is the cross product operator for e1, $P_2 = [e_{1x}F|e_2]$, or more specifically,

   $$e_x = \begin{bmatrix} 0 & -e_3 & e_2 \\ e_3 & 0 & -e_1 \\ -e_2 & e_1 & 0 \end{bmatrix}$$

7. Calculate the initial reconstructed 3D points through P_1 and P_2

   Here, the linear triangulation methods is used to calculate the 3D world coordinates. Since the relationship x = PX, x′ = P’X is valid in each image, and these equations can be combined into a form AX = 0, which is an equation linear in X. Then, $x \times PX = 0$ leads to the below equations

$$x(P^{3T}X) - (P^{1T}X) = 0$$
$$x(P^{3T}X) - (P^{2T}X) = 0$$
$$x(P^{2T}X) - (P^{1T}X) = 0$$ --- this one is not required derived from above

$$x'(P'^{3T}X) - (P'^{1T}X) = 0$$
$$x'(P'^{3T}X) - (P'^{2T}X) = 0$$
$$x'(P'^{2T}X) - (P'^{1T}X) = 0$$ --- this one is not required derived from above

And equation of the form Ax = 0 can then be composed with

$$A = \begin{bmatrix} xP^{3T} - P^{1T} \\ xP^{3T} - P^{2T} \\ x'P'^{3T} - P'^{1T} \\ x'P'^{3T} - P'^{2T} \end{bmatrix} X = 0$$

Here, I did not normalize the data point x and x' because the two images have also similar sizes, so the unit scale in these two images should be same. After the initial world point X is obtained, the geometric error can be calculated for each point correspondence from the world point X by re-projection x = PX

$$error = |x - \hat{x}| + |x' - \hat{x}'|$$

8. Refine F, P2, and 3D points
   The points calculated from the above step may not be the best solution for this assignment, because F is obtained from linear system equation. Here, we can further optimize the above F, P2, X through levenberg marquardt optimization, and the cost function I am going to use is:

   $$cost = \sum_{i}^{n}(x - P_1X)^2 + (x' - P_2X)^2$$

   The objective is to minimize the above error given initial F, fixed P1, initial P2, initial world point X

9. Projective to affine transformation
   The above calculation only reconstruct the world projectively and further transformation is required to obtain the affine transformation. Here, I used plane at infinity to transform

the projective world point to affine world point. So I find 3 intersections of sets of lines in the scene that are supposed to be parallel, and use these 3 points to define plane π, and then find a transformation H that maps the plane p to the plane at infinity (0, 0, 0,1)T.

Suppose the plane at infinity is $\pi_\infty = (a\ b\ c\ d)$, then the transformation H can be constructed as:

$$H = \begin{bmatrix} I|0 \\ \pi_\infty \end{bmatrix}$$

Then, I can apply the transformation to scene points X, P, H such as

$$X' = HX$$
$$P_1' = P_1H^{-1}$$
$$P_2' = P_2H^{-1}$$

In this process, I have selected four groups of parallel line and each parallel group have one intersection points. Since all the points are in the 3D world, and the line equation has four parameters. Here, we can use the Plu ¨cker matrices the line is represented by a 4×4 skew-symmetric homogeneous matrix. In particular, the line joining the two points A, B is represented by the matrix L with elements

$$L = AB^T - BA^T$$

Any point on the line L must satisfy the equation as $L^*X = 0$, here L^* is the dual Plu ¨cker representation and skewed.

$$l_{12} : l_{13} : l_{14} : l_{23} : l_{42} : l_{34} = l^*{}_{34} : l^*{}_{42} : l^*{}_{23} : l^*{}_{14} : l^*{}_{13} : l^*{}_{12}$$

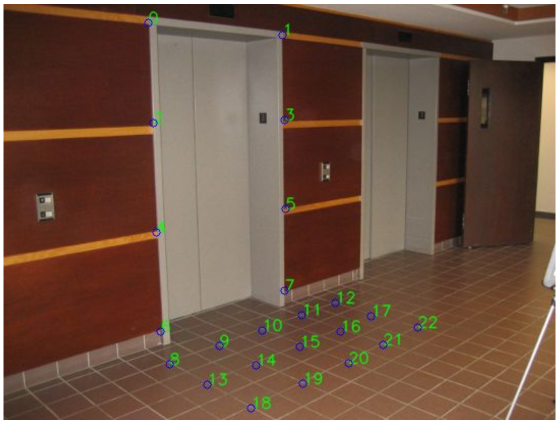
To calculate the intersection point of two parallel lines $L^*{}_1$ and $L^*{}_2$, we can use the equation $L^*{}_1X = 0$ and. Once we have the intersection points X, then we can calculate the plane for X.

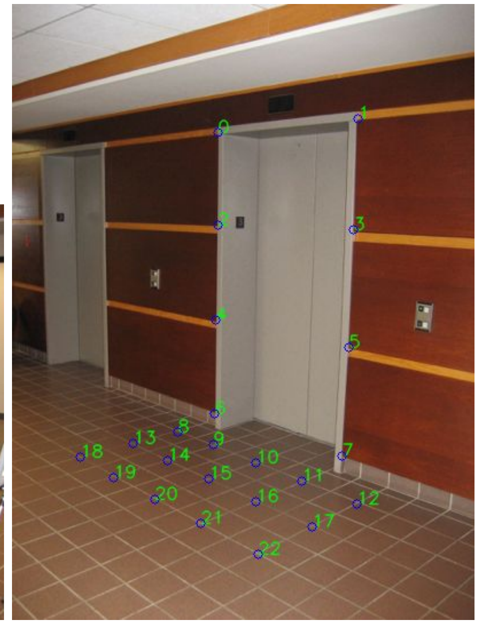
$$\pi_\infty X_1 = 0,\ \pi_\infty X_2\ ,\ \pi_\infty X_3 = 0$$

$$\begin{bmatrix} X_1^T \\ X_2^T \\ X_3^T \end{bmatrix} \pi_\infty = 0$$

Result & Analysis:

1. Marked Points on both images, points correspondence.





Coordinates in Image 1:
pts1 = {142, 18}, {274, 30}, {147, 116}, {276, 113}, {150, 223}, {277, 200},{154, 320}, {276, 280},{163, 352}, {212, 334}, {254, 319}, {293, 304}, {326, 292}, {200, 372}, {248, 353}, {291, 335}, {331, 320}, {361, 305}, {243, 395}, {294, 371}, {339, 351}, {373, 333}, {407, 316}}

Coordinates in Image 2:
pts2 = {{179, 112}, {301, 100}, {179, 193}, {297, 197},{177, 276}, {293, 300}, {176, 358}, {287, 395},{144, 374}, {175, 385}, {212, 401}, {252, 417}, {300, 437}, {105, 384}, {135, 399}, {171, 415}, {212, 435}, {261, 457}, {59, 396}, {88, 414}, {124, 433}, {164, 454}, {214, 481

2. Initial F and refined F

   Initial F is :
   [-2.4104600749374e-06, 7.83320474598814e-06, 0.0006148789914631087;
   1.228264473631073e-05, 4.383624970217529e-07, -0.009161926692898344;
   -0.002238722505791311, 0.004404124015552122, 1]

   Refined F is:
   [-2.41209816505011e-06, 7.841347262974762e-06, 0.0006187480860522574;
   1.228391774295272e-05, 4.339562708971694e-07, -0.009159249977839137;
   -0.002235241925806804, 0.00440010965893482, 1]

   Epipole in image 1:
   [0.9802519767955422;

0.197748097483116;
0.001323605016643468]

Epipoles in image 2 is:
[0.9921964408638136;
-0.1246721786062537;
-0.001752318107926059]

3. The plane at infinity and the homography matrix used for transformation to affinity

   In this step, I have selected 4 group of parallel lines, 2 group on the wall and 2 groups on the floor because I want the elevator is area is parallel horizontally and vertically. Also the parallel line on the floor should also parallel with each other. The parallel group I selected from wall and floor are:

   Group 1: (point1, point2) || (point7, point8)
   Group 2: (point1, point7) || (point2, point8)
   Group 3: (point9, point13) || (point19, point23)
   Group 4: (point9, point19) || (point13, point23)

   Then, I calculated the intersection points of each parallel group p1, p2, p3, p4 and then find the plane at infinity for these 4 points:

p1 = -0.9954 -0.0961 -0.0011 0.0020	p2 = -0.9992 -0.0398 -0.0010 0.0019	p3 = -0.9643 0.2647 0.0030 -0.0106	p4 = -0.0668 -0.9978 -0.0002 -0.0005

   Plane at infinity:
   [a, b, c, d] = [0.0004 0.0003 -0.9544 -0.2986]

   Homography matrix used for transformation to affinity H:

   $$H = \begin{bmatrix} 1.0000 & 0 & 0 & 0 \\ 0 & 1.0000 & 0 & 0 \\ 0 & 0 & 1.0000 & 0 \\ 0.0004 & 0.0003 & -0.9544 & -0.2986 \end{bmatrix}$$

4. The camera matrices P and P'.
   P =

[1, 0, 0, 0;
 0, 1, 0, 0;
 0, 0, 1, 0]

P' =

[0.0002791279351926387, -0.000549070967720464, -0.1246882332163012, 0.9921964408638136;
 0.002221256726220695, -0.004369769899620181, -0.9921975183274045, -0.1246721786062537;
 1.188627908277632e-05, 1.411524410504345e-06, -0.009013772752703986, -0.001752318107926059]

5. The 3D floor plane and the wall plane in both numerical and graphical illustration.

   Wall plane equation [a, b, c, d] = [-0.0025   0.0023   -0.5886   0.8084]

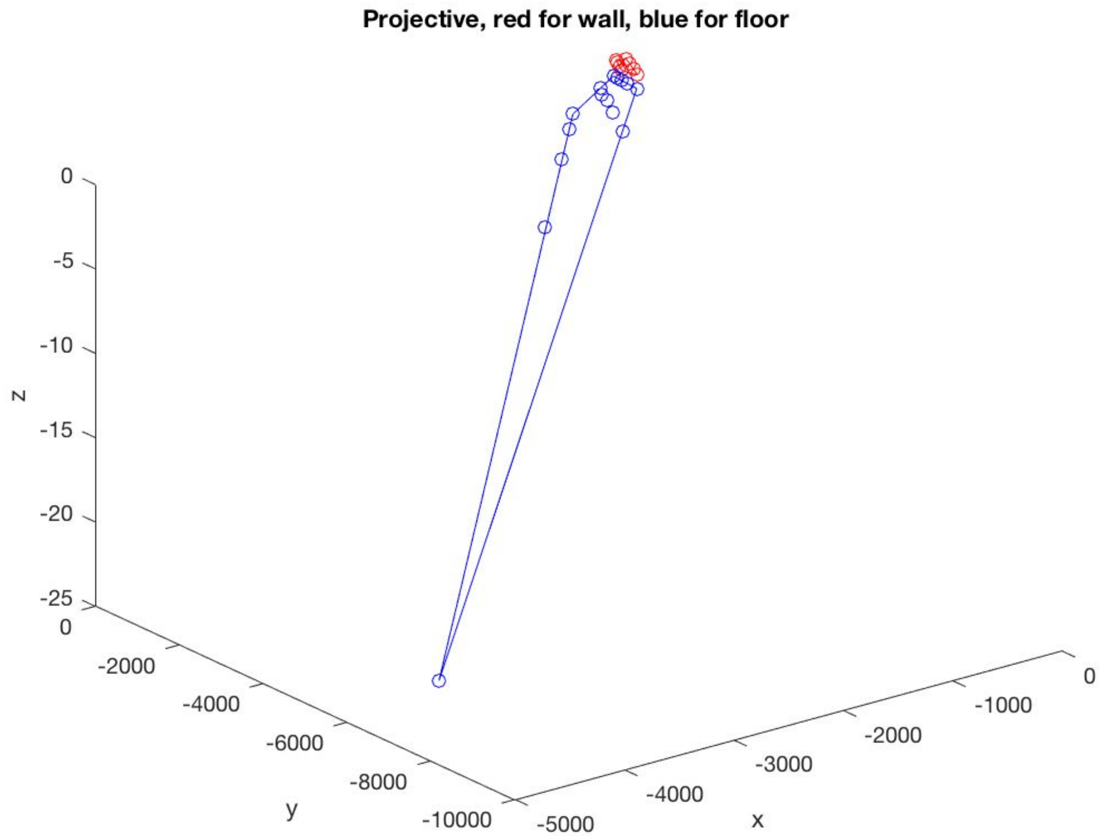
   Wall Plane Points:

   [-153.04413, -19.39994, -1.0777756;
    -251.51949, -27.538635, -0.91795433;
    -170.42828, -134.4877, -1.159376;
    -270.10541, -110.58663, -0.97864276;
    -191.99736, -285.43604, -1.2799823;
    -290.4317, -209.69798, -1.0484899;
    -215.7438, -448.29883, -1.4009337;
    -310.06503, -314.55872, -1.1234239]

   Floor plane equation [a, b, c, d] = [-0.0013   0.0012   0.9194   0.3934]

   Floor Plane Points:

   [-299.80396, -647.42944, -1.839288;
    -342.51065, -539.61578, -1.6156162;
    -365.94406, -459.59119, -1.4407246;
    -390.0806, -404.72525, -1.3313332;
    -403.74139, -361.63342, -1.2384706;
    -637.23395, -1185.255, -3.1861694;
    -605.32056, -861.60553, -2.4408088;
    -585.87091, -674.45624, -2.0133021;
    -584.94153, -565.50238, -1.767195;
    -562.13422, -474.93335, -1.5571585;
    -4938.166, -8027.0596, -20.321671;
    -1719.9468, -2170.4092, -5.8501587;
    -1234.7599, -1278.4681, -3.6423593;
    -1013.6945, -904.9873, -2.7176797;
    -905.10718, -702.73682, -2.2238505]



Points after homograph H, X' = HX,

$$H = \begin{bmatrix} 1.0000 & 0 & 0 & 0 \\ 0 & 1.0000 & 0 & 0 \\ 0 & 0 & 1.0000 & 0 \\ 0.0004 & 0.0003 & -0.9544 & -0.2986 \end{bmatrix}$$

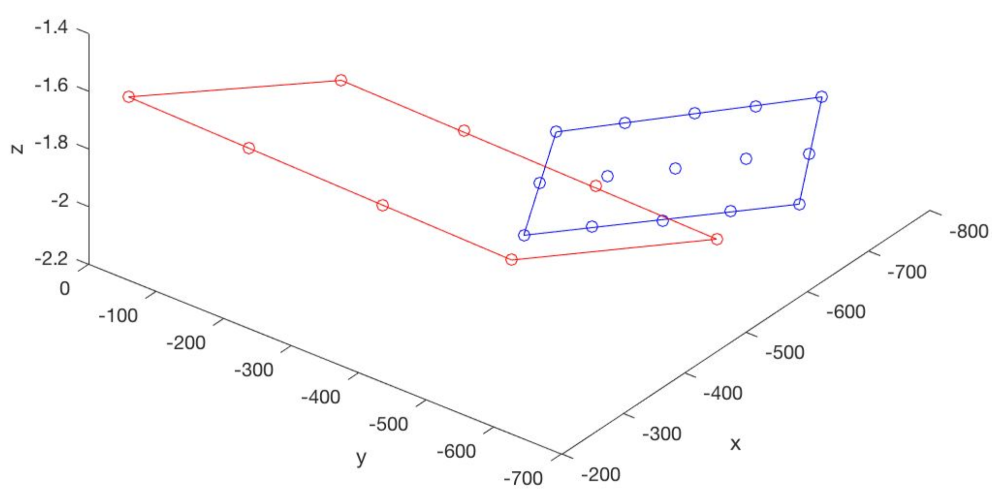
Wall Plane Points =

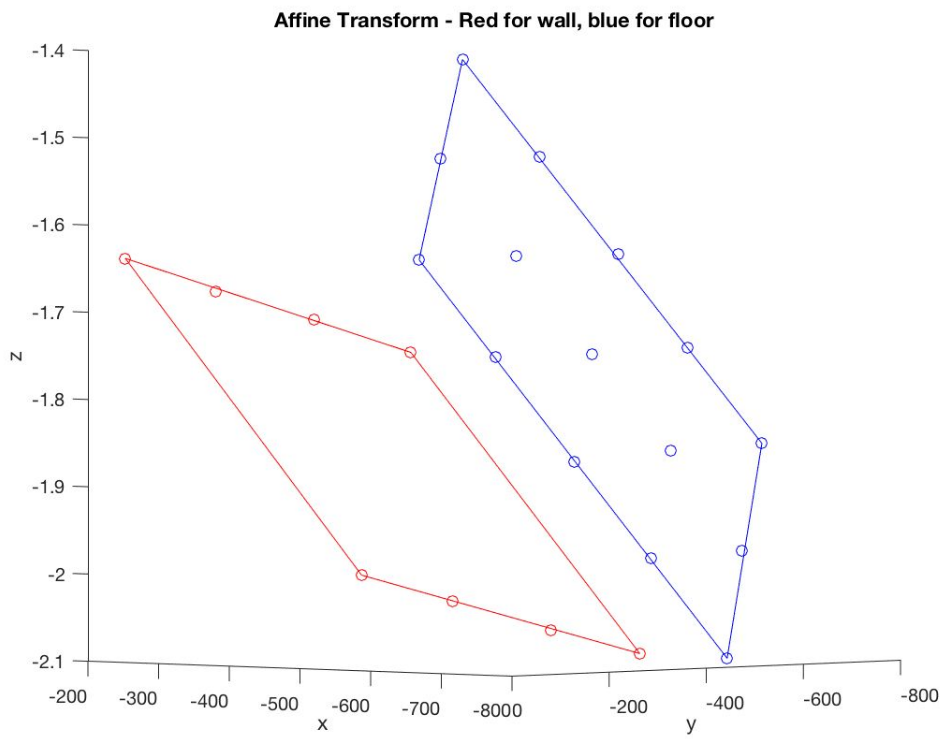
-232.6024	-29.4848	-1.6380
-546.1254	-59.7948	-1.9932
-246.8997	-194.8325	-1.6796
-559.2229	-228.9572	-2.0262
-257.3218	-382.5517	-1.7155
-571.5759	-412.6902	-2.0635
-270.5418	-562.1648	-1.7568
-577.9459	-586.3219	-2.0940

Floor Plane Points =

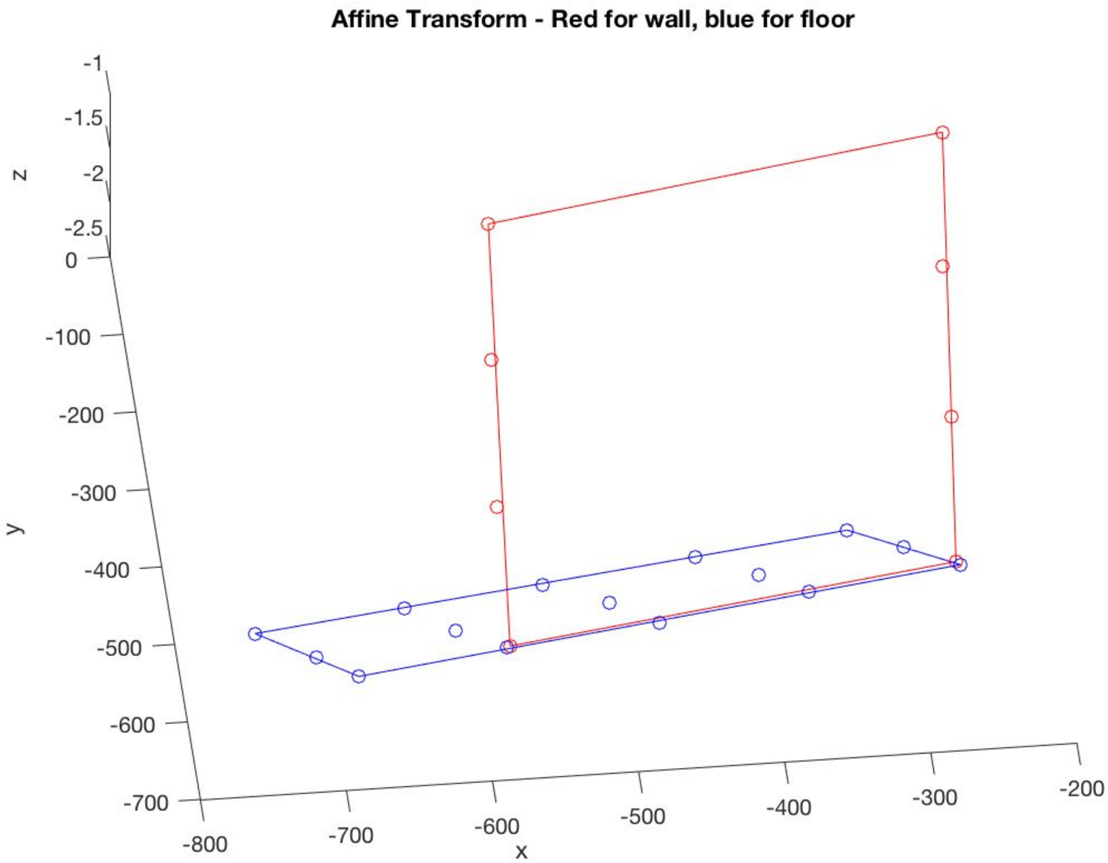
-269.1646	-581.2634	-1.6513
-373.1942	-587.9568	-1.7603
-476.9008	-598.9424	-1.8776
-581.5405	-603.3731	-1.9848
-683.6505	-612.3496	-2.0971
-306.8719	-570.7816	-1.5344
-407.3286	-579.7864	-1.6425
-510.1638	-587.3020	-1.7531
-615.7227	-595.2606	-1.8602
-712.1697	-601.6946	-1.9728
-344.8122	-560.4972	-1.4190
-449.2118	-566.8626	-1.5279
-554.9341	-574.5777	-1.6370
-649.3927	-579.7528	-1.7410
-752.2464	-584.0538	-1.8483

Affine Transform - Red for wall, blue for floor





Affine Transform - Red for wall, blue for floor
-1.4
-1.5
-1.6
-1.7
-1.8
-1.9
-2
-2.1
z
-200
-300
-400
-500
-600
-700
-8000
x
-200
-400
-600
-800
y



Discussions about pros and cons of your approach.

Pros: Easy to implement by manually select points because we can pinpoint the matching points correctly without making mistake.

Cons:Accuracy is downgraded by the manual selection process because we can not select points very accurately.