Report for HW3

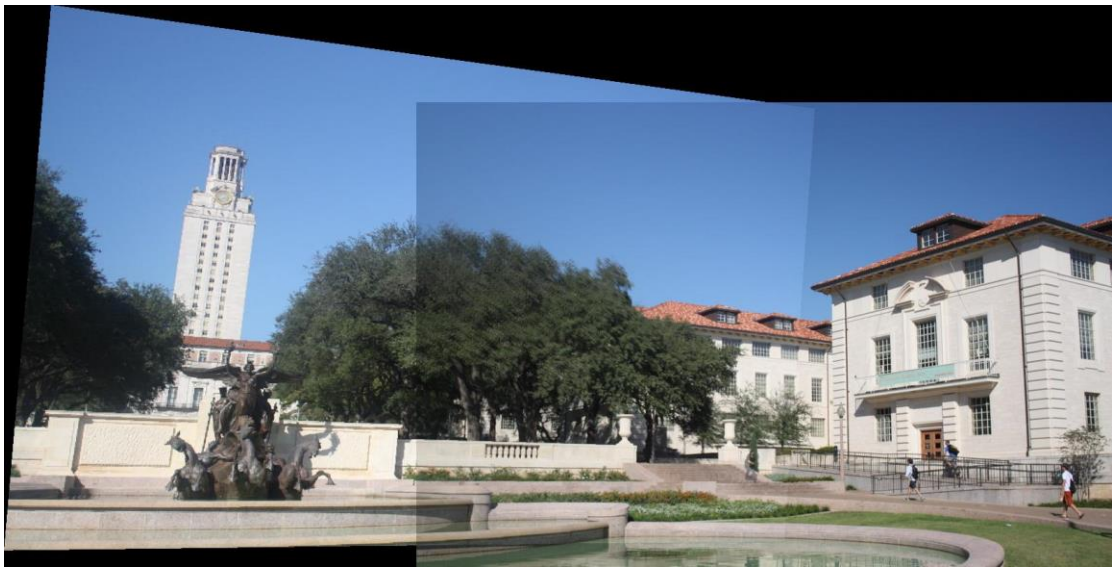
Part1

Interesting points

1. Histogram equalization benefits features detection and matching in some cases like uttower, but somehow also do harm to the system.
2. The neighbours of certain neighbours of a point are used to generate more features to generate more accurate matches. See more in 'get_features.m'.
3. Features are fuzzied to reduce affection of slightly malposed intensity.
4. Intensity is adjusted when compute distances of features.
5. It's important to choose a proper number of pairs so that many of them are correct.
6. When generate matches, no point is allowed to appear in matches for more than 2 times. The points that are similar to many other points may confuse the system.
7. When two mappings have the same number of inliers, the program would try to choose the one whose four points used for estimation form a larger quadrangle.

Example

The threshold for inliers is 2 pixels.
There are 24 inliers and 26 outliers.
The average residual for inliers is 1.0595 pixels.
Needs histogram equalization.

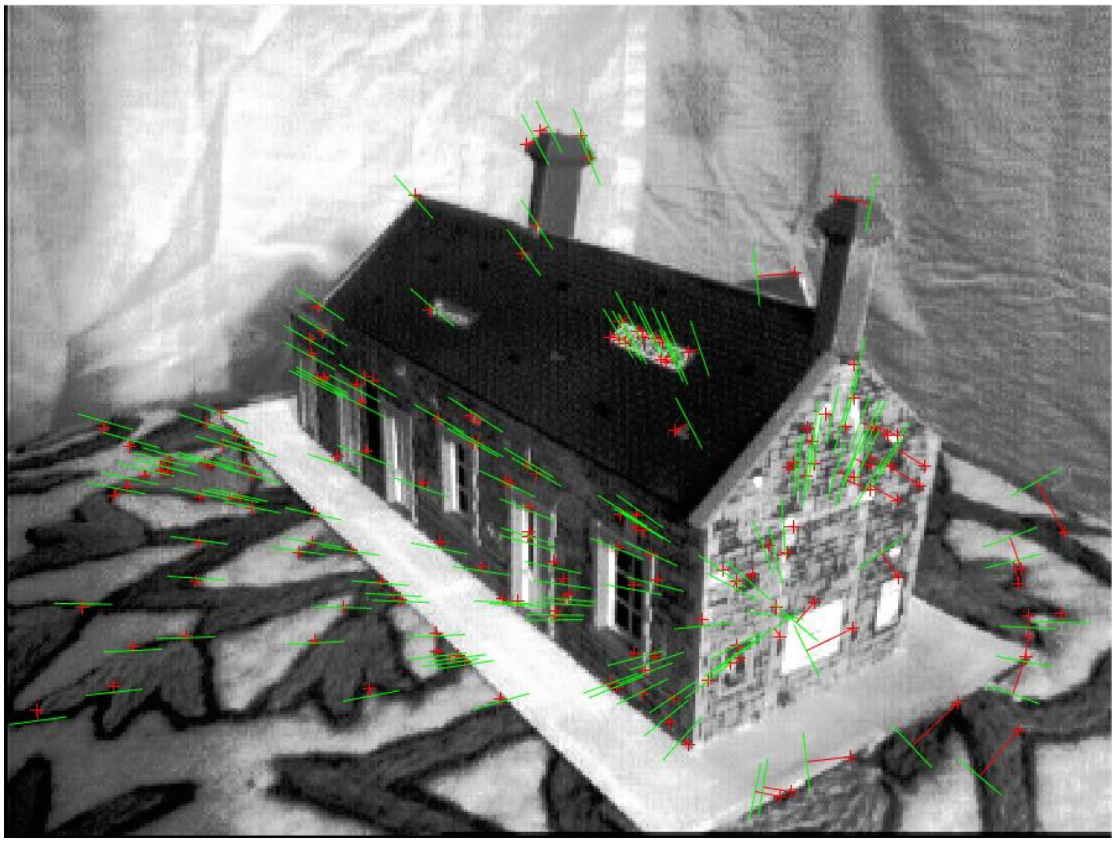


Part2

Fundamental matrix estimation

Estimation with ground truth

Unnormalized



The residual is 3.0342 pixels.

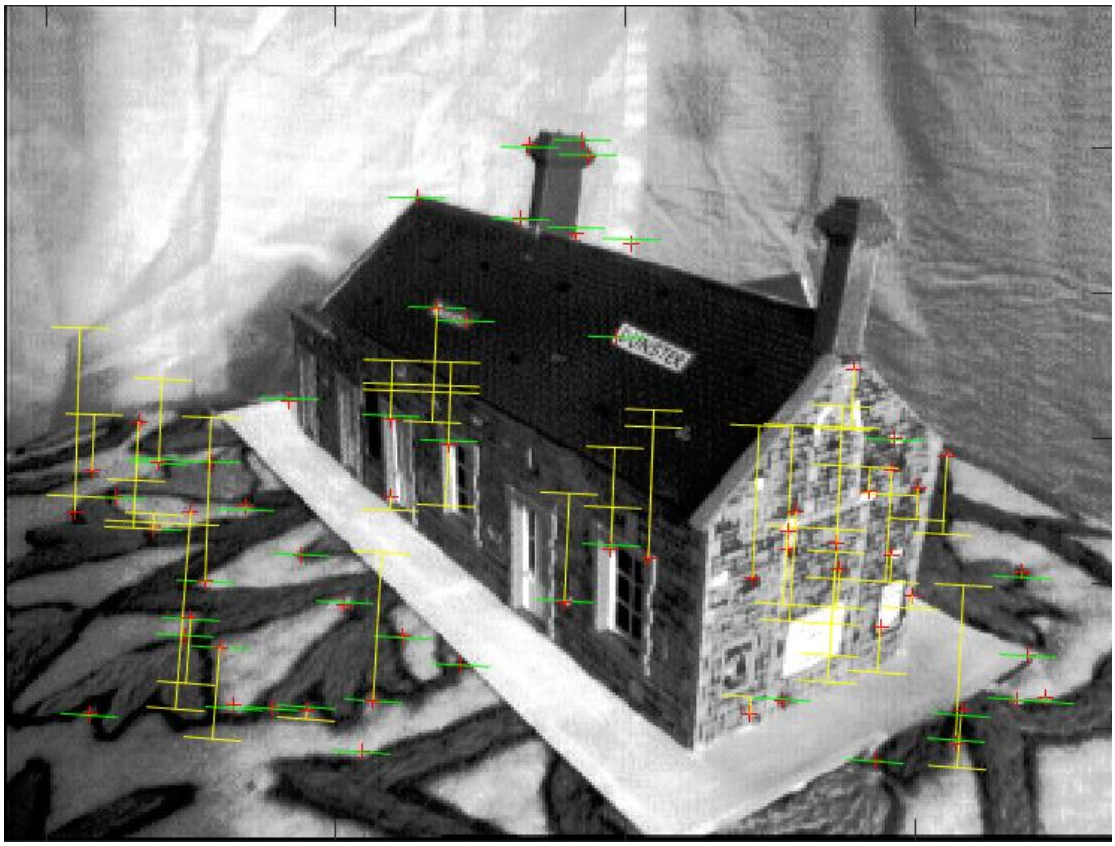
The residual is 0.3385 pixels.

Normalized

The residual is 0.2103 pixels.

The residual is 0.1836 pixels.

Estimation with RANSAC



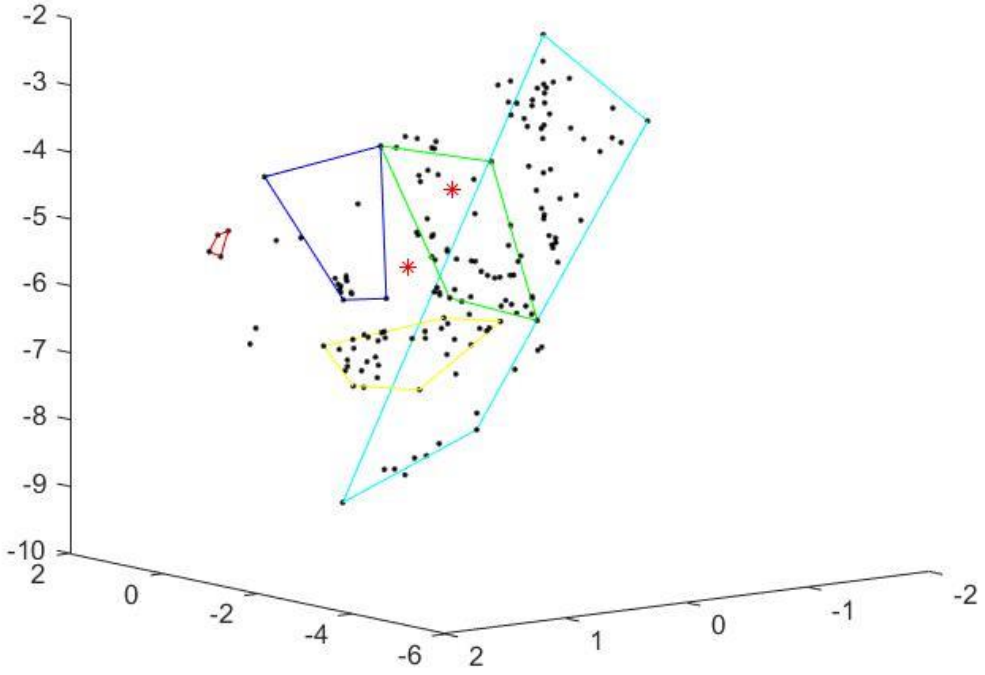
The green lines correspond to inliers and the yellow lines correspond to outliners.
The threshold for inliers to their corresponding lines is 2 pixels.
There are 44 inliers and 56 outliers.
The average residual for inliers is 0.6069 pixel.

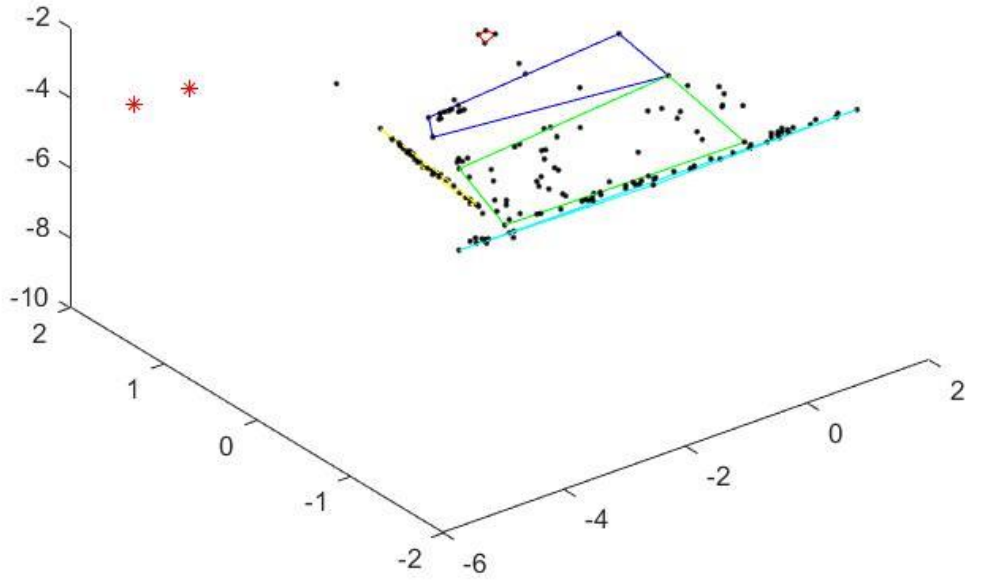
There are 28 inliers and 72 outliers.
The average residual for inliers is 0.7341 pixel.

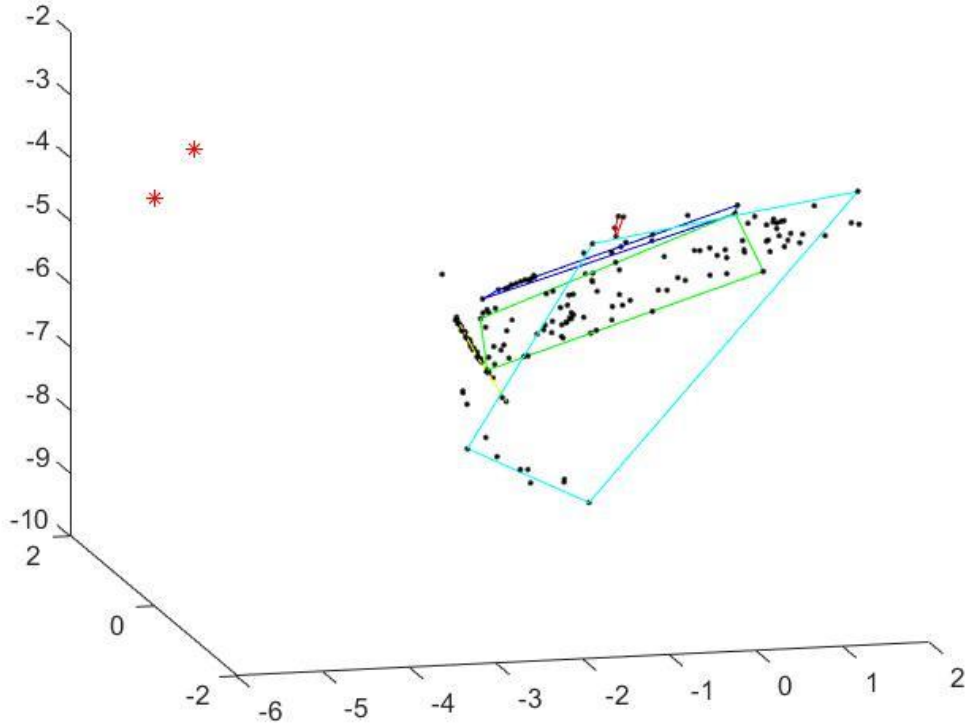
3D reconstruction

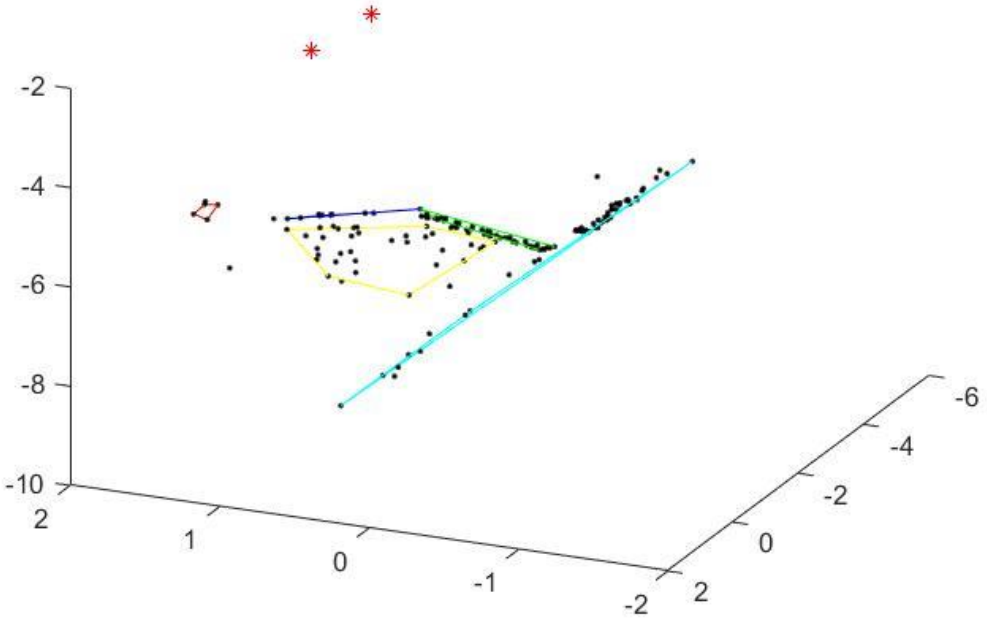
House

The two red '*' are cameras.
The red lines are from the chimney.
The green lines and yellow lines are from two walls.
The blue lines are from the roof.
The cyan lines are from the ground.
These lines chose by me may not be precisely on the objects.


-2
-3
-4
-5
-6
-7
-8
-9
-10
2
0
-2
-4
-6
2
1
0
-1
-2

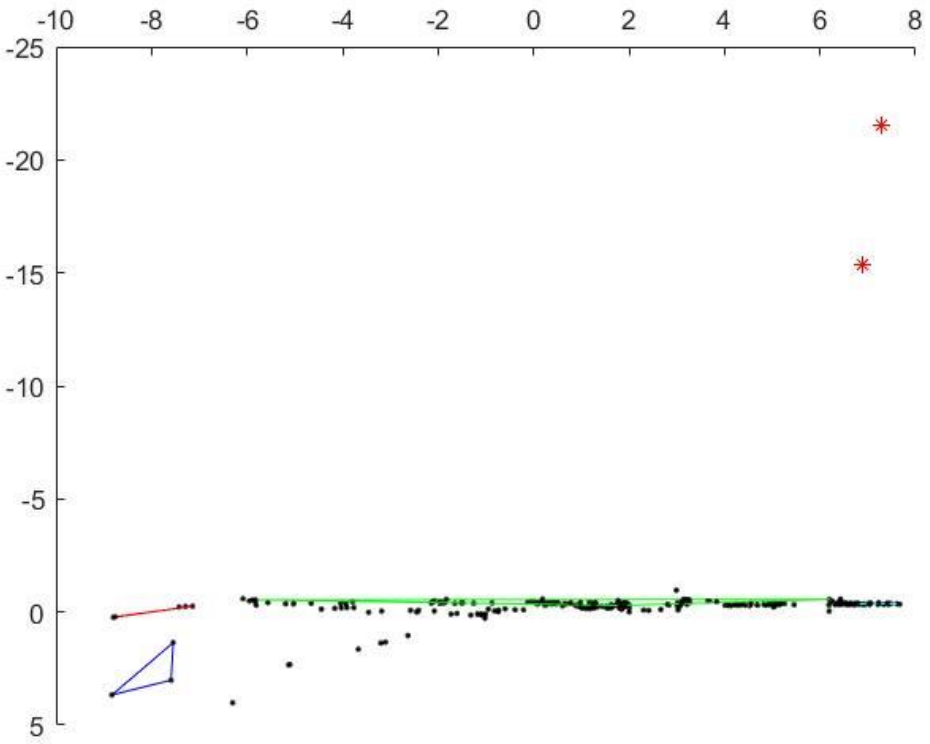

-2
-4
-6
-8
-10
2
1
0
-1
-2
-6
-4
-2
0
2

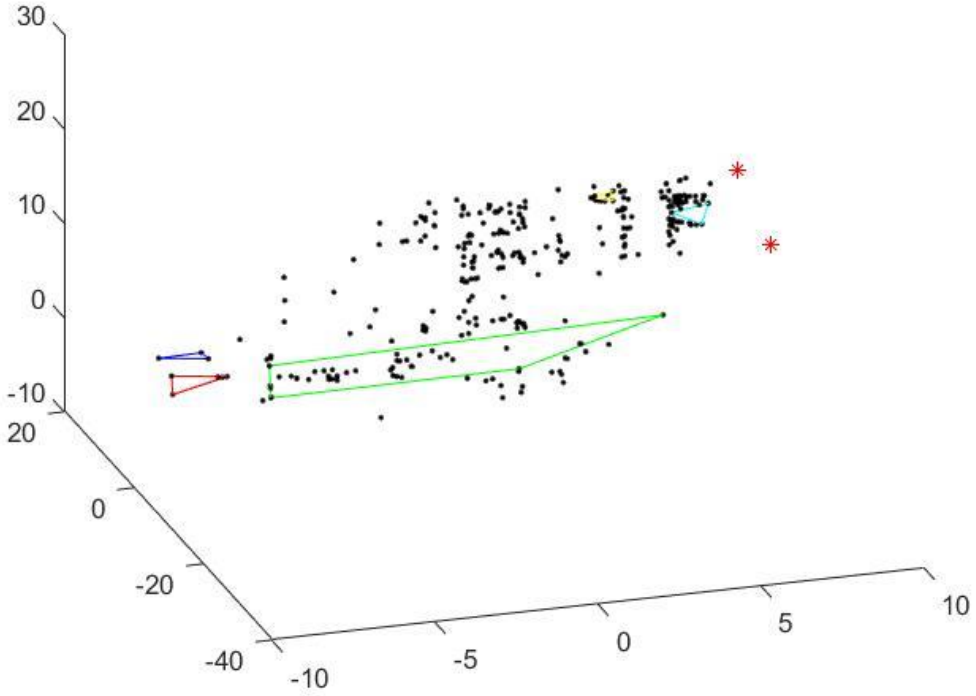

-2
-3
-4
-5
-6
-7
-8
-9
-10
2
0
-2
-6
-5
-4
-3
-2
-1
0
1
2

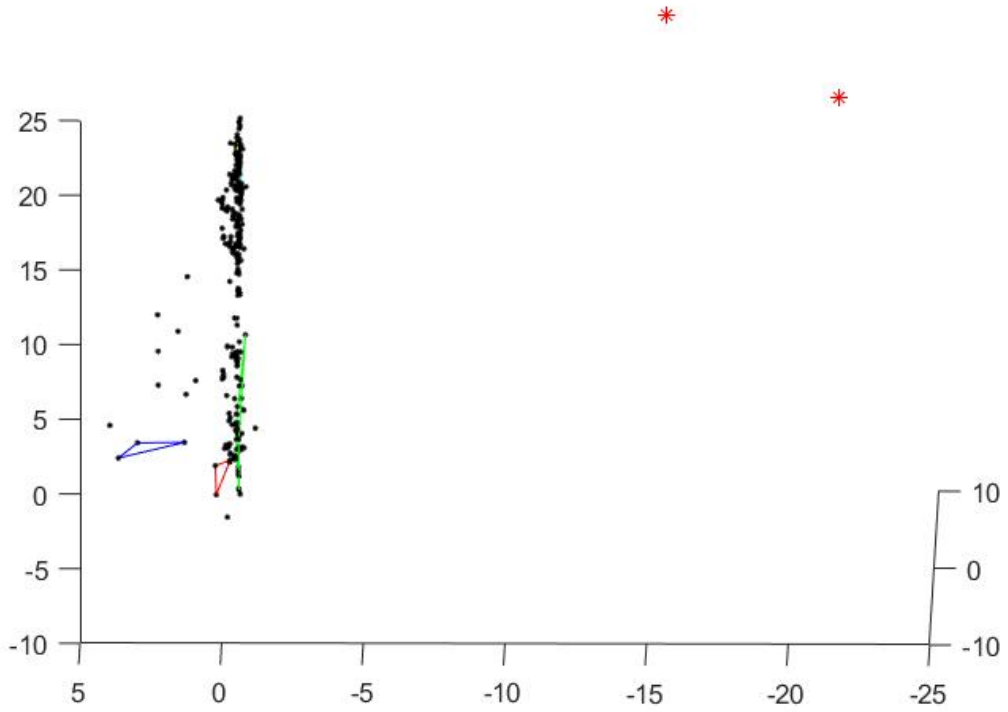

-2
-4
-6
-8
-10
2
1
0
-1
-2
-6
-4
-2
0
2

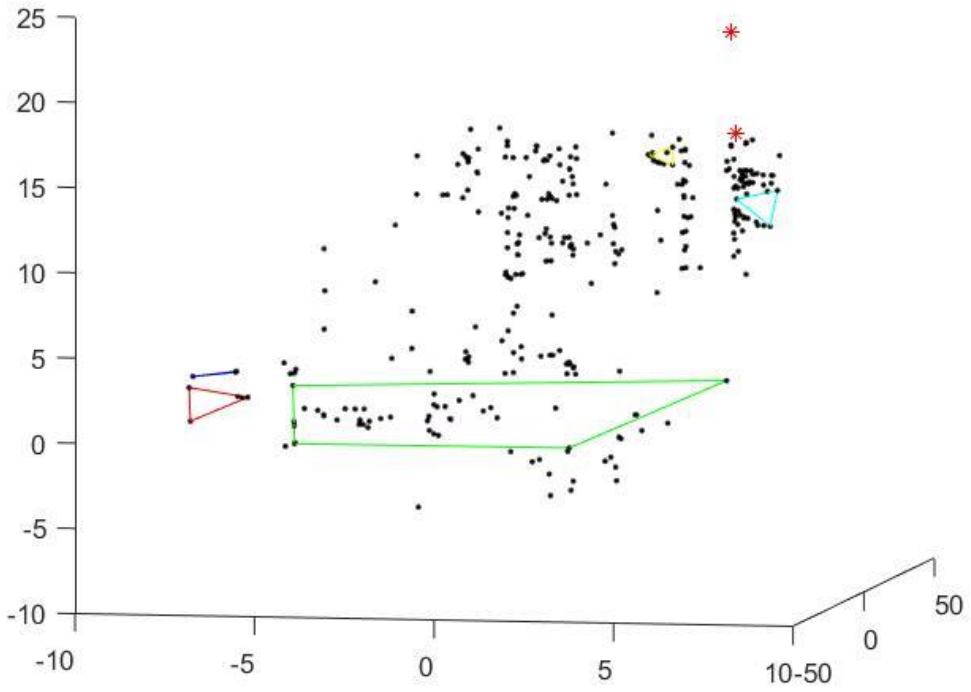
Library

The two red '*' are cameras.
The red lines and blue line are from the tower roof.
The green lines are from a tower wall.
The yellow lines are from a window.
The cyan lines are from a bush.
These lines chose by me may not be precisely on the objects.




30
20
10
0
-10
20
0
-20
-40
-10
-5
0
5
10


25
20
15
10
5
0
-5
-10
5
0
-5
-10
-15
-20
-25
10
0
-10



Extra credit

Stitch multiple images

See ‘stitch_multiple_images.m’. This program will automatically chose images pairs to stitch. The inputs are in ‘all_pic’ folder. The outputs are in ‘all_pic\output’ folder. There may be some trash outputs. There may be bug happens in imshow(). If that happened, just try again.