Assignment 2: Stereo and 3D Reconstruction from Disparity

Practical Report

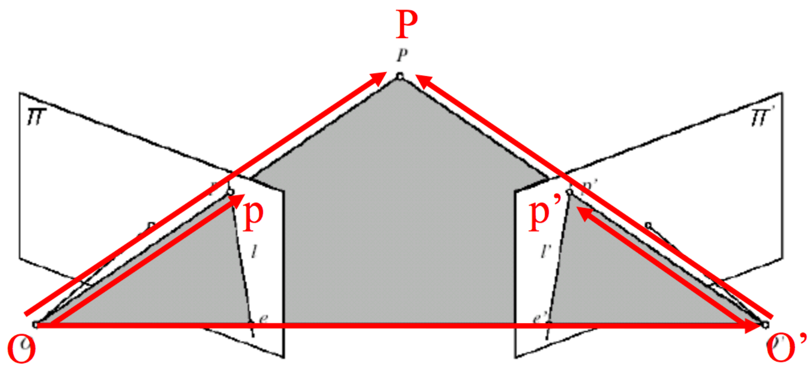
Chuan Long

Problem 4: 3D from stereo image pairs.

This objective of this assignment is to take a stereo pair of images, and explore the epiploar gemeotry and the ability to recover depth from a pair of images. For this assignment, you will need two stereo pairs of your choice, one with arbitrary translation and rotation of the two camera views, and one where you simulate a translated camera by shifting one camera horizontally and by recording the distance of the baseline.

4a. Epipolar geometry from F-matrix

According to the epipolar constraint we should have all the vectors as follow:



This leads to formulate the epipolar constraint with this equation, these vectors are coplanar:

$$\overrightarrow{Op} \cdot [\overrightarrow{OO'} \times \overrightarrow{O'p'}] = 0$$

And then, we can transform this vector into matrix form that associated with to the camera coordinates:

$$\boldsymbol{p} \cdot [\boldsymbol{t} \times (\mathcal{R}\boldsymbol{p}')] = 0 \qquad p^T[t_x]\Re p' = 0$$

And we can get the essential matrix form which is:

$$\boldsymbol{p}^T \mathcal{E} \boldsymbol{p}' = 0 \qquad \varepsilon = [t_x]\Re$$

So, we can get the fundamental matrix form based on the essential matrix and transform the camera coordinate system to image plane coordinates (assume camera calibration has not been defined):

If u and u' are corresponding image coordinates then we have:

$$\begin{array}{l} u = K_1 p \\ u' = K_2 p' \end{array} \Longrightarrow \begin{array}{l} p = K_1^{-1}u \rightarrow p^T = \left(K_1^{-1}u\right)^T = u^T K_1^{-T} \\ p' = K_2^{-1}u' \end{array}$$

$$u^T \boxed{K_1^{-T}\mathcal{E}K_2^{-1}} u' = 0$$

$$\boxed{\Rightarrow u^T F u' = 0} \qquad F = K_1^{-T}\mathcal{E}K_2^{-1}$$

- **Shoot a stereo-pair of a scene of your choice with your camera, where you take a left and a right picture where you translate and rotate the camera.**

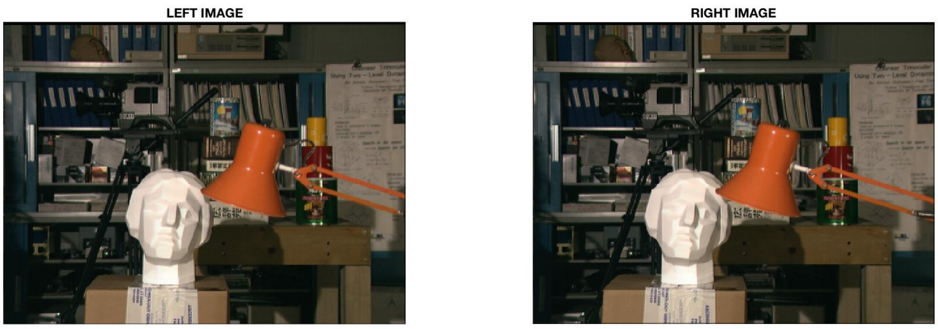


Figure 1. A Pair of Stereo images

I choose this stereo pair images because the epipole can be presented in the visible image plane so that I can verify the result easily, but I also took another pair of images that would also be presented at the following report.

- **Specify a set of corresponding pixel landmark pairs in the left and right camera views.**

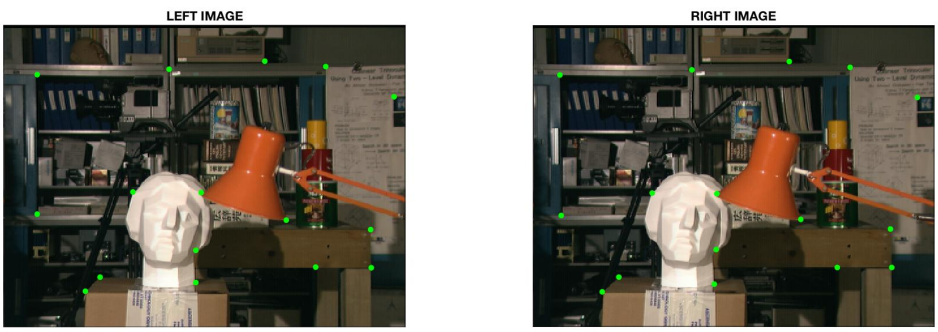


Figure 2. Stereo images with landmark points

I choose sixteen landmark points to calculate the fundamental matrix, which are marked by the green points.

- **Calculate the F-matrix using instructions as discussed in the slides.**

Because I chose sixteen points, which were more than eight points, the only solution for this situation is Least-Square-Method that we have discussed in Assignment 1.

$$\begin{bmatrix} uu' & uv' & u & u'v & vv' & v & u' & v' & 1 \end{bmatrix} \begin{bmatrix} F_{11} \\ F_{12} \\ F_{13} \\ F_{21} \\ F_{22} \\ F_{23} \\ F_{31} \\ F_{32} \\ F_{33} \end{bmatrix} = 0$$

First, I selected the sixteen pairs points, and calculate the big corresponding $16 * 9$ matrix that to be solved.

Then, I used the least-squares solution to minimize the $\sum_{i=1}^{n} (p_i^F F p_i{}')$ that we have used in Assignment 1. Now we can solve the big matrix and then get the fundamental matrix F:

$$F = \begin{bmatrix} 0.000000039806871 & -0.000874298523640 & 0.166150110467643 \\ 0.000881948924598 & -0.000028918235675 & -0.150517279539975 \\ -0.167755828037415 & 0.151578083809081 & 0.947955280037853 \end{bmatrix}$$

- **Choose a point in the left image and calculate the epipolar line in the right image. Display the point and the line as overlays in your images. Do the same for a point in the right image and epipolar line in the left image.**

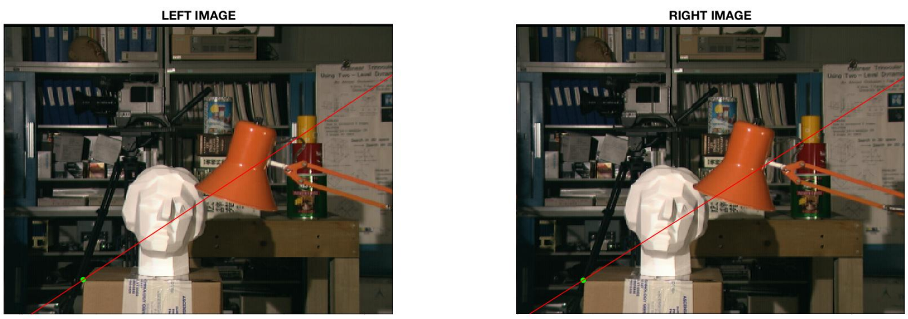


Figure 3. Stereo Images with one epipolar line

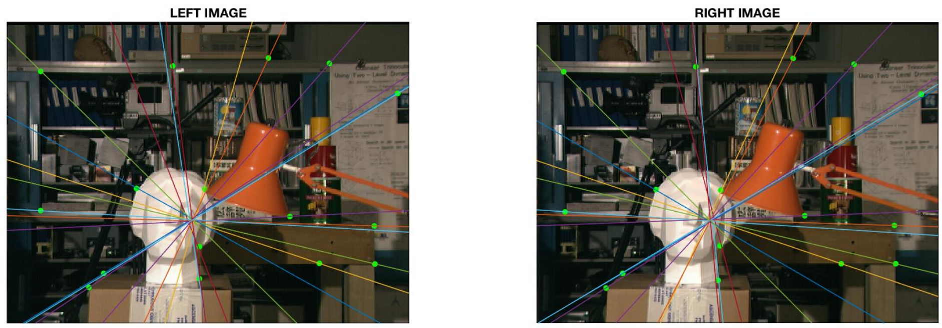


Figure 4. Stereo Images with all available epipolar lines

The equation for the epipolar lines are:

$$l' = Fx$$

$$l = F^T x'$$

Based on the fundamental matrix and the coordinates of all the points, I calculated all the epipolar lines of the corresponding points that I have selected, and presented them in the image, which are those colorful lines.

- **Calculate the position of the epipole of the left camera (see last two slides for instructions). Discuss if the calculated position seems reasonable.**

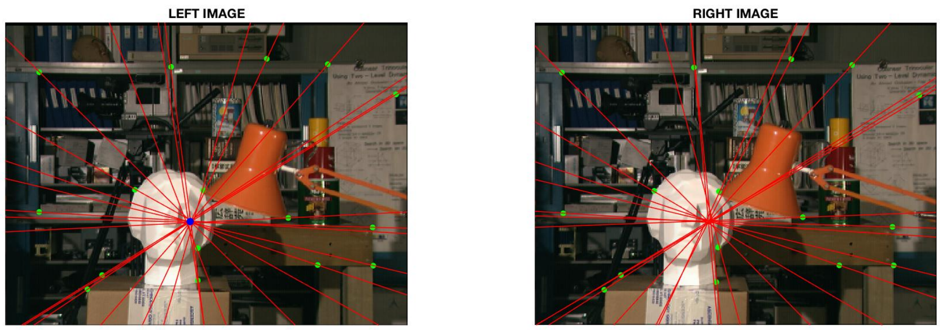


Figure 5. Stereo Images with all epipolar lines and a epipole

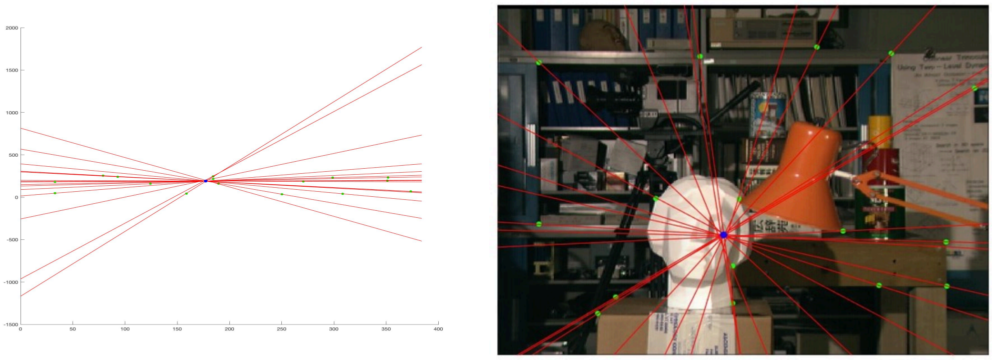


Figure 5. Stereo Images with all epipolar lines and a epipole

The equation for epipoles are:

$$Fe = 0$$
$$F^T e' = 0$$

The MATLAB method is as follow:

$$[u, d] = eigs(F' * F)$$

$$u = \begin{bmatrix} 0.682313687697575 & -0.712420649752994 & -0.164026977616666 \\ 0.731049381111298 & 0.666090449243085 & 0.147953762385191 \\ 0.003851487676813 & -0.220862697693991 & 0.975297305856019 \end{bmatrix}$$

Then, we select the eigenvector associated with smallest eigenvalue, and transform it into homogeneous coordinate.

$$e = u(:, 1)$$
$$e = e / e(3)$$

Based on the fundamental matrix and the MATLAB method, I calculated the coordinate of epipole, which is:

$$e = [177.1559 \;\; 189.8096 \;\; 1.0000]$$

The presented blue point is the epipole that I calculated based on the above method. And we can see that the blue point is almost the same as the point that the epipolar lines converge on. So we can conclude that I have got a very reasonable answer, which proves the correctness of my calculation of this part.

Discussion:

Beyond the part as required, I also used the eight-points method to calculate the fundamental matrix. But the epipole in the left image is not the same as the result that got by LSM. The following image is the result image that I used eight-points-method.

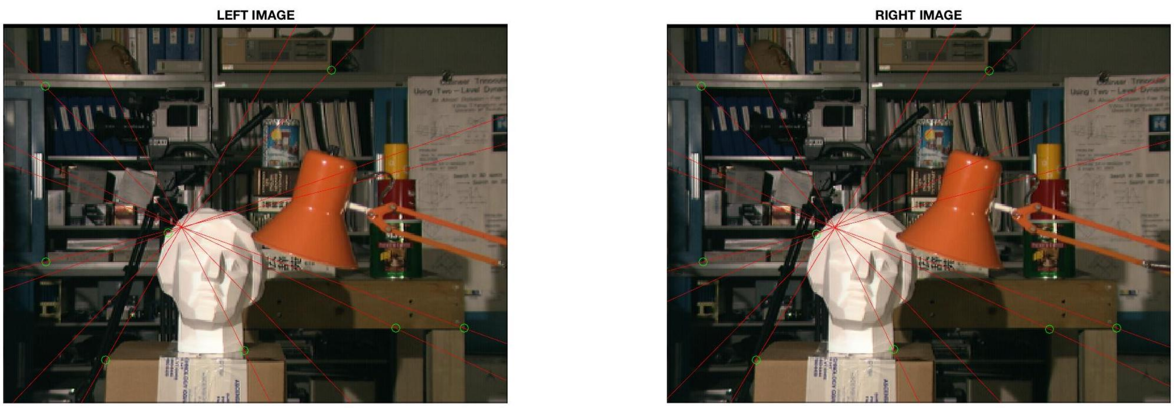


Figure 6. Stereo images with eight points algorithm

What’s more, when I choose almost the same points (not the exact SAME points as before), I get a total different result. The position that the epipolar lines that converge on is shifted right a lot, which is also the position of the epipole point. The reason why I get these different results is that the magnitudes of the elements in the fundamental matrix matter a lot in this case.

$$F = \begin{pmatrix} -0.00310695 & -0.0025646 & 2.96584 \\ -0.028094 & -0.00771621 & 56.3813 \\ 13.1905 & -29.2007 & -9999.79 \end{pmatrix}$$

For example, the magnitude of F(1, 1) differs a lot from the magnitude of F(3, 3). So, even slight change may cause a very different result.

But it is possible that the epipole would not be presented in the visible image plane, which is as follow.

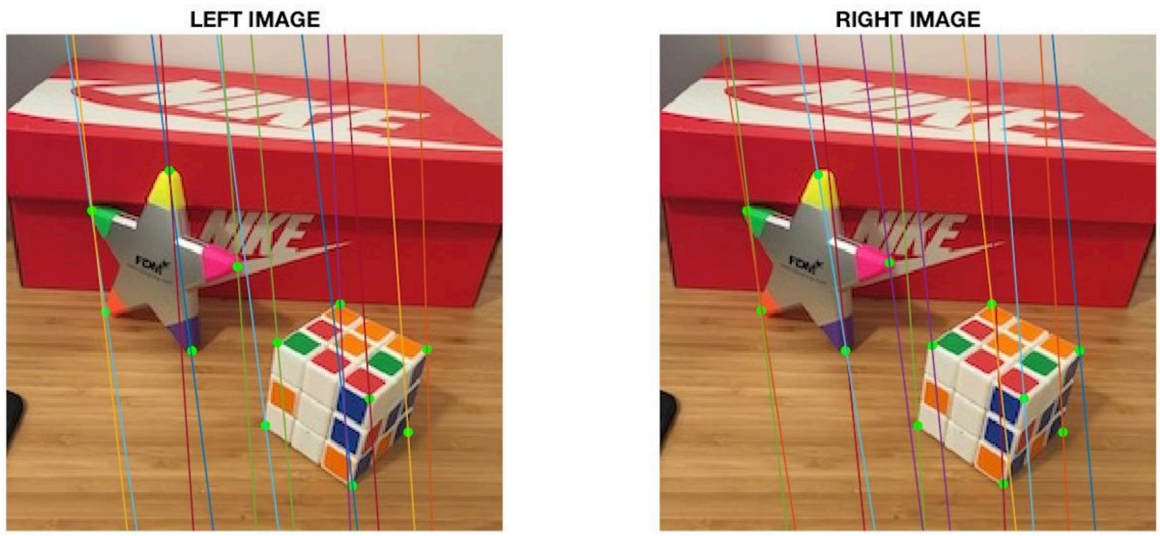


Figure. Stereo Pair Images Without Visible Epipole

The epipole in the left image is

$$e = [93.214606858498 \;\; 1146.6071557431 \;\; 1]$$

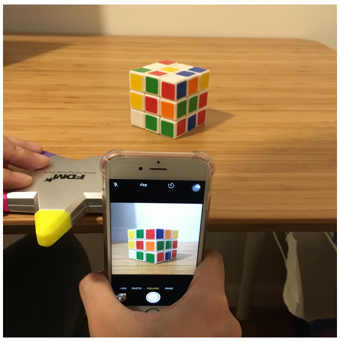
The visible range of this image is $240 * 240$. The position of this epipole is beyond the range of the visible image plane. It is the reasonable result, although the epipole lines seems like converge on a very far position.

4.2 3D Object geometry via triangulation

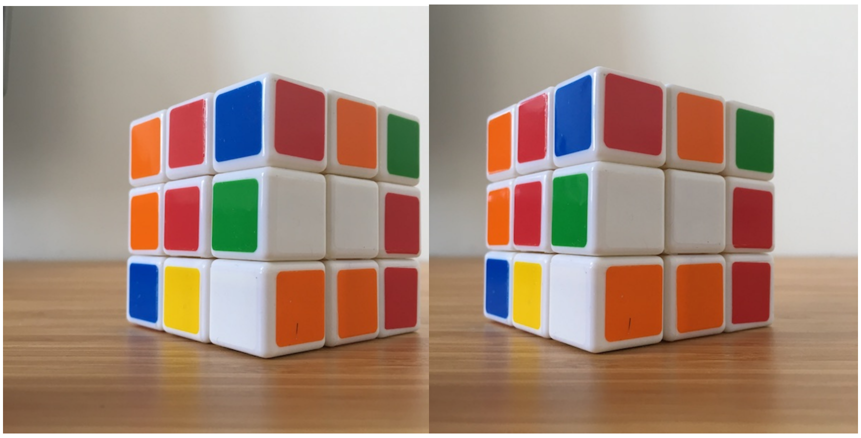
Using a setup with cameras with parallel image planes, take a stereo picture of an object with simple geometry, e.g. a cube (see Fig. 4. Such objects are defined by a small set of landmarks (here 3D corners) and can be reconstructed by displaying a set of lines joining these key points.

- **Define a small set of corresponding key locations by either manual definition of landmarks or a correlation-based image processing method (with selection of only the major key points).**

1) Snap two stereo parallel images



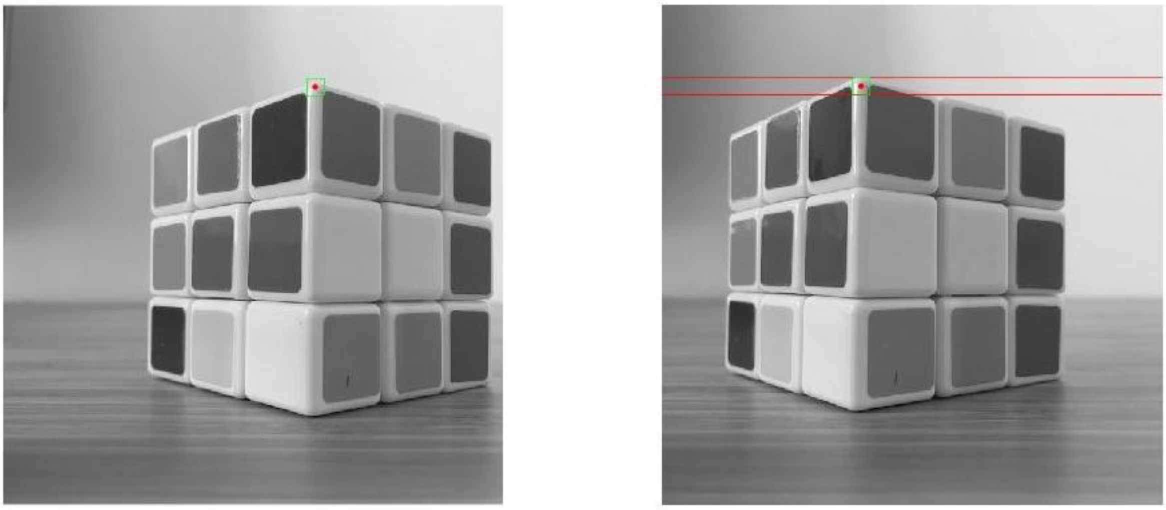
Here is the way that I snapped two parallel images. I used a box that stable the camera and made it perpendicular to my desk, and then, I moved the camera along the side of the desk, snapped the second photo. What's more, I also drew some lines that made me calibrate the position of the camera to minimize the error as much as possible. The length of base line is **20mm**. And the following are the two images.



2) Find the key locations of the cube based on cross-correlation image processing method

In this case, I adopted the cross-correlation method that have discussed in the class to find the corresponding points in the second image.

First, we have to transform the color image to the gray-scale image. Second, to minimize the error, I assume the epipolar line to the selected points in the left image is the parallel line which is the between the two red lines that presented in the right image.



The adopted method is as follow.

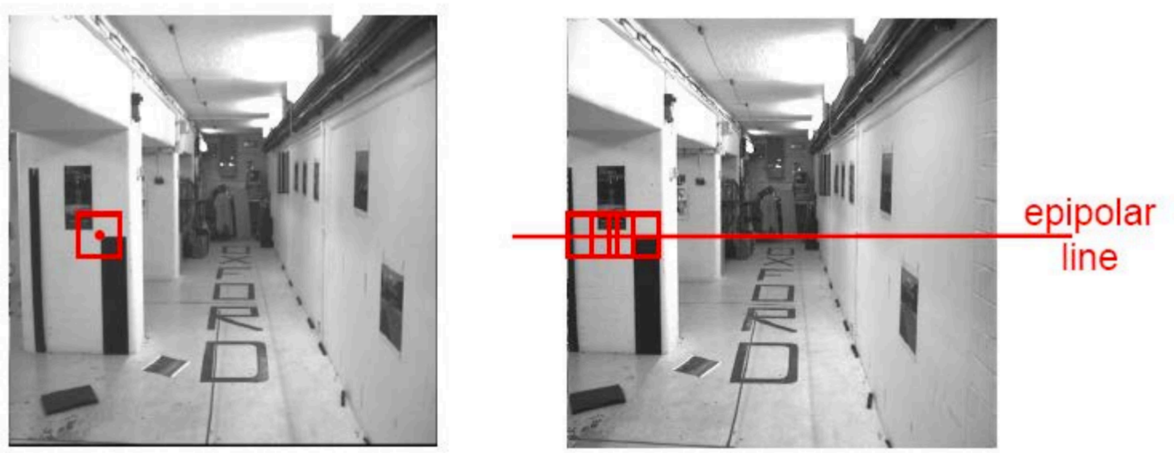


Figure. Cross Correlation Method

The correlation method is as follow.

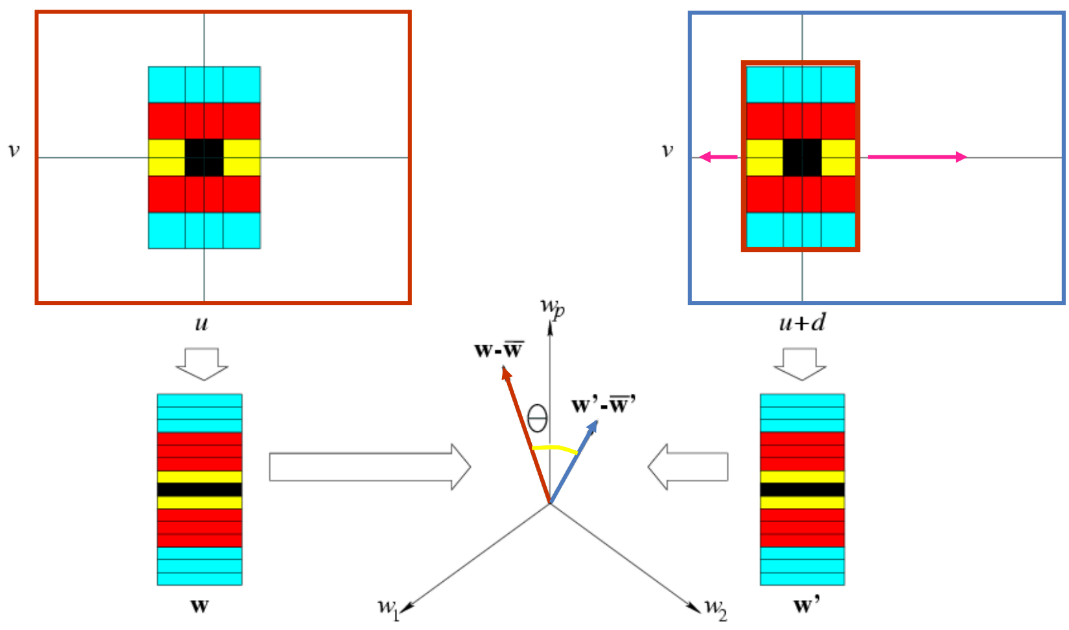


Figure. Cross Correlation Method

The w vector corresponds to the intensity profile of the selected points.
To determine which point is the most matched point, we have to select the maximum value of the following equation:

Normalized cross-correlation:

$$C(d) = \frac{1}{||\boldsymbol{w} - \bar{\boldsymbol{w}}||} \frac{1}{||\boldsymbol{w}' - \bar{\boldsymbol{w}}'||} [(\boldsymbol{w} - \bar{\boldsymbol{w}}) \cdot (\boldsymbol{w}' - \bar{\boldsymbol{w}}')].$$

The value of normalized cross-correlation ranges from -1 to 1, which means, when the values is close to 1, the vectors are more similar to each other, and when the value is close to -1, the vectors are less similar to each other. The example output result is as follow.

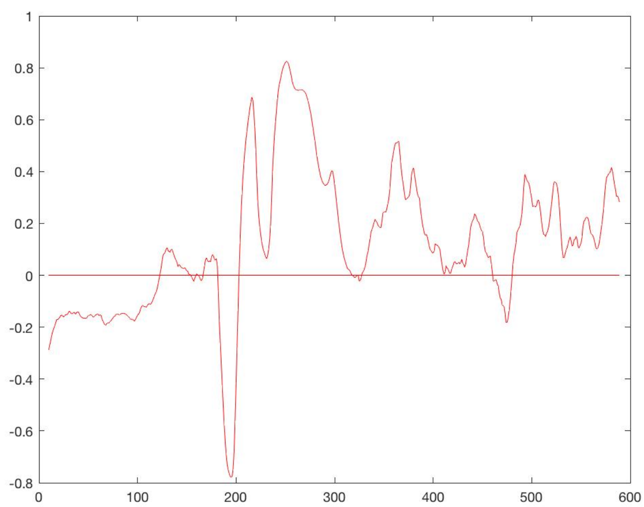


Figure. Example Output

From this image, we can easily get the x coordinate with the max value of NCC. With the epipolar line, we can define the point that we have to find.

Basically, we choose different block size to determine the intensity profile of a point. The size of a block does not have decisive effect on the result of the point matching, because the noise and also the real situation of each sliding window also counts on the result of matching.
What's more, in this case, I can get a very exact corresponding point because this point is very different from the other point along the same epipolar line.
To illustrate the bad result, which is as follow.

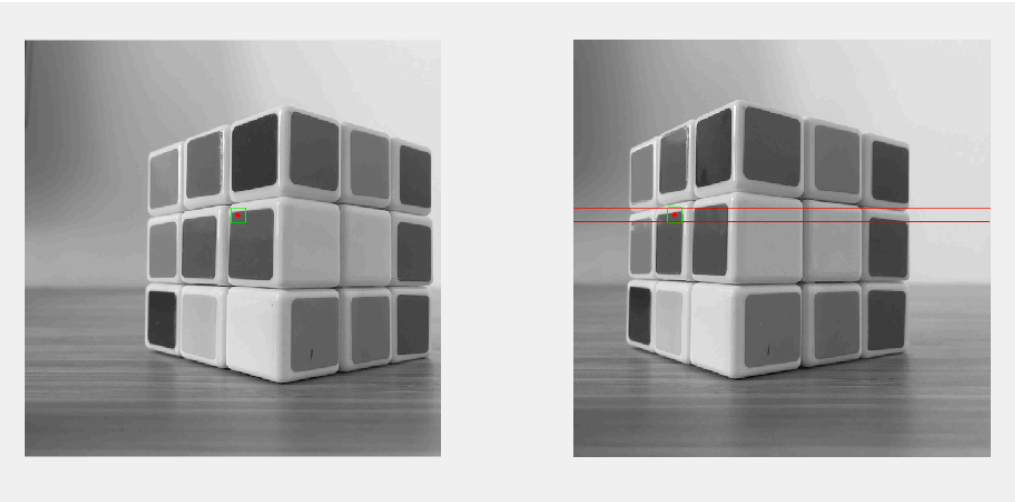


Figure. Unexpected Result

And the relative matching process is:

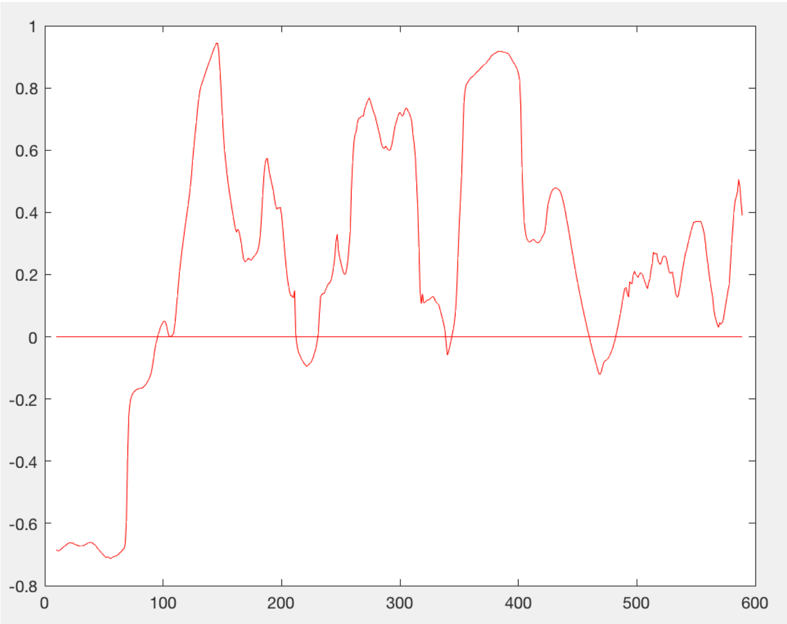
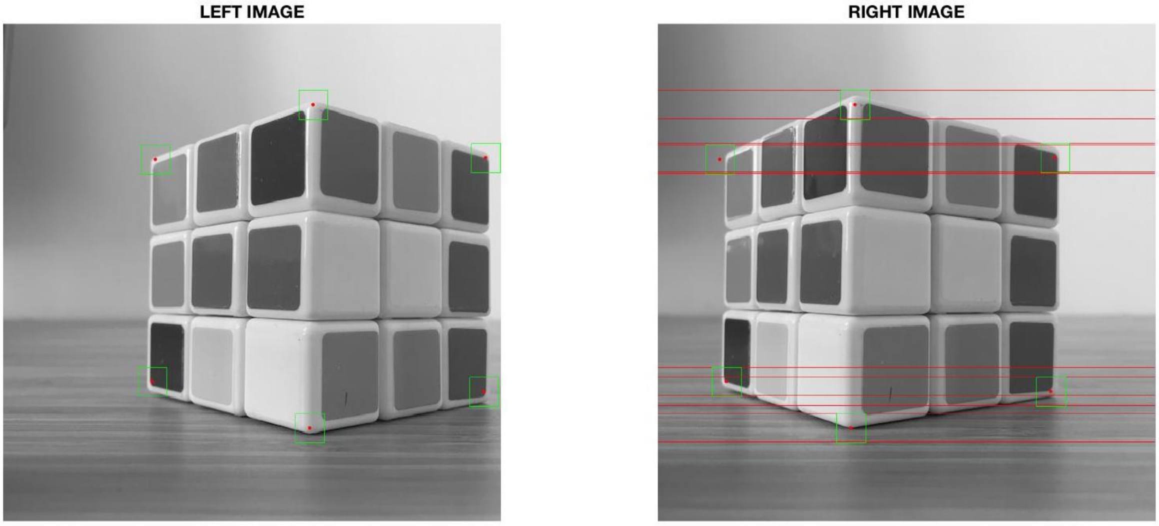


Figure. Unexpected Matching Process

From above, I got an unexpected but reasonable result. Actually, there are multiple points that are very close to the max value, this is the error which would choose the wrong point that we would encounter in the real case.

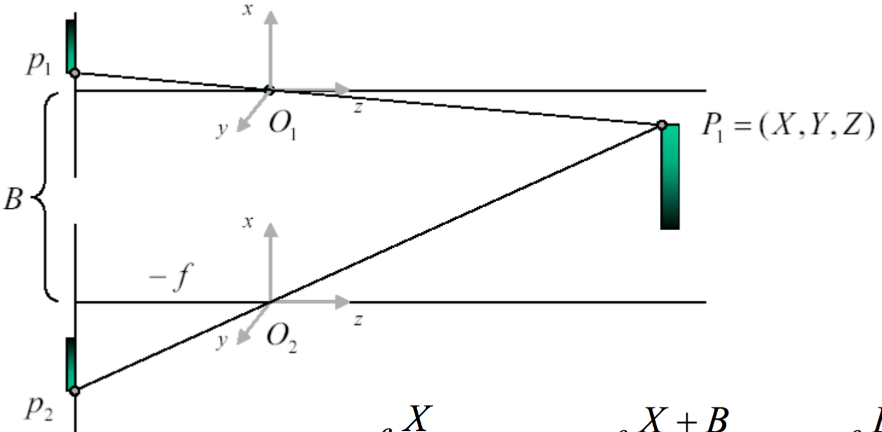
Based on what we have discussed before, I implemented the algorithm myself and define several key points of the left image, and find the corresponding points of the second image based on cross-correlation method.



To determine the disparity without not much error, now I have to choose the result with most matched points, which is presented like above. I have tested with many block size to find the six expected matching points, and this is the best situation I have got. The horizontal disparity can be calculated by the difference value between the x coordinates of the corresponding points in these two images.

- **Calculate horizontal and vertical disparity in pixel units and mm-units, and from those the 3D point coordinates (X,Y,Z).**

  Based on the geometry method that discussed on the class, we have a representative geometric model.



$$x_1 = -f\frac{X}{Z} \qquad x_2 = -f\frac{X+B}{Z} = x_1 - f\frac{B}{Z}$$

$$Z = \frac{fB}{x_1 - x_2}$$

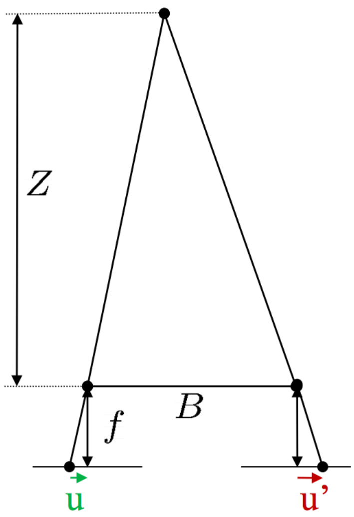


Figure. Geometry Model For Disparity

With $d = |u - u'|$, and the length of baseline which is $B = 20\ mm$ and $f = 4.15\ mm$, we can easily calculate the depth Z, based on the equation:

$$Z = \frac{fB}{d}$$

Note that the disparity we have got at the last step is in pixel unit. But the unit of focal length and baseline is mm-unit, so we have to transform the pixel unit into mm-unit. The relation between these two units can be found at the Assignment 1, which are intrinsic parameters of the camera that we have calibrated before. The pixel density of the two optical axes of my own camera is the same, which is $k = l = 833.33\ pixel/mm$. So, we can get $disparity\ in\ mm\ unit = disparity\ in\ pixel\ unit\ /\ k$

Since $f = 4.15\ mm$, and $B = 20\ mm$. We can easily get our Z value of each points. Beyond this, based on the equation for the pinhole camera projection, which are

$$\begin{cases} \mathrm{x} = -f\dfrac{X}{Z} \\ y = -f\dfrac{Y}{Z} \end{cases}$$

Now that we can get the pixel coordinates and the corresponding world coordinates (origin at the focal center) of each points:

Pixel coordinate(x, y)	World coordinate$(\mathrm{X}, \mathrm{Y}, \mathrm{Z})$
(178, 434)	$(-36.7010\quad -89.4845\quad 713.0584)$
(372, 488)	(-54.7059 -71.7647 508.5784)

(582, 450)	(-109.8113 -84.9057 652.5157)
(182, 168)	(-33.0909 -30.5455 628.7879)
(376, 98)	(-54.8905 -14.3066 504.8662)
(586, 162)	(-113.7864 -31.4563 671.5210)

As we can see, the depth ranges in a reasonable range.

- **Use Matlab or your software to display the 3D points and edge lines for the reconstructed object (only those visible in your images). Choose display viewpoints different from the camera views to verify the quality of 3D reconstruction.**

  Because I only choose six points in this case (I can only see these six points in this case), I connected the points that may show the outline of the object.

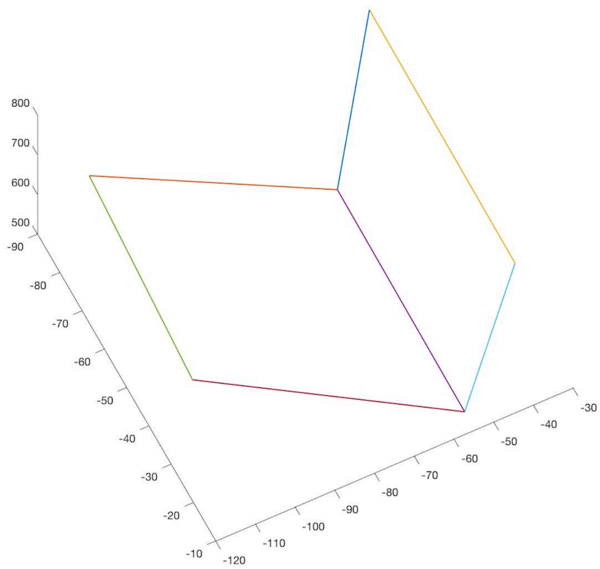


Figure. Reconstructed Outline of the Cube

The reconstructed 3D image looks similar to the outline of the cube that I showed before, the six points are exactly the selected points that presented in the world coordinates (origin at the optical center). Let's display it in other viewpoints.

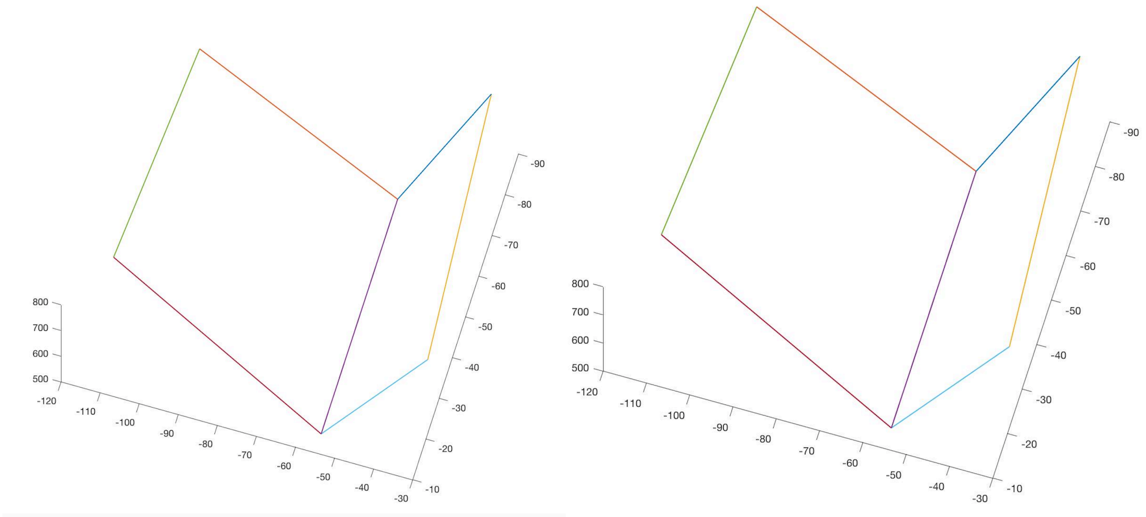


Figure. Reconstructed Outline of the Cube From Different Viewpoints

- **Discuss your results.**

To some extend, the final result make sense and is reasonable. Although the corresponding lines that should be equal to each other since it is a cube, the noise in the image and the error we get in the period of disparity-calculation would make effect on the calculation of the world coordinates. **But the magnitudes of them are very reasonable because they are in the same magnitude**. What's more, the difference between **the depth with max value and the depth with min value is also make sense**, which is about $200\ mm\ =\ 20\ cm$.