Assignment 2 Stereo and 3D Reconstruction from Disparity Report

Zebin Xu (zx586)

zebinxu@nyu.edu

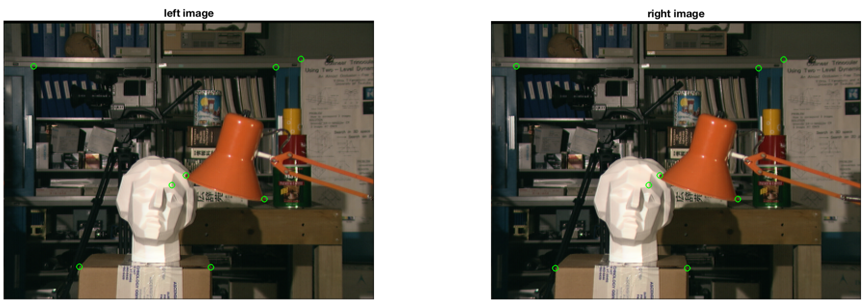
March 14, 2017

Epipolar Geometry from F-matrix

1. Setup

Choose an existing stereo image pair and specify a set of corresponding points in left and right images. We experiment solving the F matrix using 8 and 10 corresponding points respectively.

Figure 1. A stereo image pair.



(a)

(b)

Figure 2. Stereo image pairs with (a) 8 and (b) 10 corresponding landmark pairs.

2. Calculate F matrix using 8-point algorithm.

From epipolar constraint we know that $\mathcal{E}$ is the essential matrix:

$$\mathcal{E} = [t_x]\Re$$

where $[t_x] = \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix}$, $[t_x, t_y, t_z]^T$ and $\Re$ is the translation and rotation between two views respectively.

F is the fundamental matrix:

$$F = K_1^{-T}\mathcal{E}K_2^{-1}$$

where K_1 and K_2 is the calibration matrix of the two cameras (if two cameras are the same then $K_1 = K_2$).

We know that the point correspondence between the stereo image pair can be expresses as a linear equation:

$$u^T F u' = 0$$

where u and u' is the pixel coordinate of the left and right image respectively.

The above equation is:

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

which can be simplified as one matrix multiplication:

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

2.1 Solve with 8 Corresponding Points

Since F is defined up to scale, we let $F_{33} = 1$ and solve the remaining 8 parameters using 8 corresponding points, which give 8 equations. The F matrix is

$$F = \begin{bmatrix} -0.000016059139809 & -0.00193061145712099 & 0.26329939381432 \\ 0.00191566909903207 & -0.00001285536273232 & -0.6508370595505488 \\ -0.256935594662707 & 0.641224014620353 & 1 \end{bmatrix}$$

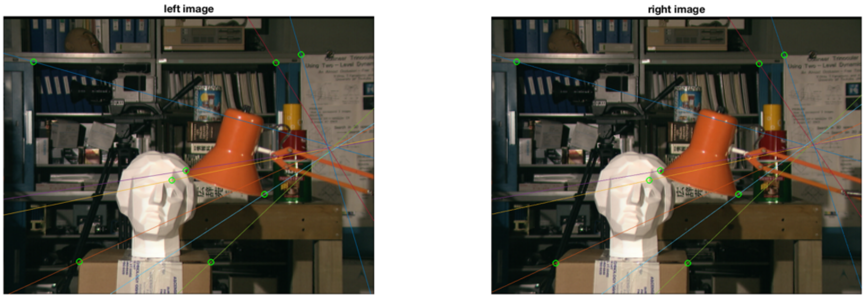
To reduce error, we need to reduce the rank of F from 3 to 2.

1. Take the SVD of F: [U D V] = SVD(F).
2. Set D(3, 3) to zero.
3. Re-compute F = U * D * V'.

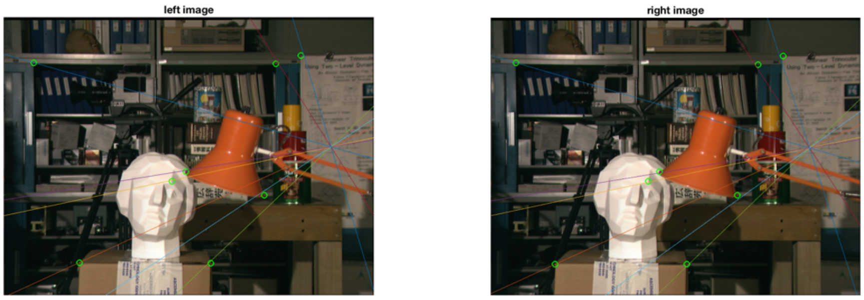
After rank deprivation, we have:

$$F = \begin{bmatrix} -0.0000102288385317 & -0.0019283020117509 & 0.263299410954296 \\ 0.00191805478405094 & -0.0000119103671034 & -0.650837052492029 \\ -0.256935577085393 & 0.641224021582917 & 1.00000000005167 \end{bmatrix}$$

The following two figures pairs show the epipolar lines of the two images without and with rank deprivation. It can be clearly seen that the result if more accurate with rank deprivation. Epipolar lines in each image all intersect at the epipoles of each other.



(a) Epipolar lines of two image pair without performing rank deprivation



(b) Epipolar lines of two image pair after rank deprivation

Figure 3. Epipolar lines of 8 corresponding points in stereo image pair without and with rank deprivation.

2.2 Solve with 10 Corresponding Points using Least Squares.

We sampled 10 corresponding points and solve $u^T F u' = 0$ using least squares method.

1. Construct a 10×9 matrix A such that

$$A = [uu' \quad uv' \quad u \quad u'v \quad vv' \quad v \quad u' \quad v' \quad 1]$$

2. Take the SVD of A: [U S V] = SVD(A).
3. The last column of V (eigenvector associated with the smallest eigenvalue) is the elements of F matrix.
4. Reshape this column to a 3×3 F matrix.
5. Perform rank deprivation (similar to the steps in solving 8 equations).

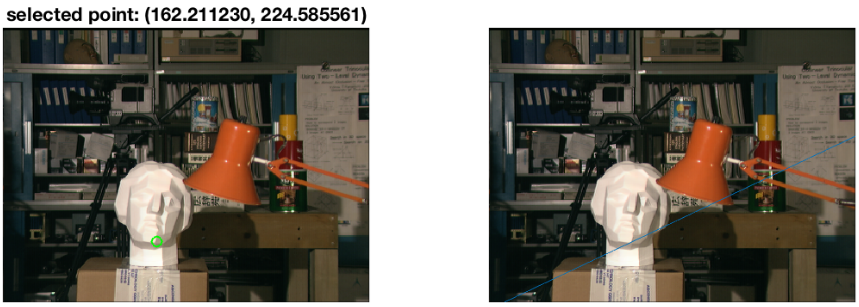
Finally, we get F matrix solved using 10 corresponding points.

$$F = \begin{bmatrix} -0.00005677271817234 & -0.00104688474793069 & 0.168428043932593 \\ 0.00104180185458174 & -0.00001579264984773 & 0.304294908613248 \\ -0.169371539518221 & 0.301673986785379 & 0.871401462189994 \end{bmatrix}$$

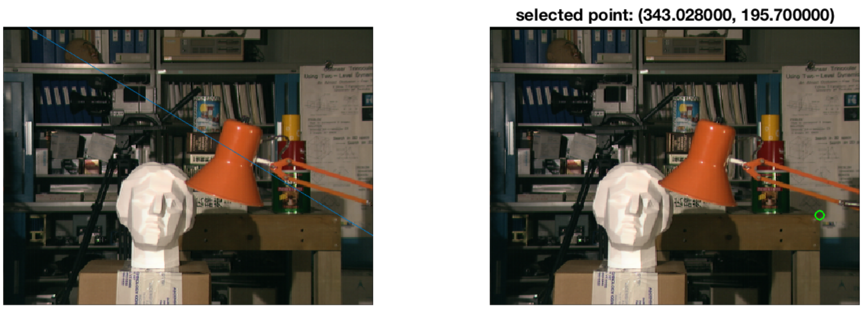
Figure 4. Epipolar lines of 10 corresponding points in stereo image pair

3. F-matrix Verification with Epipolar Lines

To verify our F matrix, for a stereo image pair, we select a point in one image and draw the epipolar line in another image to see if the corresponding point lies on the epipolar line. From Figure 5 we can see that the epipolar lines pass exactly through the corresponding point and thus the F matrix is correct enough.



(a) Select a point (162.2, 224.6) on the left, draw the epipolar line on the right.



(b) Select a point (343.0, 195.7) on the right, draw the epipolar line on the left.

Figure 5. Corresponding epipolar line of a select point

4. Epipole

The left and right epipoles in the stereo image pair can be computed using the equations:

Left epipole: $Fe = 0$

Right epipole: $F^T e' = 0$

We can compute the left epipole e by calculating the eigenvalues and eigenvectors of $F^T F$, and obtain the eigenvector associated with the smallest eigenvalue and scale the vector so that the third element is 1. Equivalently, we can take the SVD of F, since $eig(F^T F) = svd(F)$. And similarly extract the eigenvector associated with the smallest eigenvector.

We get the left epipole at (294.5, 162.5), the right epipole at (285.7, 161.0) and draw the epipoles as red circle on the images. Comparing the intersection points of epipolar lines and the computed epipoles, we can see that they perfect coincide.

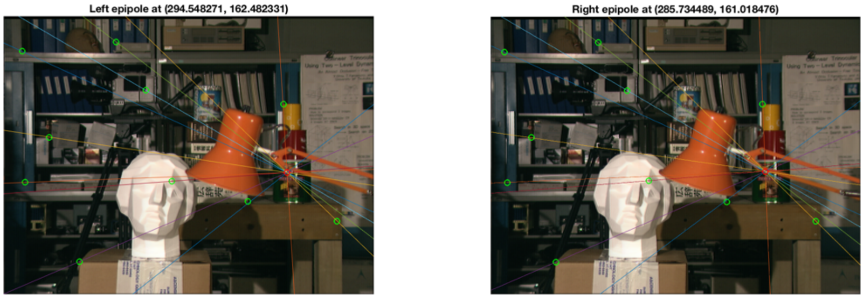


Figure 6. Epipoles (red circle) in two images.

5. Verification for Another Image Pair

To further verify the process of calculating F-matrix and epipolar lines, we shoot a stereo-pair of a scene. The right image is taken by translating and rotating the camera pose when taking the left.

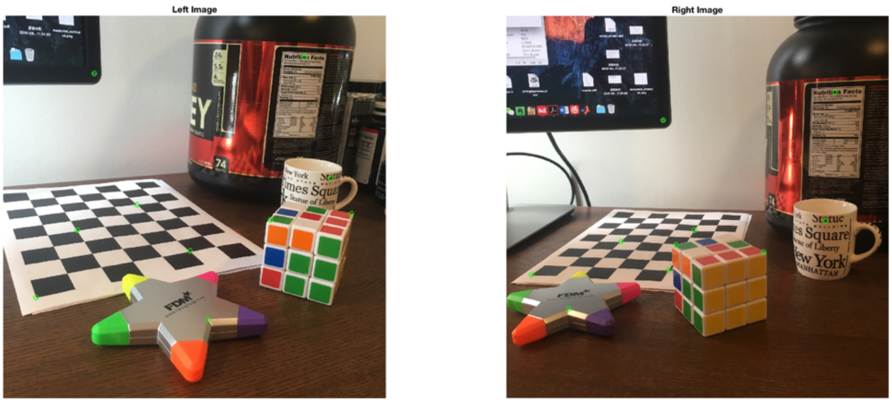


Figure 7. Stereo image pair by translating and rotating the camera. Each image has 10 corresponding points.

We go through the same procedure as in the previous image pair.

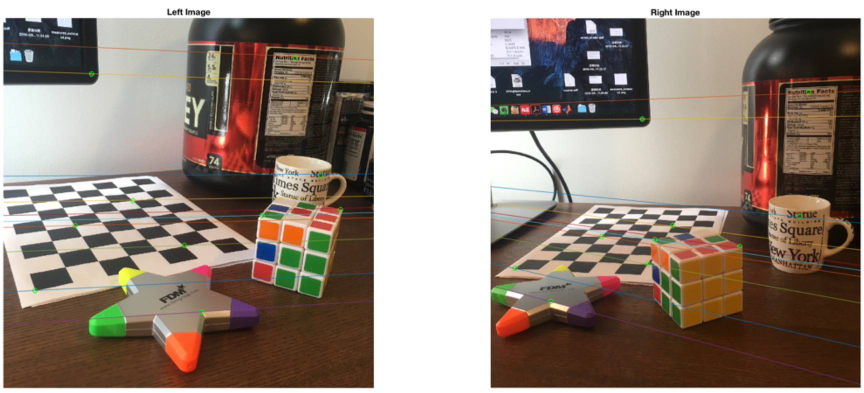


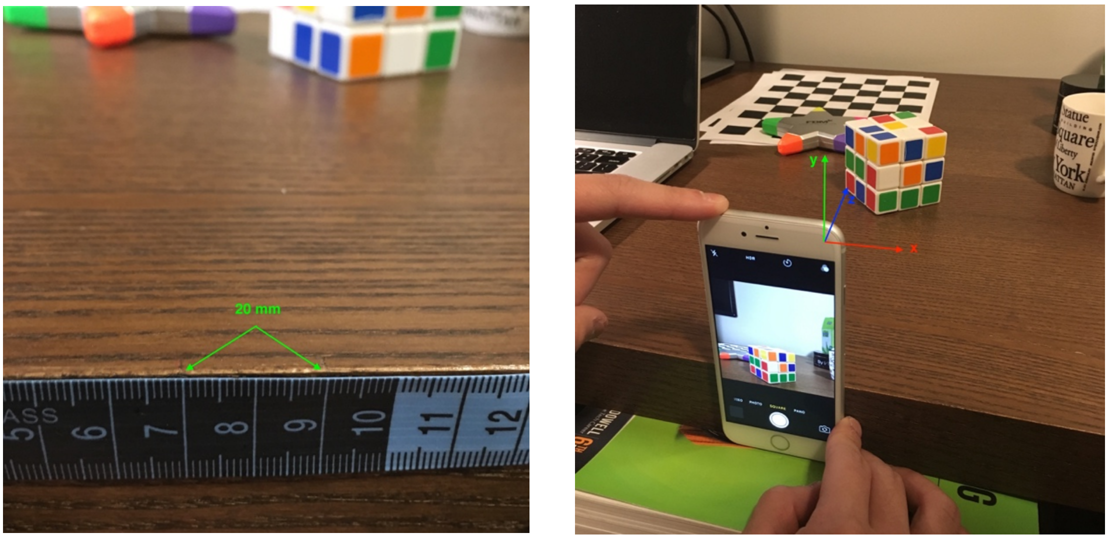
Figure 8. Stereo image pair with corresponding epipolar lines

As can be seen from Figure 8, the calculated epipolar lines lie on the right positions. But in this case, the epipole of each view is not inside the image. The epipole of the left image is at (20870, 1159) while the epipole of the right image is at (-4799, 904). From the trend of the epipolar lines above, the calculated epipoles are reasonable.

3D Object geometry via Triangulation

1. Setup

To simplify the 3D reconstruction process, image rectification is not needed in this assignment. Thus the image pair must have parallel image planes so that the epipolar lines are horizontal and the transformation between two image planes is horizontal. This means that the disparity will be horizontal only. To achieve this, we record two fixed point on the table as indicated in Figure 7(a). The length of this fixed points is the baseline of two image planes. Take one picture when the left side of cellphone is aligned with the left fixed point, and then move camera along the table for 20 mm to take the second picture. Since the camera is at the top-right corner of the cellphone, we define a camera coordinate whose origin is at the top-right corner as shown in Figure 7(b). To obtain positive z values, we define z axis facing forward.



(a) (b)

Figure 9. (a) A fixed baseline (20mm) for camera to move along. (b) Parallel image pair capturing setup. The camera is at the top-right corner of the phone.

The stereo image pair with parallel image planes can be seen in Figure 10.

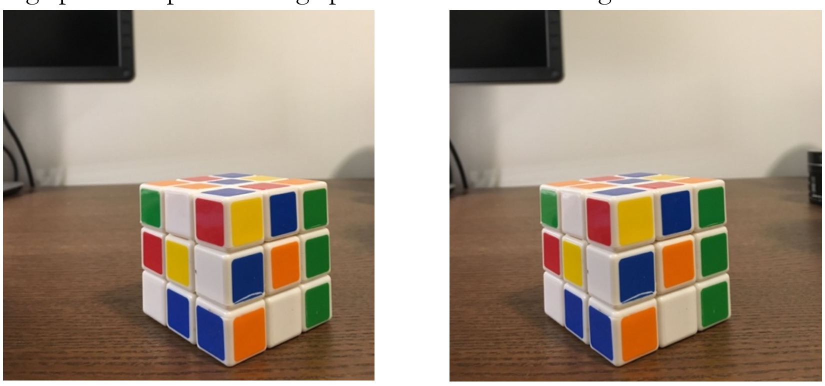


Figure 10. Stereo image pair with parallel image planes

2. Find Corresponding Points using Normalized Cross Correlation

For each point in left image, we can find matched point in right image as the images now have horizontal epipolar lines so that we can search on a 1D line instead of the whole 2D image array. We use zero-mean normalized cross-correlation (NCC) to find matched pattern. I will show later that some optimization has been done to reduce the time to compute NCC so that we don't actually need to search the whole line.

The method to find match point using NCC can be summarized as following:

1. Zero-padding for two images. For a $m \times n$ window to slide, add (m-1)/2 rows of zero both to the top and bottom of the image, (n-1)/2 columns of zero both to the left and right.
2. For a point p in the left image, extract from left padded image a $m \times n$ window whose center is p. Convert this window to a vector w and then to a zero-mean vector $w_z = w - \bar{w}$.
3. Starting from the first pixel (the first pixel refers to first column in the image before padding) of the same row (line) on the right image, extract the $m \times n$ window and convert it to a vector w' and then to a zero-mean vector $w_z' = w' - \bar{w}'$.
4. Calculate normalized cross-correlation: $C(d) = \frac{w_z \cdot w_z{'}}{\|w_z\| \|w_z{'}\|}$
5. Go back to step 3 to search the remaining points along the line in the right image. The point with the maximum correlation is the match point we are looking for.

However, we don't have to search a whole line for each point.

It is due to the fact that the disparity or difference of each pixel pair in two images are consistent, which means that the difference of corresponding pixels' positions is either positive for the whole image or negative for the whole image, depending on how we subtract the positions.

So we don't have to start from the first column in the right image every time. Instead, we can start from the same position of the left image.

Before doing this, we need a swap: for each point in the right image, find matched points in the left. As shown in Figure 11, before swapping, we have to fine corresponding point at (1900, y) by searching backward in the right image(upper-right); after swapping, we can find corresponding point searching forward in the right image(bottom-right).

Another optimization is that we don't have to search until the end of line. We first estimate a maximum disparity of the image pair, and search at most maximum disparity number of pixels along the line.

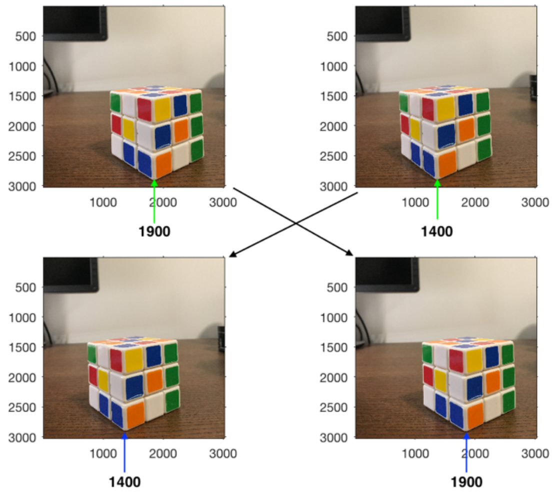
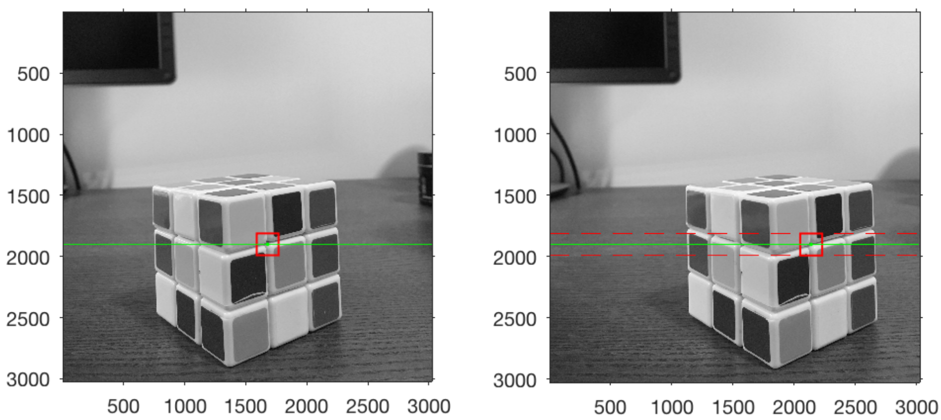
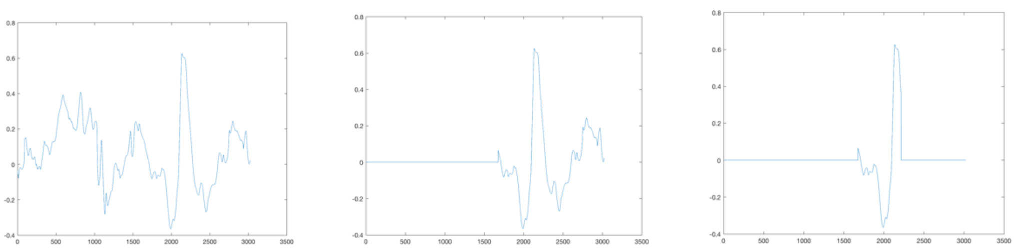


Figure 11. Image pair swapping diagram. Before swapping, the point (1900, y) in the upper-left corresponds to the point (1400, y) in the upper-right. After swapping, the point (1400, y) in the bottom-left corresponds to the point (1900, y) in the right. We can search forward after swapping.

As shown in Figure 12, we can see that our optimized method reduces the search range, increase the searching efficiency significantly.



(a) Search along the line to find a corresponding point.



(b) Original method (c) Optimized method (d) Further optimized

Figure 12. A comparison to visualize the optimization of normalized cross correlation to find match points. (b) The original method to search the whole line. (c) Search by starting from the position of the left point. (d) Search by ending earlier.

The optimized method not only increase the searching efficiency but also accuracy. As can be show in Figure 13 and 14, the optimized method can avoid incorrect peaks of correlation.

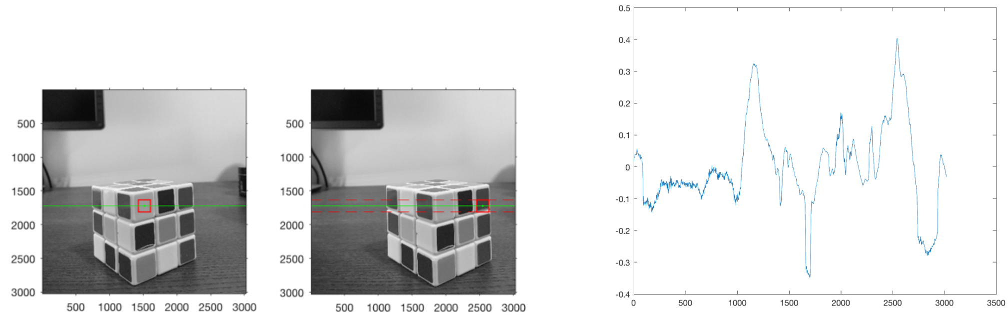


Figure 13. Incorrect match due to two peaks in correlation.

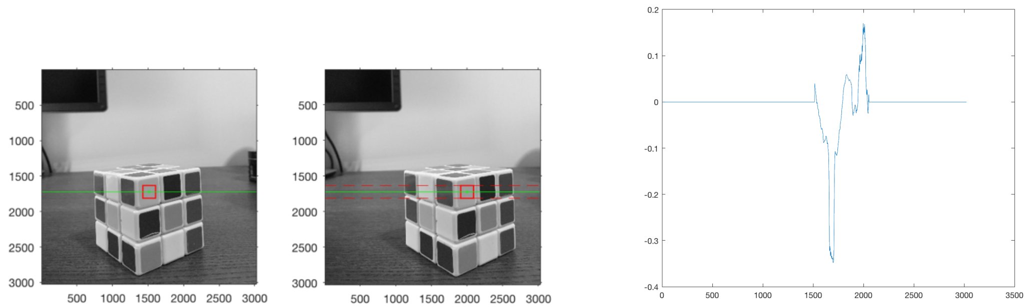


Figure 14. Match points using optimized method.

3.Calculate Disparity and 3D point coordinates.

Disparity is the difference between the same pixel location in two images. To compute depth z, we have z=fB/d, where f is the focal length, and B is the length of baseline, d is the disparity. Since we set our z axis facing forward, we don't have to add a minus here. However, the unit of d should be mm, we need to convert pixel unit to mm unit. To do this, we need to know that horizontal sensor size 1/k from the camera.

Based on the intrinsic parameters we've got from camera calibration in assignment 1 (see Table 1), we know the focal length of the camera is 4.15 mm, horizontal sensor size is $\frac{1}{k} = \frac{1}{\frac{\alpha}{f}} = \frac{f}{\alpha} = \frac{4.15}{3368.84} = 0.00123$ mm.

Table 1. Intrinsic parameters

Parameter	Value
θ	1.5764 $\approx 90.32°$
u_0	1518
v_0	1512
α	3368.84
β	3353.62

Next we need to compute x, y. From camera intrinsic property we know that

$$\begin{cases} u = \alpha \dfrac{x}{z} + u_0 \\ v = \beta \dfrac{y}{z} + v_0 \end{cases}$$

where (u, v) is the pixel coordinate in pixel unit, (x, y, z) is the 3D coordinate. We can easily calculate x and y:

$$\begin{cases} x = \dfrac{(u - u_0)z}{\alpha} \\ y = \dfrac{(v - v_0)z}{\beta} \end{cases}$$

Figure 15 shows the depth of 7 viewable key points of a cube. We can reconstruct these 7 points 3D coordinates using the above equations for x and y.

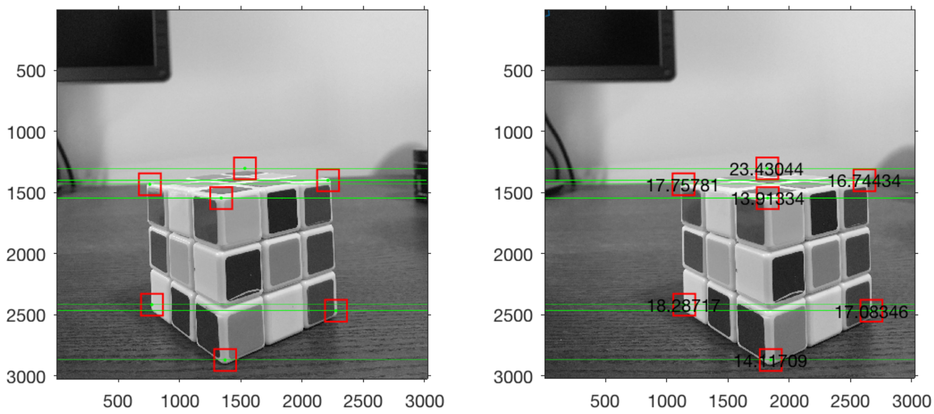


Figure 15. Depth (cm) of 7 key points of a cube.

Figure 16 shows the disparity map of different window size. It can be seen that smaller window size results in a depth map with more details.

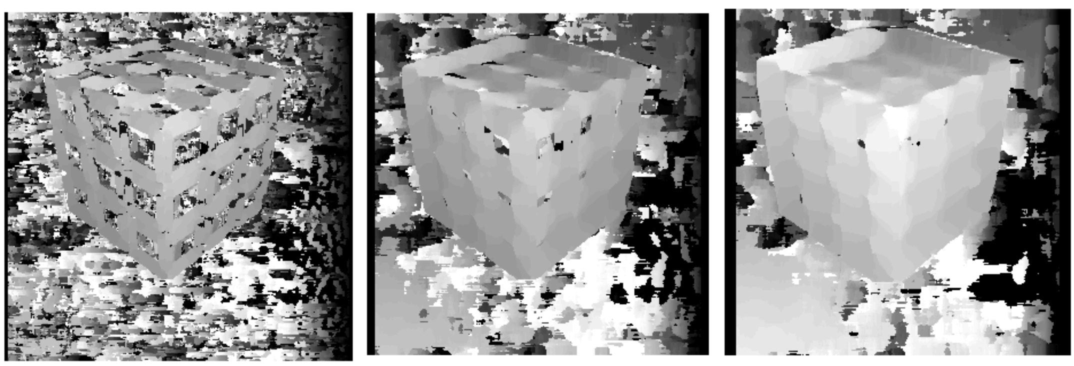


Figure 16. Disparity map of the cube with window size of 7x7, 15x15, 21x21. The image is downsized to 302x302.

4. Cube Reconstruction.

Since we can only see 7 points of the cube, we lack 1 vertex to reconstruct the cube. The followings are different viewport of the reconstructed shape.

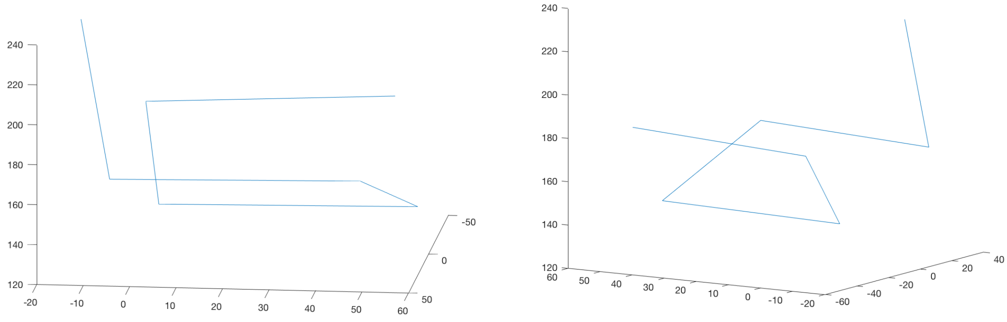


Figure 17. 7 key points of a cube joined with edge lines

5. Discussion

It can be seen that the edges are not perfectly parallel or perpendicular. This may be because of the sampling error of the key points or the intrinsic parameter errors, etc.