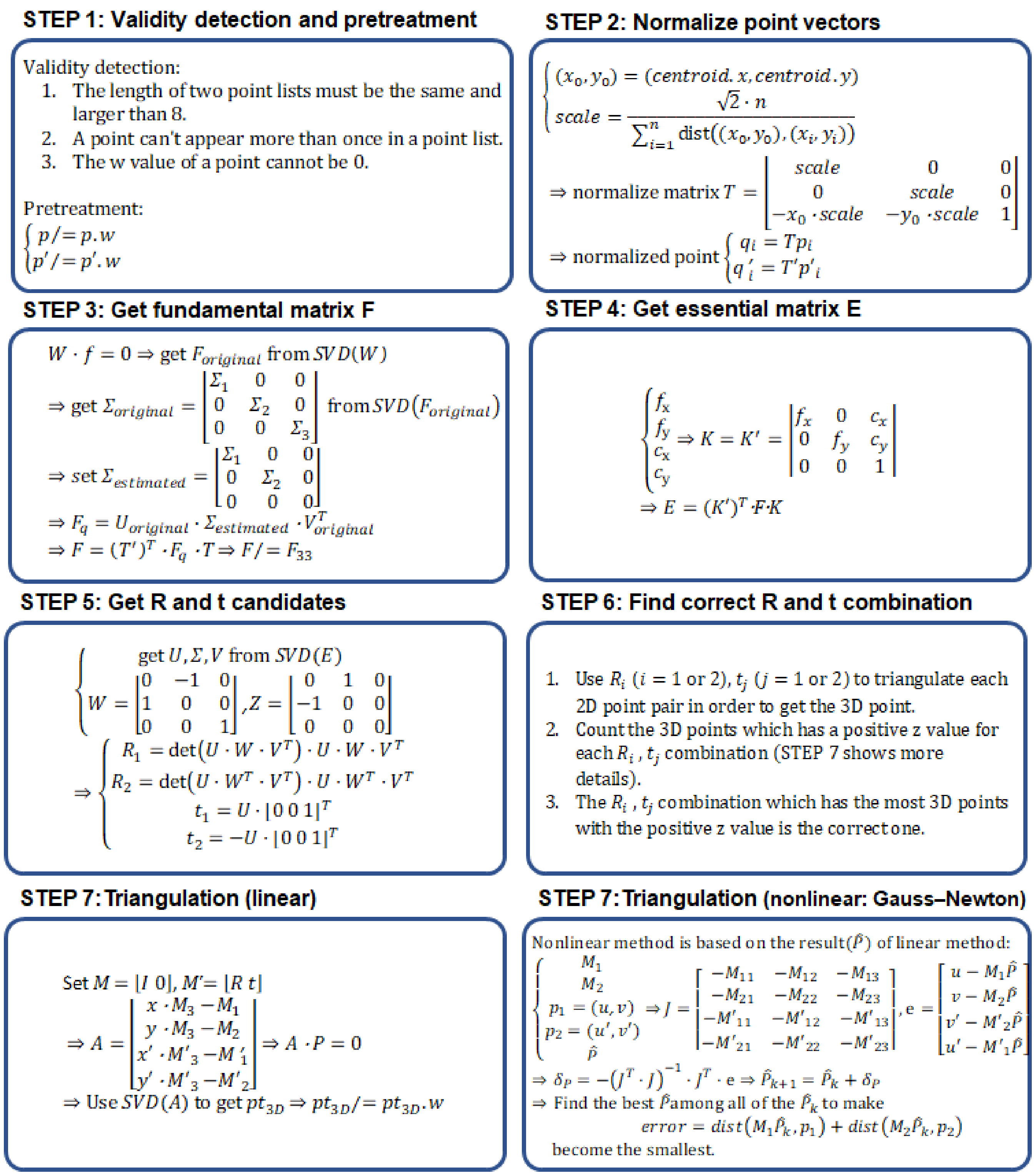
Assignment 2: Triangulation

Linjun Wang(5214513), Zhenyu Liu(5386586), Ziyan Wu(5360684)

1.Methodology Description



STEP 1: Validity detection and pretreatment

Validity detection:

1. The length of two point lists must be the same and larger than 8.
2. A point can't appear more than once in a point list.
3. The w value of a point cannot be 0.

Pretreatment:

$$\begin{cases} p/=p.w \\ p'/=p'.w \end{cases}$$

STEP 2: Normalize point vectors

$$\begin{cases} (x_0, y_0) = (centroid.x, centroid.y) \\ scale = \dfrac{\sqrt{2}\cdot n}{\sum_{i=1}^{n} \text{dist}((x_0,y_0),(x_i,y_i))} \end{cases}$$

$$\Rightarrow \text{normalize matrix } T = \begin{bmatrix} scale & 0 & 0 \\ 0 & scale & 0 \\ -x_0\cdot scale & -y_0\cdot scale & 1 \end{bmatrix}$$

$$\Rightarrow \text{normalized point} \begin{cases} q_i = Tp_i \\ q'_i = T'p'_i \end{cases}$$

STEP 3: Get fundamental matrix F

$$W\cdot f = 0 \Rightarrow \text{get } F_{original} \text{ from } SVD(W)$$

$$\Rightarrow \text{get } \Sigma_{original} = \begin{bmatrix} \Sigma_1 & 0 & 0 \\ 0 & \Sigma_2 & 0 \\ 0 & 0 & \Sigma_3 \end{bmatrix} \text{ from } SVD(F_{original})$$

$$\Rightarrow \text{set } \Sigma_{estimated} = \begin{bmatrix} \Sigma_1 & 0 & 0 \\ 0 & \Sigma_2 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\Rightarrow F_q = U_{original}\cdot \Sigma_{estimated}\cdot V_{original}^T$$

$$\Rightarrow F = (T')^T\cdot F_q\cdot T \Rightarrow F/=F_{33}$$

STEP 4: Get essential matrix E

$$\begin{cases} f_x \\ f_y \\ c_x \\ c_y \end{cases} \Rightarrow K = K' = \begin{vmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{vmatrix}$$

$$\Rightarrow E = (K')^T\cdot F\cdot K$$

STEP 5: Get R and t candidates

$$\begin{cases} \text{get } U,\Sigma,V \text{ from } SVD(E) \\ W = \begin{bmatrix} 0 & -1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix}, Z = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \end{cases}$$

$$\Rightarrow \begin{cases} R_1 = \det(U\cdot W\cdot V^T)\cdot U\cdot W\cdot V^T \\ R_2 = \det(U\cdot W^T\cdot V^T)\cdot U\cdot W^T\cdot V^T \\ t_1 = U\cdot |0\ 0\ 1|^T \\ t_2 = -U\cdot |0\ 0\ 1|^T \end{cases}$$

STEP 6: Find correct R and t combination

1. Use R_i ($i = 1$ or 2), t_j ($j = 1$ or 2) to triangulate each 2D point pair in order to get the 3D point.
2. Count the 3D points which has a positive z value for each R_i, t_j combination (STEP 7 shows more details).
3. The R_i, t_j combination which has the most 3D points with the positive z value is the correct one.

STEP 7: Triangulation (linear)

Set $M = [I\ 0]$, $M' = [R\ t]$

$$\Rightarrow A = \begin{bmatrix} x\cdot M_3 - M_1 \\ y\cdot M_3 - M_2 \\ x'\cdot M'_3 - M'_1 \\ y'\cdot M'_3 - M'_2 \end{bmatrix} \Rightarrow A\cdot P = 0$$

$\Rightarrow$ Use $SVD(A)$ to get $pt_{3D} \Rightarrow pt_{3D}/=pt_{3D}.w$

STEP 7: Triangulation (nonlinear: Gauss–Newton)

Nonlinear method is based on the result($\hat{P}$) of linear method:

$$\begin{cases} M_1 \\ M_2 \\ p_1 = (u,v) \\ p_2 = (u',v') \\ \hat{P} \end{cases} \Rightarrow J = \begin{bmatrix} -M_{11} & -M_{12} & -M_{13} \\ -M_{21} & -M_{22} & -M_{23} \\ -M'_{11} & -M'_{12} & -M'_{13} \\ -M'_{21} & -M'_{22} & -M'_{23} \end{bmatrix}, e = \begin{bmatrix} u - M_1\hat{P} \\ v - M_2\hat{P} \\ v' - M'_2\hat{P} \\ u' - M'_1\hat{P} \end{bmatrix}$$

$$\Rightarrow \delta_P = -(J^T\cdot J)^{-1}\cdot J^T\cdot e \Rightarrow \hat{P}_{k+1} = \hat{P}_k + \delta_P$$

$\Rightarrow$ Find the best $\hat{P}$ among all of the $\hat{P}_k$ to make

$$error = dist(M_1\hat{P}_k, p_1) + dist(M_2\hat{P}_k, p_2)$$

become the smallest.

Figure 1: Main workflow

2. Result Show

All result files are stored in our Github repository: https://github.com/fiodccobw/geo1016-2 .
We implement this assignment by using both linear method and nonlinear method (Gauss-Newton). Set the parameter *is_linear* of the function *get3dpoint()* as *false* the program will use the nonlinear method. Or it will use the linear method.

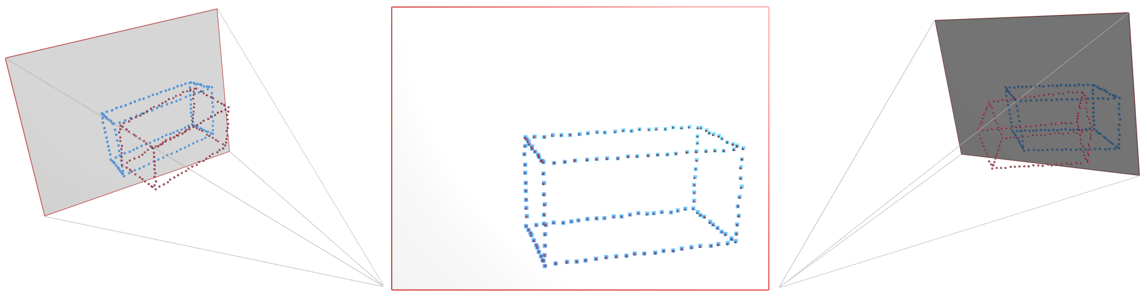


Figure 2: result using linear method

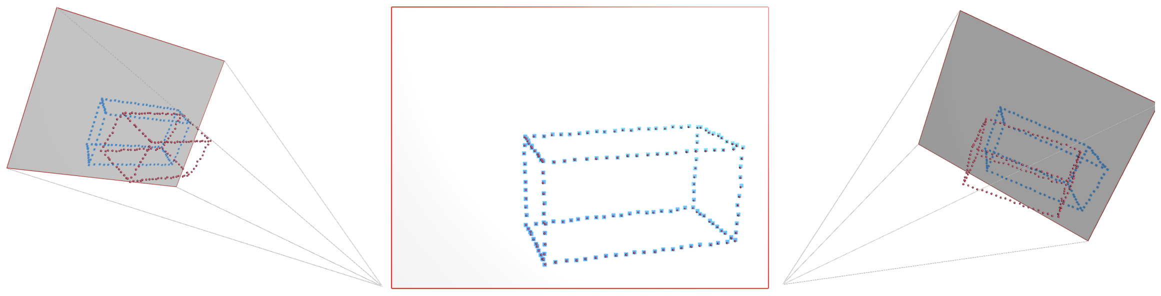


Figure 3: result using nonlinear method

3.Evaluation of results

3.1 Problems to be evaluated

After getting the final results, we mainly focus on two questions:

- **How accurate are our results?**
  In order to solve this problem, we first need to design a reasonable accuracy evaluation method.This is explained in detail in section 3.2.
- **Is the nonlinear method more accurate than the linear method?**
  Since in our program, the nonlinear method uses the final result of the linear method as its initial data and carries on the iterative optimization. So in theory, the nonlinear method will have better accuracy than the linear method. But whether this is the case needs to be confirmed by the relevant data.

3.2 Accuracy evaluation method

We calculate average distance between the projection points of each 3D point on the two image planes and its corresponding two 2D points, and take this value as the error:

$$error = \frac{\sum_{i=0}^{n} (dist(M \cdot \hat{P}_n, p_n) + dist(M' \cdot \hat{P}_n, p'_n))}{n}$$

3.3 Accuracy evaluation result

From table 1, we can find that the nonlinear method has better accuracy than the linear method. This is the same as our inference.

Table 1: The accuracy of linear and nonlinear methods

	linear method	**nonlinear method**
Error	1.95221	1.94681

3.4 Accuracy improvement methods

Generally speaking, the accuracy can be improved during the data collection and data processing.

3.4.1 During data collection

- **The coordinates of the points must be accurate**
  The coordinate accuracy of the point itself is the most critical requirement. If the accuracy of the point itself has serious problems, the improvement methods listed below are useless.

- **Increase the number of point pairs**
  Although it is theoretically possible to complete 3D reconstruction with only 8 points or less, the more points the better for noise and other factors.

- **Increase the images number for SfM (StructurefromMotion)**
  In addition to providing more point pairs, more images make the overall processing more robust.

- **Make sure that the key feature points of the geometry are collected and accurate**
  We find that the importance of different points in the input file for 3D reconstruction is different. For example, we delete the third point pair in input files, the accuracy doesn't change obviously. While we delete the first pair or the last pair of points, the result will have a very poor accuracy. We think the reason may be that some points are the key feature points, such as the vertices of polyhedra, and the missing of these points or the error of their coordinates will seriously affect the overall accuracy.

3.4.2 During data processing

- **Remove the exception point pairs**
  Use methods such as RANSAC to rule out outliers.

- **Residual optimization**

In this homework, the Gauss-Newton method can be considered as an optimization of the linear method, further improving the accuracy. In addition, Levenberg-Marquardt algorithm and other methods can also be chosen in this step.

4.Intrinsic parameters

4.1 How to obtain intrinsic parameters

- Use camera calibration.
- Get the parameters information from camera producers.

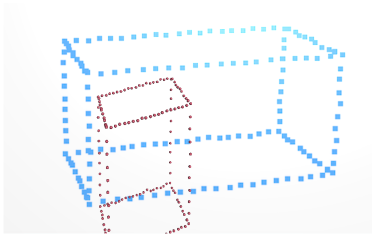
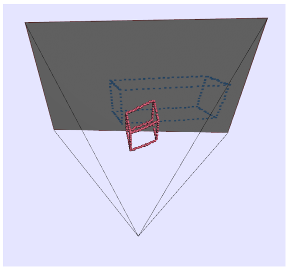
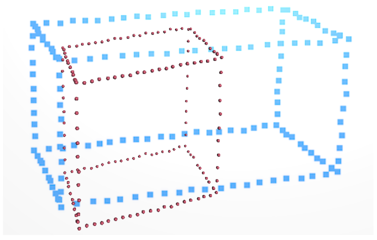
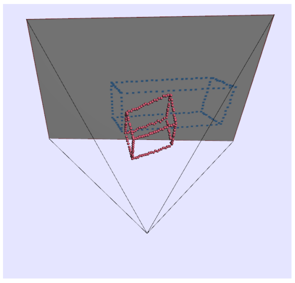
4.2 Effect of errors in the intrinsic parameters

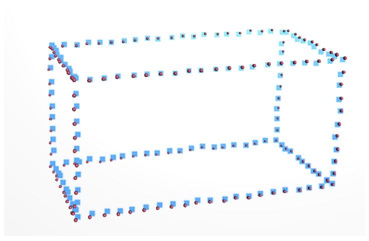
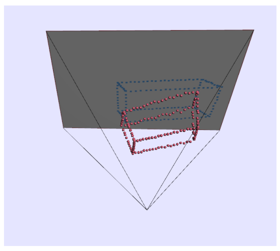
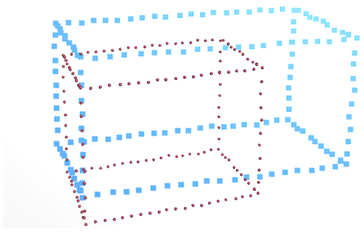
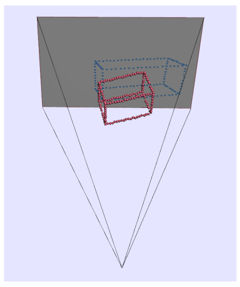
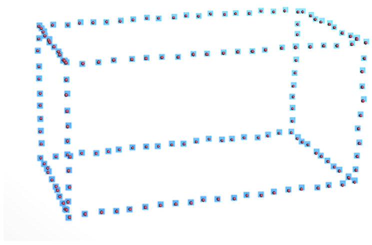
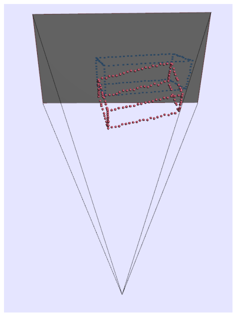
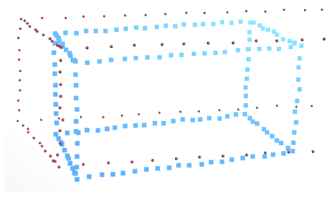
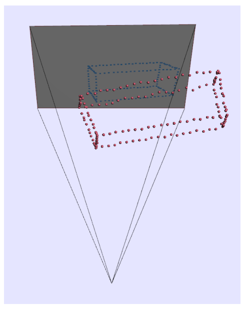
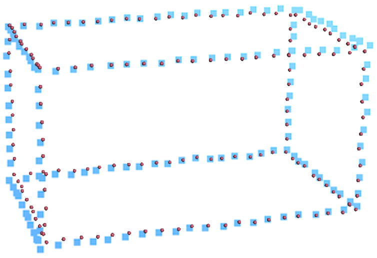
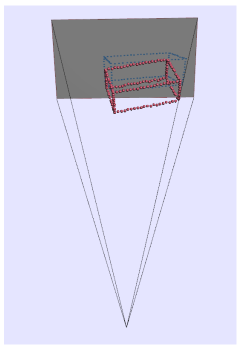
4.1.1 Effect caused by fx/fy

To research the effect of errors in fx and fy, we fixed the values of cx and cy, adjusted fx and fy to observe the changes in results. As we can see in table 2, we changed fx and fy in units of 500, which is a large enough span, so we can see the changes more clearly:

- It mainly affects the geometric shape, not the displacement.This variation usually has a large effect on the error.
- The larger the fx or fy, the smaller the width of the 3D object in the horizontal direction, it may seem like more squashed.
- When fx and fy are equal, the error would be relatively small. The distortion of 3D geometry is no longer along the axis, but the closer to the edge of the image field, the greater the distortion will be (see table 3).

Table 2: Effect of errors in intrinsic parameters (fx and fy)

fx	fy	result		error
1500	500			179.695
1000				68.5862

500				5.09798
1500	1000			104.392
1000				1.95221
500				61.7149
1500	1500			9.43699

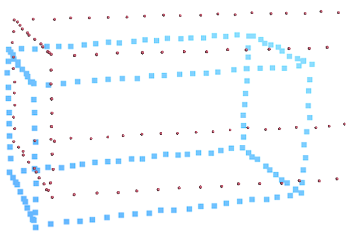
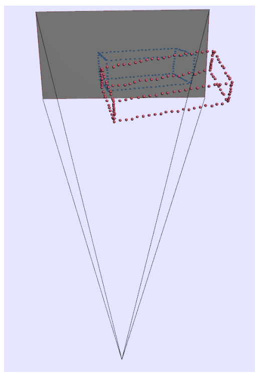
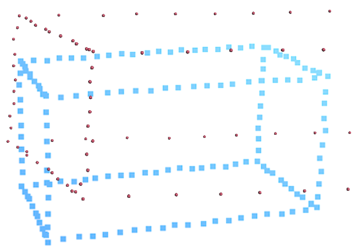
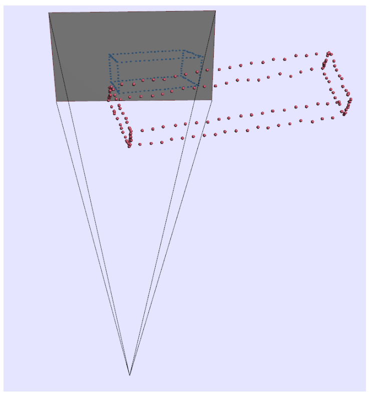
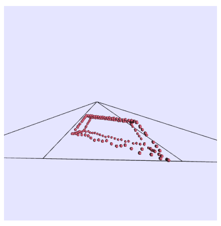
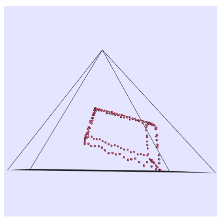
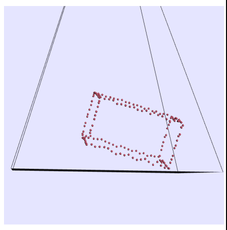
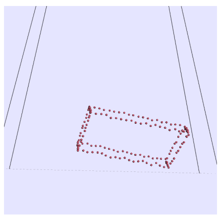
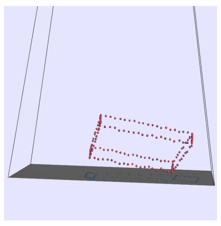
1000				101
500				168.167

Table 3: Effect of errors in intrinsic parameters (fx and fy are equal)

fx=200, fy=200	fx=500, fy=500	fx=1000, fy=1000	fx=1500, fy=1500	fx=2000, fy=2000
error = 6.7511	error = 5.0979	error = 1.9522	error = 9.3427	error = 19.3276
				

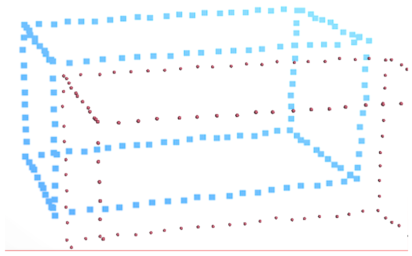
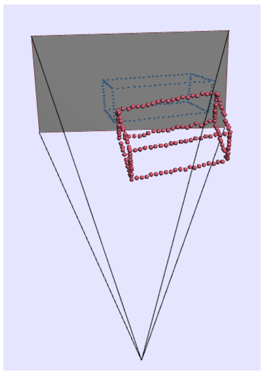
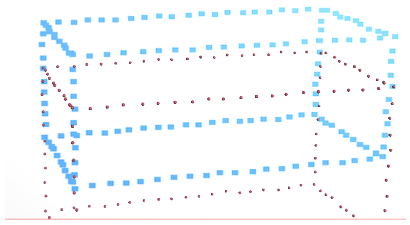
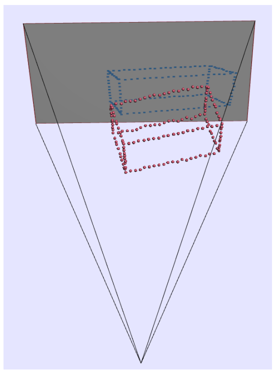
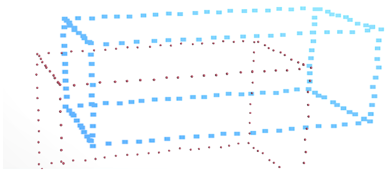
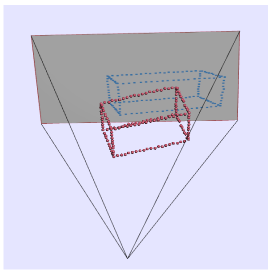
4.1.2 Effect caused by cx/cy

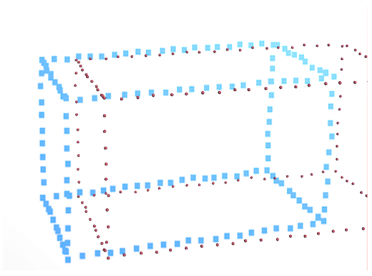
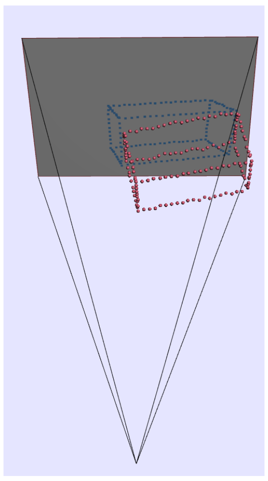
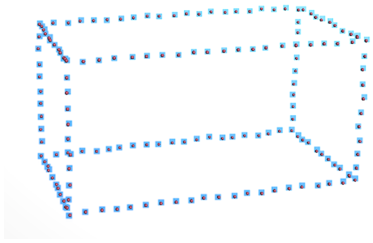
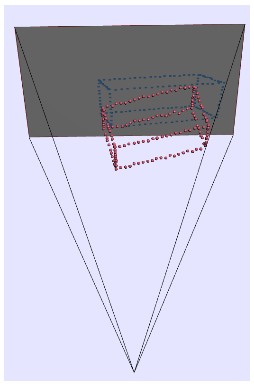
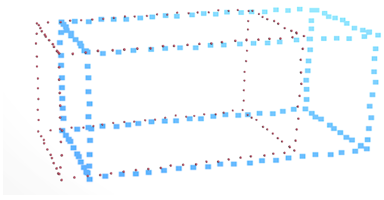
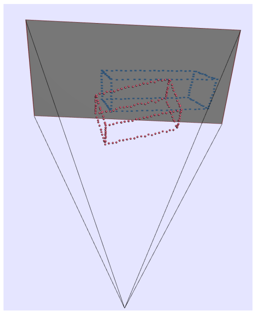
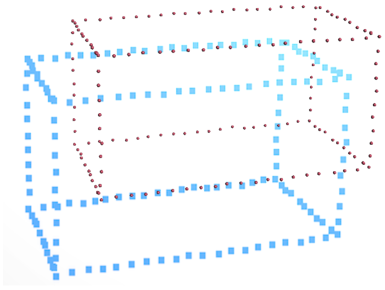
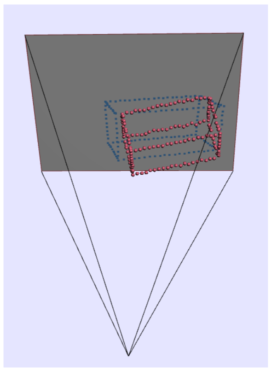
We changed fx and fy in units of 50 to observe the changes of results. As we can see from table4:

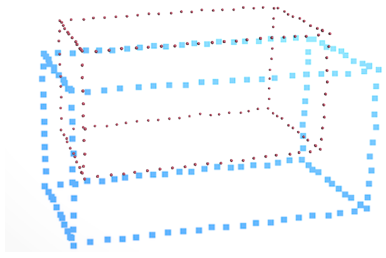
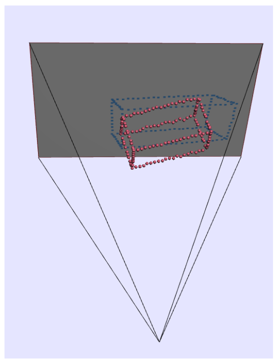
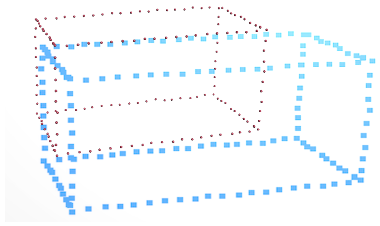
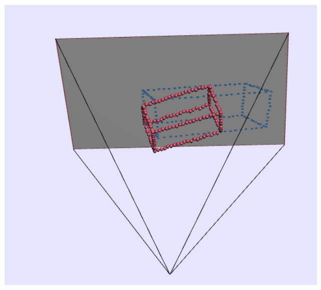
- cx and cy will not influence the shape of the 3D object, they will just generate displacement of the object in the x and y directions.

- When we adjust the cx or cy, the errors are relatively small compared with the error caused by fx/fy.

Table 4: Effect of errors in intrinsic parameters (cx and cy)

cx	cy	result		error
270	190			3.58218
320				2.86051
370				3.63714

270	240			2.48149
320				1.95221
370				3.24932
270	290			1.57243

320				1.18974
370				2.62307

5. Contributions

In order to better understand the camera calibration process, each of us completed the code independently this time, and then combined the good parts of the code into the final result.

Linjun Wang(33%)
-Estimate fundamental matrix F
-Description of the methodology
-Demonstration result

Zhenyu Liu(34%)
- Implement two triangulation methods
- Evaluation of results
- Verify the validity of data

Ziyan Wu(33%)
-Recover relative pose
-Determine the 3D coordinates
- Intrinsic parameters