Computer Vision

Assignment 2 Report
By Chris Jimenez

I - Epipolar Geometry

Introduction
For this part of the assignment, given the two camera arrangements shown in Figure 1, the following questions were answered:

1) The camera matrices given this arrangement.
2) Compute the epipolar line.
3) Determine where potential correspondences to the left image pint (0,0) can lie in the right image.
4) Describe the rotation and translation that should be applied to the left camera that would make the epipolar lines in the two images horizontal.

The work for the above questions can be seen in the following pages.

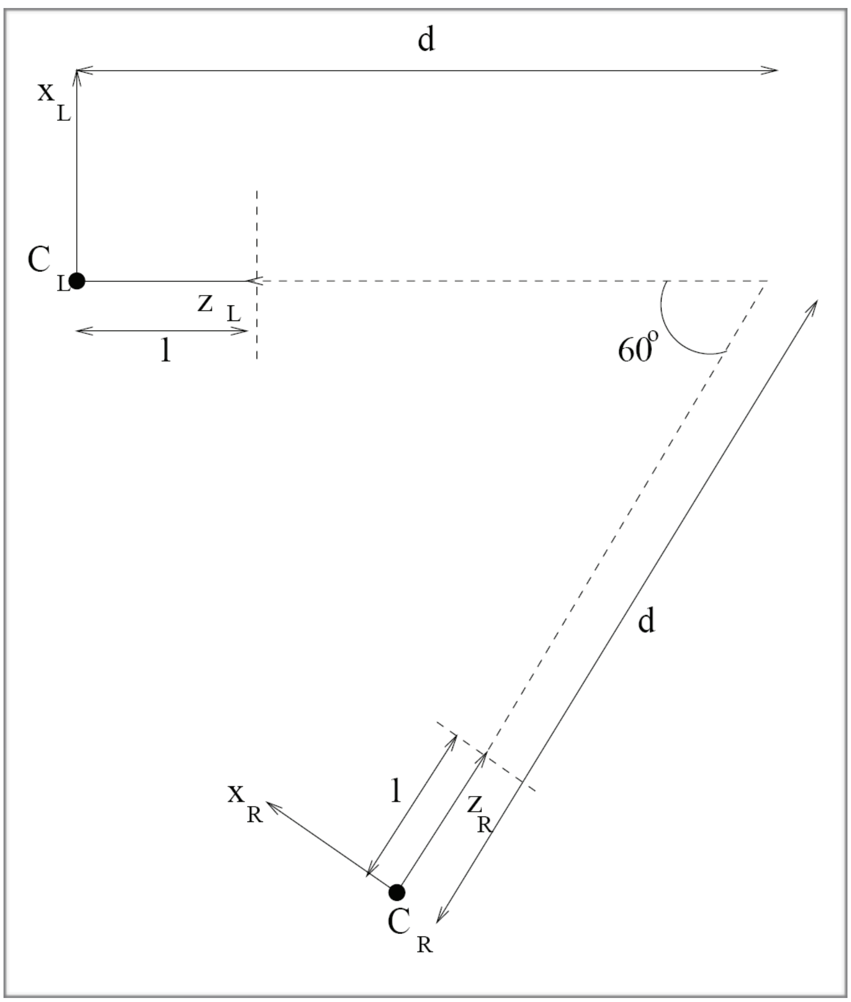

Figure 1: Camera arrangement figure for part 1.

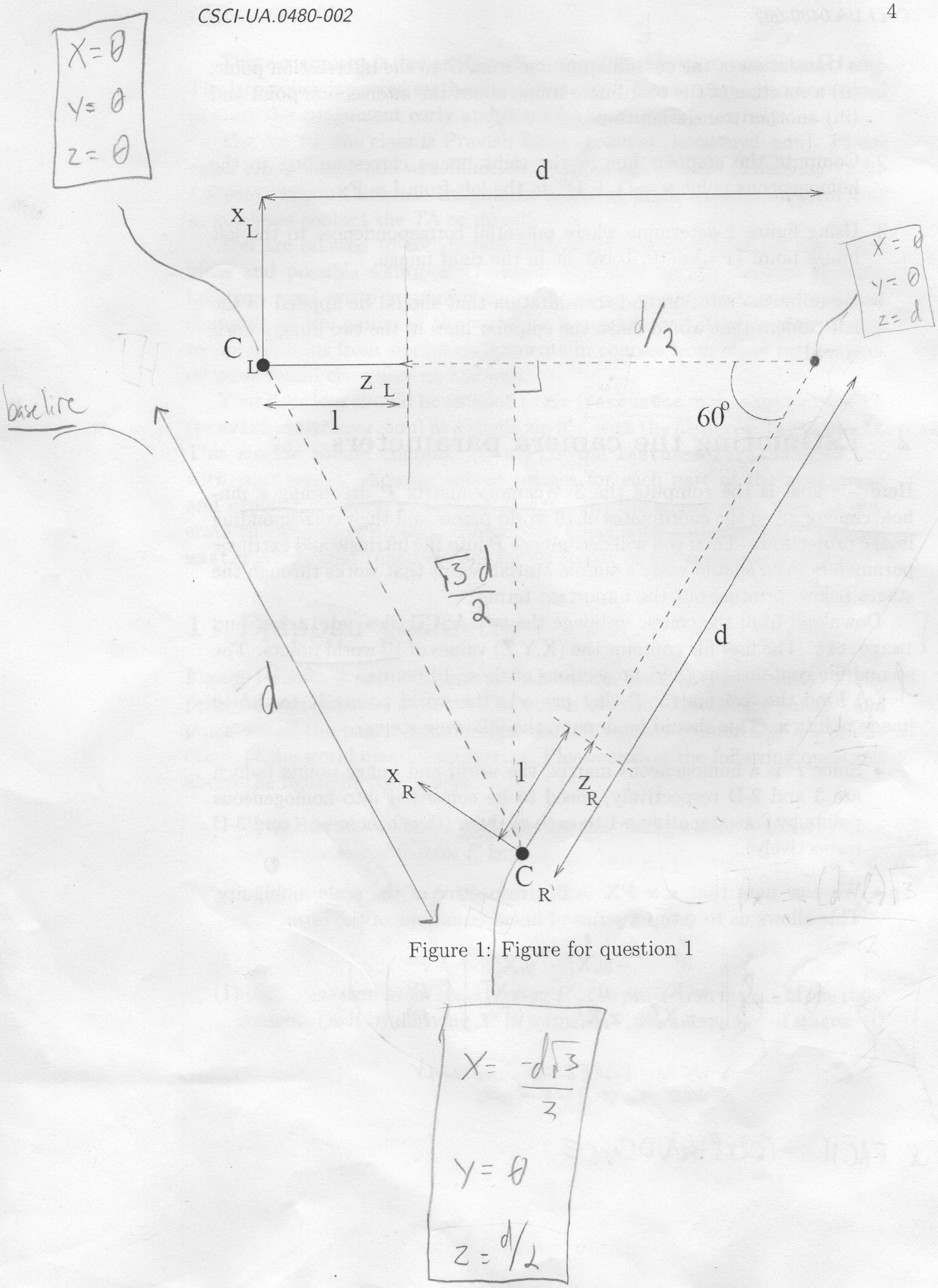

Figure 1: Figure for question 1

Christopher Jimenez

Computer Vision Assignment # 1

Part 1

1

$$F = [K_R t]_\times K_R R K_L^{-1} \qquad \text{Verify} \quad F = \begin{bmatrix} 0 & -d/2 & 0 \\ -d/2 & 0 & -\sqrt{3}d/2 \\ 0 & \sqrt{3}d/2 & 0 \end{bmatrix}$$

Left camera

$$K_L = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad P_L = K_L [I|0] = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \overset{\text{world point}}{\begin{pmatrix} 0 \\ 0 \\ d \\ 1 \end{pmatrix}}$$

Right camera

$$K_R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad R = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad t = \begin{bmatrix} \sqrt{3}d/2 \\ 0 \\ d/2 \end{bmatrix}$$

$$K_R t = \begin{bmatrix} \sqrt{3}d/2 \\ 0 \\ d/2 \end{bmatrix} \qquad P_R = K_R [R|t]$$

$$P_R = \overset{R}{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}} \left[\begin{array}{ccc|c} -1 & 0 & 0 & \sqrt{3}d/2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d/2 \end{array}\right]^{t} \begin{pmatrix} 0 \\ 0 \\ d \\ 1 \end{pmatrix}$$

$$F = [K_R t]_\times K_R R K_L^{-1}$$

$$F = \begin{array}{cccc} [K_R t]_\times & K_R & R & K_L^{-1} \\ \begin{bmatrix} 0 & -d/2 & 0 \\ d/2 & 0 & -\sqrt{3}d/2 \\ 0 & \sqrt{3}d/2 & 0 \end{bmatrix} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} & \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} & \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \end{array}$$

$$F = \begin{bmatrix} 0 & -d/2 & 0 \\ d/2 & 0 & -\sqrt{3}d/2 \\ 0 & \sqrt{3}d/2 & 0 \end{bmatrix} \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\boxed{F} = \begin{bmatrix} 0 & -d/2 & 0 \\ -d/2 & 0 & -\sqrt{3}d/2 \\ 0 & \sqrt{3}d/2 & 0 \end{bmatrix}$$

2) Epipolar Line

$$x = (1,1,1)^T$$

Let l_R be the epipolar line for the right camera

$$l_R = Fx$$

$$\boxed{l_R} = \begin{bmatrix} 0 & -d/2 & 0 \\ -d/2 & 0 & -\sqrt{3}d/2 \\ 0 & \sqrt{3}d/2 & 0 \end{bmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{bmatrix} -d/2 \\ \frac{-d}{2} + \left(-\frac{\sqrt{3}d}{2}\right) \\ \frac{\sqrt{3}d}{2} \end{bmatrix} = \begin{bmatrix} -d/2 \\ -\frac{(d+\sqrt{3}d)}{2} \\ \frac{\sqrt{3}d}{2} \end{bmatrix}$$

(3) *see Figure*

(4) For the epipolar lines in the left camera and right camera to be parallel, the cameras need to be translated without rotation. Which means that the left camera cannot be rotated in order for the epipolar lines to be parallel.

II - Estimating Camera Parameters

For this part of the assignment, the goal was to compute the 3 by 4 projection matrix P of a pinhole camera given 10 world points and their corresponding image projections. Once P is computed, it is decomposed to its intrinsic and extrinsic parameters. The input files used were the `world.txt` file, which contain the (X,Y,Z) values of each work points, and `image.txt` file, which contains the (x,y) image points of each corresponding world point. The homogenous world points and image points can be see below.

```
world_points =

0.8518 0.5579 0.8162 0.7037 0.7134 0.1722 0.0490 0.2861 0.1310 0.8477
0.7595 0.0142 0.9771 0.5221 0.2280 0.9688 0.7553 0.2512 0.9408 0.2093
0.9498 0.5962 0.2219 0.9329 0.4496 0.3557 0.8948 0.9327 0.7019 0.4551
1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000 1.0000


image_points =

5.1177 5.5237 7.1631 5.2222 5.6048 13.5949 8.7345 6.2243 9.7476 5.0903
4.7654 3.8703 7.3594 4.4280 4.6748 10.0522 5.5642 3.9082 6.9042 4.5509
1.0000 1.0000 1.0000 1.0000 1.0000 1.0000  1.0000 1.0000 1.0000 1.0000
```

Firstly, since P is a homogenous matrix, the world points and the image points need to be converted to homogenous points by adding a row of 1 for both sets of data. This makes the image points 3d coordinates and the world points 4d coordinates. The computed P matrix can be seen below.

```
P =

43.8409    4.4986    0.9831    0.1762
11.3986    3.9213    0.5129    0.0675
 9.1004    1.1845    0.2300    0.0000
```

Now the the projection center of the camera C must be computed. Since PC = 0, C lies in the null space of P. Therefore C can be computed by getting the SVD of P and picking the vector corresponding to that null space. The C parameters can be seen below.

```
C =

-0.2016
-1.4547
15.4695
```

III - Structure from Motion(SfM)

For this part of the assignment, an affine structure from motion algorithm was programmed in Matlab. A 3D 2 by 600 by 10 matrix, holding the (x, y) coordinates of 600 world points which are projected onto the image plane of the camera in 10 different locations, were retrieved from the given `sfm_points.mat`. An affine reconstruction was applied to these 10 different data sets. The translation and camera location for the first camera is shown below. Additionally, the (X, Y, Z) coordinates of the first 10 world points are shown in the table below.

```
t1 =

1.0e-16 *

  0.5496
 -0.0703


M1 =

-7.9184
 1.6305
```

World point	X	Y	Z
1	0.0058	0.0646	-0.025
2	5.7610E-04	0.0689	-0.0346
3	-0.0429	0.0633	0.0286
4	0.0475	0.0490	-0.0126
5	-0.0421	0.0679	0.0118
6	0.0596	0.0461	-0.0144
7	0.0091	0.0600	-0.0123
8	0.0104	0.0460	0.0353
9	-0.0259	0.0570	0.0334
10	0.0175	0.0405	0.0473

V - Camera Calibration from a Set of Images

For this part of the assignment a standard toolbox used to calibrate cameras was used for the given 20 images. The toolbox used is the Camera Calibration Toolbox written by Jean-Yves Bouguet. The result of the corner detection for all 20 images can be seen below in Figure 2.

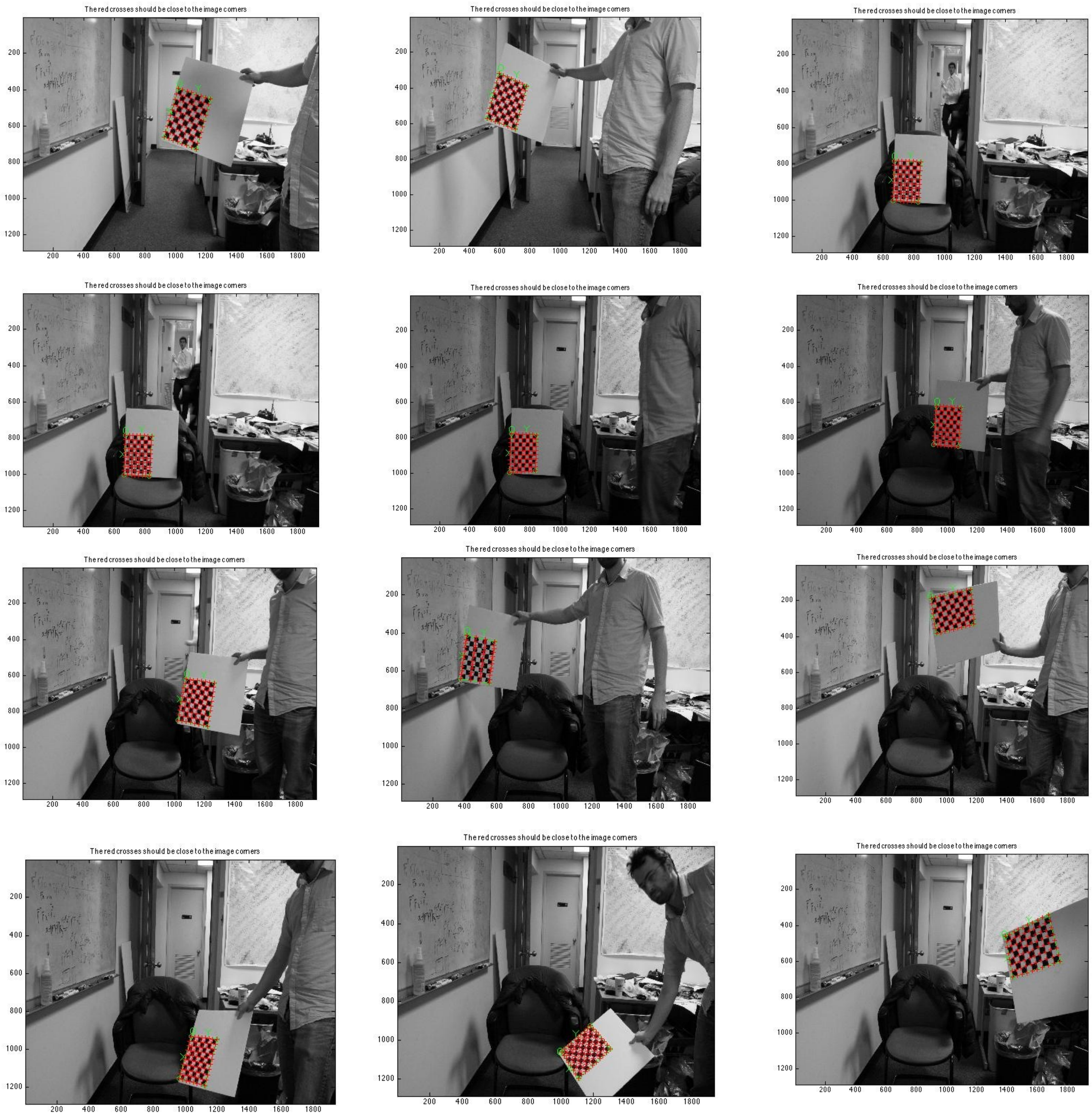

Figure 2: Images 1-20 showing the corner detection of each grid.

The calibration parameters calculated can be seen below(excluding the skew and distortion values) , followed by the 3 by 3 intrinsic parameters matrix.

```
Focal Length:          fc = [ 354.73630   354.73630 ]
Principal point:       cc = [ 967.50000   643.50000 ]

mX = ((27.5*10E-3) * 9)/31 = 0.07983870967
mY = ((27.5*10E-3) * 7)/31 = 0.06209677419

ax = mX * 354.73630 = 28.3216884677
ay = mY * 354.73630 = 22.0279799194

Bx = mX * 967.50000 = 77.2439516057
By = mY * 643.50000 = 51.3762096726
```

$$A = \begin{bmatrix} 28.322 & 0 & 77.244 \\ 0 & 22.028 & 51.376 \\ 0 & 0 & 1 \end{bmatrix}$$

The visualization of the position of the chart relative to the camera for each frame can be seen in Figure 3 below.

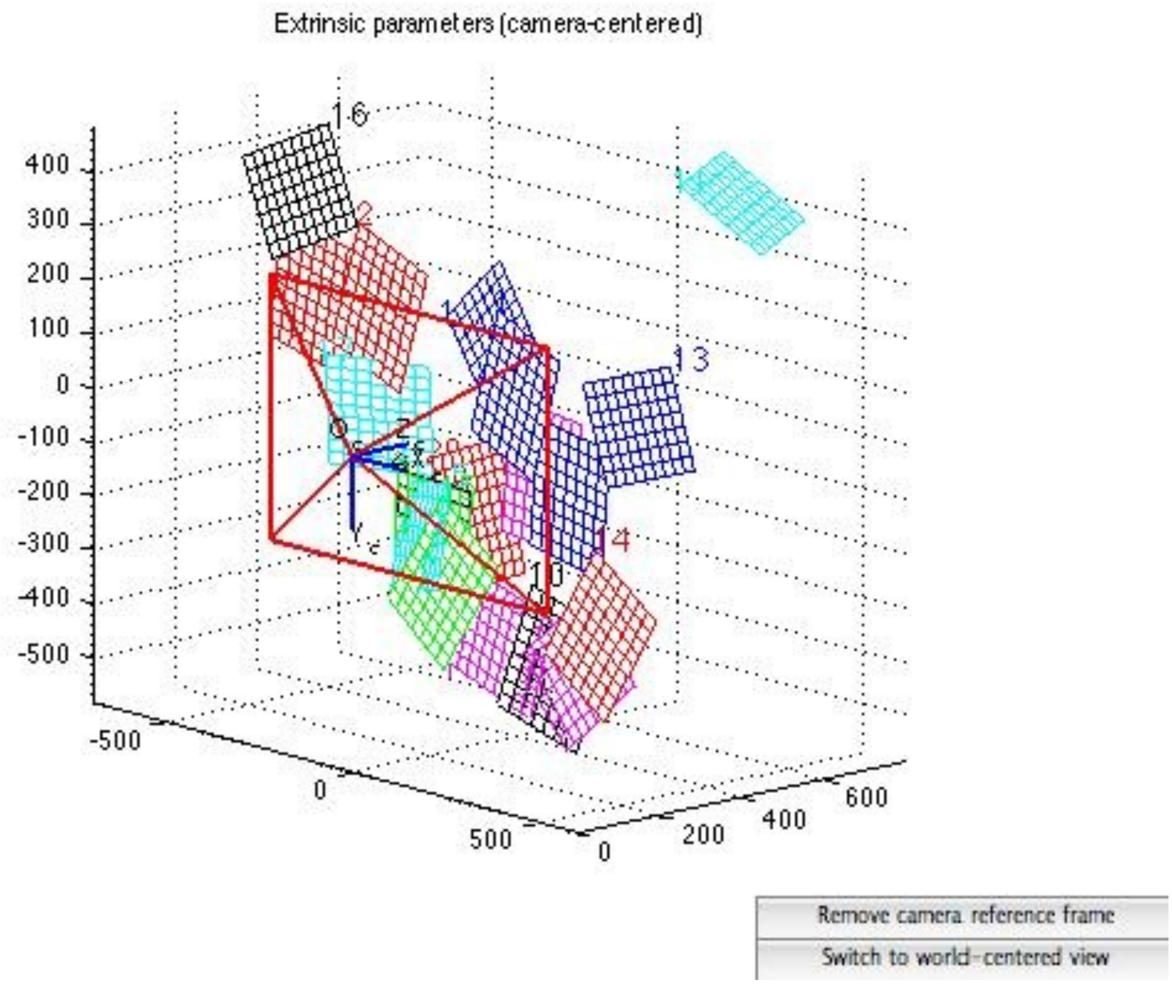

Figure 3: Visualization of the Extrinsic parameters.