CMPE 264 - Project Assignment 2

Yanan Xie
yaxie@ucsc.edu

Ziqiang Wang
zwang232@ucsc.edu

1. Camera radiometric calibration

The first step of this project is to calibrate the camera. Since we are going to implement the plane-sweeping multiview stero, we have to know the intrinsic parameters and the distortion coefficients of the camera we use. The camera we used in the project is the Cannon 5D, after the calibration we use the same camera to take multiple views and implemented the plane-sweeping muliview stero.

We learned from the tutorial that the distortion is composed of two differernt parts, radial distortion and tangential distortion. As for the radial distortion, we have

$$x_{distorted} = x(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$
$$y_{distorted} = y(1 + k_1 r^2 + k_2 r^4 + k_3 r^6)$$

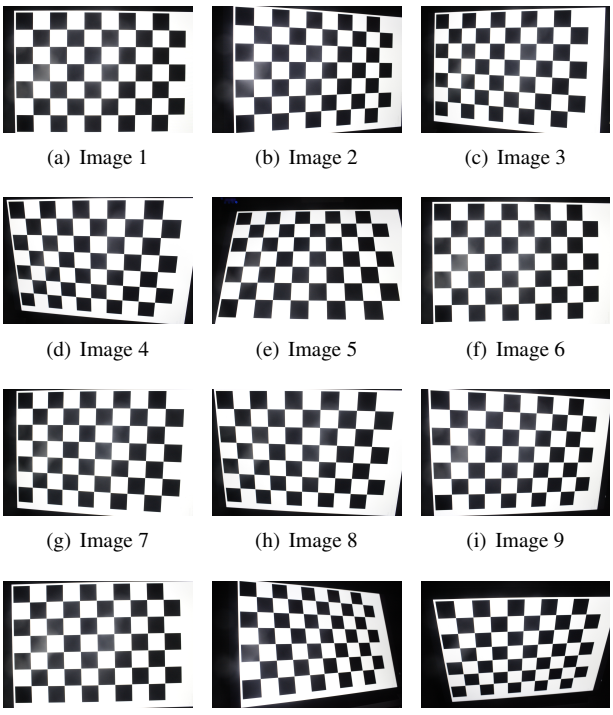
As for the tangental distortion, we have

$$x_{distorted} = x + [2p_1 xy + p_2(r^2 + 2x^2)]$$
$$y_{distorted} = y + [p_1(r^2 + 2y^2) + 2p_2 xy]$$

So the distortion coefficients we need to find are k_1, k_2, p_1, p_2, p_3.

We also learned from the tutorial that the camera intrinsic matrix consist four degrees of freedoomthis instead of three. These are more than what we have learned about the distortion and the camera matrix, but we learned it and take it for the following project parts.

We need a perfect flat surface of the chessboard to be as the standard world coordinate reference system. However, we tried to printed the chessboard picture and glue it onto the flat desk surface, then we take pictures and compute, but the outcome of the calibration is not idea to do undistortion since the printed piece of paper is not perfect flat. So finally we show the chessboard on the full-screen of our Macbook and take pictures and then compute, the intrinsic parameters we got this time is better, this is understandable since the screen is much more perfect flat than the printed paper, and we got much better result in the following undistortion step. There are two key points of taking the chessboard pictures we concluded as we try this process, first is we should full fill each picture we take with the whole chessboard, the second is the surface of the chessboard has to be prefect flat so that all the Z coordinate of the chessboard point respect to the world reference system could be the same. All of these pictures we took for the chessboard is as shown in Figure 1.



(a) Image 1 (b) Image 2 (c) Image 3

(d) Image 4 (e) Image 5 (f) Image 6

(g) Image 7 (h) Image 8 (i) Image 9

(j) Image 10 (k) Image 11 (l) Image 12

Figure 1. Chessboard Pictures taken from different view points

As for the implementation of how to get the intrinsic paremeters, we adapted the $findChessboardCorners()$ fucntion to find the chessboard corners in each picture we took, if the returned boolean value show that the corners was found in the picture then we add the corners into the $imgpoints[]$ list and accordingly make a pair of the world reference system coordinate in the $objpoints[]$ list. Then we pass the $imgpoints[]$ list and the $objpoints[]$ list to the $calibrateCamera()$ function and we got the camera matrix and the distortion coefficients.

The camera matrix K is

(a) Image 1 (b) Image 2 (c) Image 3

Figure 2. Pictures taken from different distances and directions

$$\mathbf{K} = \begin{bmatrix} 2.839\text{e}{+03} & 0.000\text{e}{+00} & 1.389\text{e}{+03} \\ 0.000\text{e}{+00} & 2.836\text{e}{+03} & 9.369\text{e}{+02} \\ 0.000\text{e}{+00} & 0.000\text{e}{+00} & 1.000\text{e}{+00} \end{bmatrix}$$

We also have the distortion coefficients after the computation, shown as below

$k_1 = -0.12244297$, $k_2 = 0.20499093$, $p_1 = 0.00035033$, $p_2 = -0.00058871$, $p_3 = -0.13275741$

After we obtained all the coefficients we need, we need to check the quality of the calibration by computing the mean square error. The out come of the square error of our project is optimal and the total error value is zero. This could be shown by the outcome of the $part1.py$

2. Take the pictures

We use the same camera and lens take 3 pictures as shown in Figure 2. We placed some objects in the front and also left enough distant background.

3. Compute the essential matrix of all three camera pairs

We first undistort pictures with calibration parameters as shown in Figure. There were 25 pixels loss in height and 12 pixels loss in width. Straight lines in those images confirms the calibration was done correctly.

In order to get pixel pairs over 3 camera pairs, we built a tool to allow us to pick those pixel pairs by hand. Figure 3 shows the interface of our tool.

For each camera pair, we picked 17 to 18 pixel pairs to provide enough data calculating essential matrices and fundamental matrices. Figure 5 shows the pixel pairs we picked over 3 camera pairs.

The essential matrices we calculated are

$$\mathbf{E}_{12} = \begin{bmatrix} -2.944\text{e}{-07} & -1.993\text{e}{-06} & 2.051\text{e}{-02} \\ 3.242\text{e}{-06} & -9.280\text{e}{-07} & -2.045\text{e}{-02} \\ -2.030\text{e}{-02} & 1.995\text{e}{-02} & 1.000\text{e}{+00} \end{bmatrix}$$

(a) Image 1 (b) Image 2 (c) Image 3

Figure 3. Undistorted pictures.

Figure 4. Pictures taken from different distances and directions

$$\mathbf{E}_{13} = \begin{bmatrix} 4.323\text{e}{-08} & 1.144\text{e}{-06} & -1.610\text{e}{-03} \\ -2.850\text{e}{-07} & 2.763\text{e}{-07} & -7.243\text{e}{-03} \\ 5.939\text{e}{-04} & 5.592\text{e}{-03} & 1.000\text{e}{+00} \end{bmatrix}$$

$$\mathbf{E}_{23} = \begin{bmatrix} 1.608\text{e}{-07} & 2.496\text{e}{-07} & -2.468\text{e}{-03} \\ -1.274\text{e}{-07} & 1.268\text{e}{-08} & -1.230\text{e}{-03} \\ 1.877\text{e}{-03} & 9.439\text{e}{-04} & 1.000\text{e}{+00} \end{bmatrix}$$

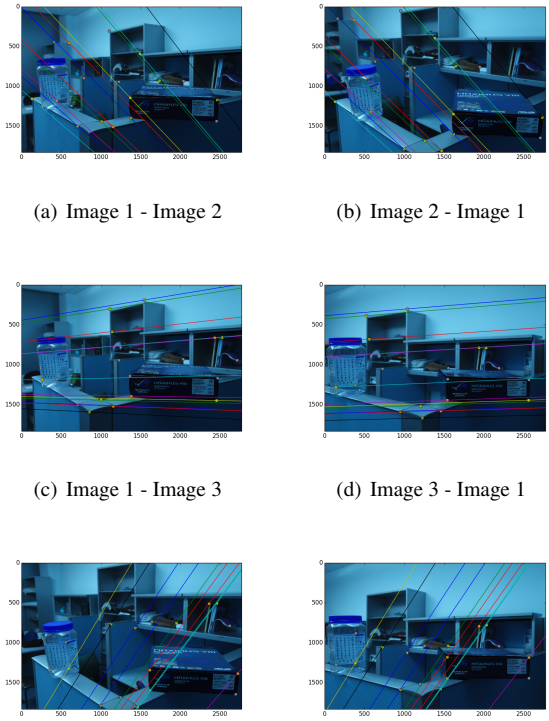
4. Find the extrinsic parameters

5. Rescale the translation vector

After we have found the extrinsic parameters, we obtained the rotation matrices $\mathbf{R}_1^2$, $\mathbf{R}_2^3$, $\mathbf{R}_1^3$, we also obtained the vector $\mathbf{r}_{12}^1$ representing camera 2 seen by the camera 1, expressed in reference of the camera 1. The same to vector $\mathbf{r}_{23}^2$ and vector $\mathbf{r}_{13}^1$ we obtained.

In this part, the first work we did is to express the vectors in terms of a common reference frame. We choose the camera 1 reference system as the common reference system, we obtain $\mathbf{r}_{23}^1$ by doing matrix multiplication on transpose matrix of $\mathbf{R}_1^2$ and vector $\mathbf{r}_{23}^2$. We can also have $\mathbf{r}_{31}^1 = -\mathbf{r}_{13}^1$. Now we have $\mathbf{r}_{12}^1$, $\mathbf{r}_{31}^1$ and $\mathbf{r}_{31}^1$, all the 3 vectors are to the same common reference system.

The next work we do on is to convert all the 3 vectors to unit form. We divided all the 3 vector by their length respectively. Now we have $\mathbf{r}_{12unit}^1$, $\mathbf{r}_{23unit}^1$ and $\mathbf{r}_{31unit}^1$.



(a) Image 1 - Image 2 (b) Image 2 - Image 1

(c) Image 1 - Image 3 (d) Image 3 - Image 1

(e) Image 2 - Image 3 (f) Image 3 - Image 2

Figure 5. Hand-pick pixel pairs (Inliers are yellow while outliers are gray) and corresponding epipolar lines.

After the mathematics performed by the project guidance, we have the formula $|\mathbf{r}^1_{12unit} + \beta\mathbf{r}^1_{23unit} + \gamma\mathbf{r}^1_{31unit}|$. In an ideal world, the $|\mathbf{r}^1_{12unit} + \beta\mathbf{r}^1_{23unit} + \gamma\mathbf{r}^1_{31unit}|$ should equal to 0, however in practice due to the error and noise, we just can minimize the value as small as possible. We use the $minimize()$ function imported from $scipy.optimize$ to do the minimization and find the β and γ value such that the $|\mathbf{r}^1_{12unit} + \beta\mathbf{r}^1_{23unit} + \gamma\mathbf{r}^1_{31unit}|$ value is minimized.

The outcome of the optimization gave us the value $\beta = 0.000822176591359$ and $\gamma = 1.00011889642$, we can also see that by this time $|\mathbf{r}^1_{12unit} + \mathbf{r}^1_{23unit} + \mathbf{r}^1_{31unit}| = 0.999195082441$ and $|\mathbf{r}^1_{12unit} + \beta\mathbf{r}^1_{23unit} + \gamma\mathbf{r}^1_{31unit}| = 0.000115952725309$, the second is smaller as shown.

6. Acknowledgment

We did all the project together. However, Ziqiang Wang is mainly responsible for Section **??** and Section **??** while Yanan Xie is responsible for the rest parts of this project including providing his professional photography devices.

Thanks to LaTeX for providing such a great document preparation system. And many appreciates to the authors of Python, numpy, sklearn, OpenCV and matplotlib.