Computer Vision Project 2

Yuan Yang (yyang175@ucsc.edu) Weiting Zhan(wzhan83@ucsc.edu)

Sunday November 28 2018

1 Camera radiometric calibration(intrinsic parameters)

1.1 Chessboard pattern

Take 11 pictures of chessboard pattern from different viewing angles, as shown in Figure

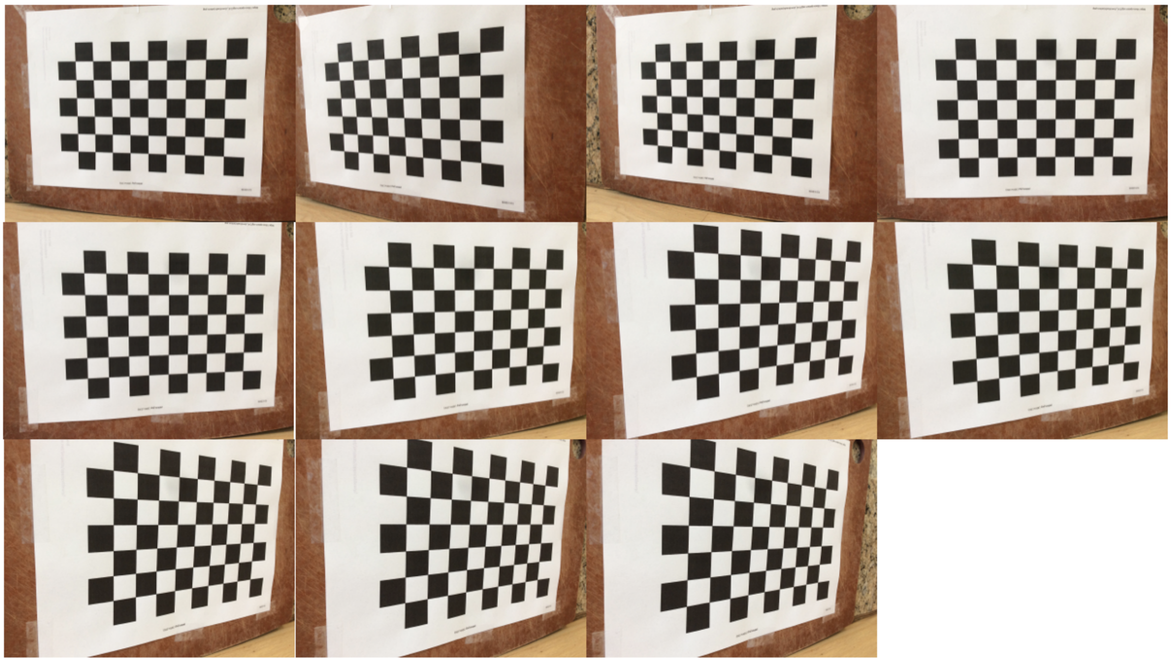

Figure 1: chessboard pattern

1.2 The intrinsic matrix K

The intrinsic matrix K we got:

$$K = \begin{bmatrix} 2.35823484e+03 & 0.00000000e+00 & 1.27505685e+03 \\ 0.00000000e+00 & 2.35363433e+03 & 9.73403504e+02 \\ 0.00000000e+00 & 0.00000000e+00 & 1.00000000e+00 \end{bmatrix} \quad (1)$$

1.3 The radial distortion coefficients

The radial distortion coefficients is $\begin{bmatrix} 0.10990381 & -0.01088416 & -0.00567028 & -0.00663648 & -1.30729001 \end{bmatrix}$

1.4 The re-projection mean square error

The re-projection mean square error is 0.217.

2 Take the pictures

We took a pair of images in where the maximum distance is finite. Two images contains the same parts of the scene. The objects are with texture and at different distances.

Figure 2: Image pair

3 Compute the relative camera pose R_L^R and r^R

3.1 Epipolar lines on the images

We undistort the images using camera calibration parameters we got, get features points using SIFT, match the features points and refine the matches, calculate the fundamental matrix and caclulate the epipolar lines in two images.

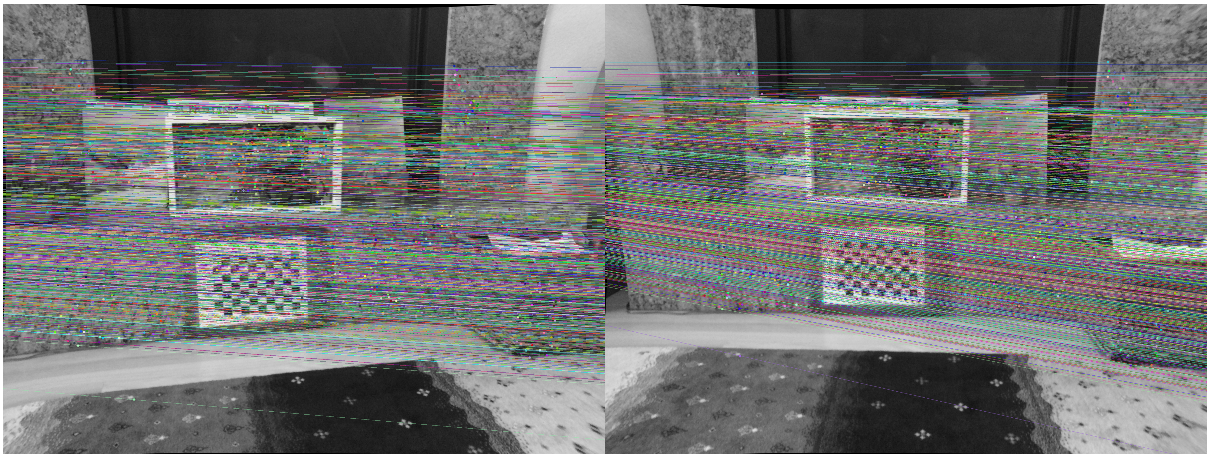

Figure 3: Epipolar lines on the images

3.2 R_r^R and r^R matrix

We calculate the essential matrix E and decompose E to get R_r^R and r^R. There are 4 possible combination of the pose matrix. We use all the combination to calculate the point in world reference system. The combination gives the positive Z(depth) chose as the right R_r^R and r^R.

There are 4 possible combination for the projection matrix.

Case 1: projMatr1 = K [I 0] projMatr2 = K[R1 t]

Case 2: projMatr1 = K[I 0] projMatr2 = K[R1 -t]

Case 3: projMatr1 = K[I 0] projMatr2 = K[R2 t]

Case 4: projMatr1 = K[I 0] projMatr2 = K[R2 -t]

Figure 4: 4 possible solution

3.3 re-projected feature points on the first image

$$R_r^R = \begin{bmatrix} 9.68320962e-01 & -5.99263511e-02 & 2.42411525e-01 \\ 6.19095553e-02 & 9.98081604e-01 & -5.64872089e-04 \\ -2.41912633e-01 & 1.55545672e-02 & 9.70173352e-01 \end{bmatrix}$$
$$r^R = \begin{bmatrix} 0.91936328 \\ 0.07852536 \\ -0.38549309 \end{bmatrix}$$

Figure 5: Reprojected Points

We re project the 3D points into the first image and we found that positions of original points are the same as positions of corresponding 3D points, which means that we get the correct depth values.

4 Plane-sweeping stereo

4.1 d_{min} and d_max

Extract the third column of triangulating points, we get the minimum depth (d_{min}) is 3.368 and maximum depth(d_{max}) is 5.98. Then generate the uniformly generate 20 depth between the minimum depth and maximum depth.

4.2 one warped image per lane

We use the left camera's referees system as world reference system. So for left camera rotation matrix is the identity matrix and transition matrix is zero. For right camera, we use the result from part 3.2. We calculate the homogeneous matrix follows equation shown below.

$$H = K \begin{bmatrix} I & 0 \\ n^T & d \end{bmatrix} \begin{bmatrix} R2 & t \\ n^T & d \end{bmatrix}^{-1} K^{-1}$$

According to the 20 difference depth, we got 20 different homography matrix, then apply these homography matrix to the original picture, we got the warped image. As shown in figure 6.

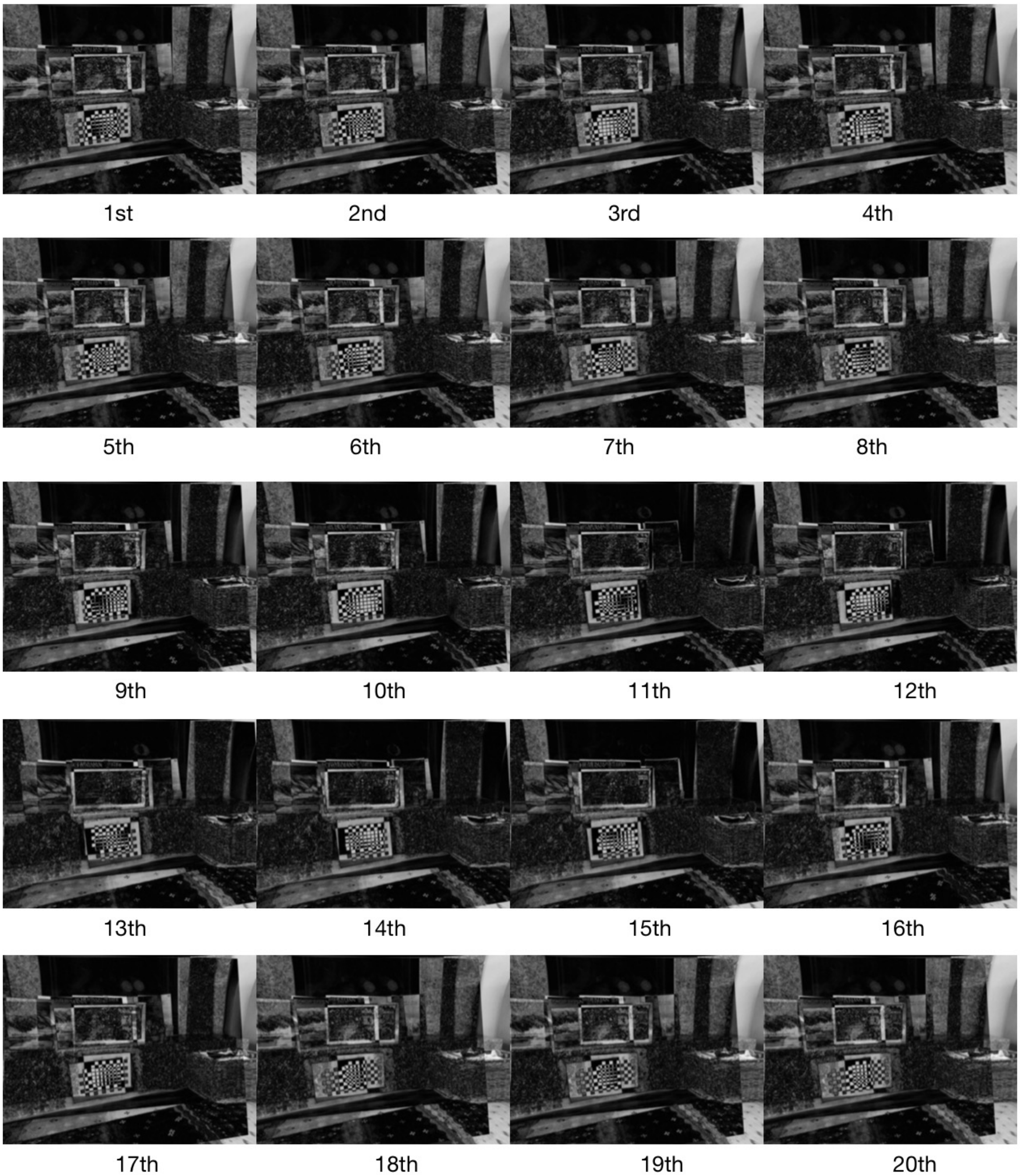

Figure 6: Warp Image

4.3 depth image

For each pixel, we use the depth of the least brightness value of the blockfiltered images, then scaling them in to the range of 0 to 255. assign d_{min} with 255, the d_{max} with 0. We got the depth image as shown figure 7.

Figure 7: depth image

5 Code

Weiting Zhan implement the homework using Python.

Github:https://github.com/WaitingZhan/Plane-sweeping-stereo