**Building 3D image from 2D scenes**

**By Cao Tianwei**

1.Principle of the project

The method I use to build 3D image from 2D scenes comes from binocular stereo imaging, which uses two cameras to simulate the process of human eyes recognizing images and constructing 3D scenes in their minds. In experiment environment, it takes four steps to build the 3D image.

To start with, it is necessary to calibrate the two cameras. By doing so, we can obtain the parameters of the two cameras, which plays an important role in the next steps.

After that, we take a picture of our target from each camera and these two pictures are our stereo images. Stereo images are of important since we use them to calculate the depth of our target, however, the images we get from our cameras are often distorted. It is because we can not ensure that the imaging planes of the two cameras are in the same plane. In this case, we should rectify our stereo images before we make use of them.

Then, we extra the characteristic points from both images and match them with one another. For example, if point A is from stereo image 1 and point A’ is from stereo image 2. If and only if point A and point A’ represent the same position on target object, we say A and A’ is a pair of matching points. By doing so, we describe the characteristic of our target.

Finally, we use characteristics of our target to build 3D image.

2.The process of experiment

2.1 Camera calibration

The purpose of camera calibration is to get the internal and external parameters of the camera. Here we do binocular calibration. The concept of coordinate system is introduced before calibration.

2.1.1 Image coordinate system（u，v）

This coordinate system is based on pixels, which represents the position of pixels in the image.

2.1.2 Physical coordinate system of image（x，y）

This coordinate system is established in the pixel coordinate system, and the position of the pixel in the image is represented by physical units. Point *C1* is the intersection of the camera optical axis and the image plane, the coordinate of point C in the (u,v) coordinate system is (u0,v0) and the unit length of the two axes of the coordinate system is *dx*, *dy*. In this case, the Physical coordinate system of image (x,y) has the following transformation relationship with the Image coordinate system (u, v) :

, (2-1)

 (2-2)

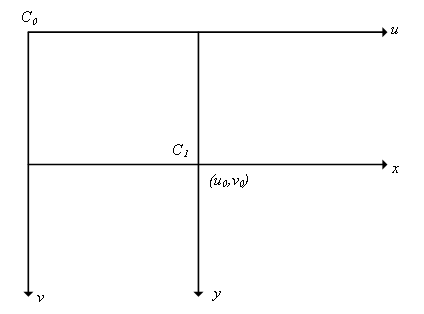


Fig.2-1 Physical coordinate system of image（x，y）

2.1.3 Camera coordinate system

This coordinate system is used to represent the projection relationship between the object and the image plane in three-dimensional space.

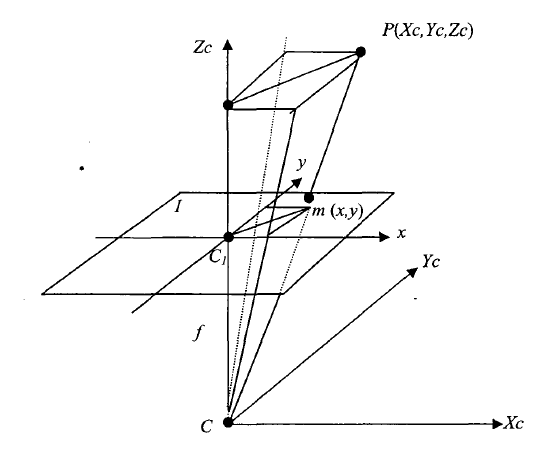


Fig.2-2 The Camera coordinate system

As shown in the figure, *f* is the focal length of the camera and (XC,YC,ZC) is the coordinate of point P in the Camera coordinate system. Therefore, we can obtain the relationship between (XC,YC,ZC) and (x,y,z) :

 (2-3)

2.1.4 World coordinate system

This coordinate system is used when placing the camera in a scene. It has following relationship with Camera coordinate system:

 (2-4)

Where *R* is the rotation matrix and *t* is the 3D translation.

Simultaneous formula(2-1),(2-2),(2-3),(2-4), We can deduce the following relations:

 (2-5)

2.2 Binocular calibration

From the above transformation relations, *M1* matrix represents the internal parameters of the camera, and *M2* matrix represents the external parameters of the camera. The parameters of the camera can be obtained by determining*M1* and *M2*. Because we use two cameras to shoot the same object at different angles for calibration, we also need to determine the relative position between the two cameras.

In my experiment, I placed two smart phones in the same plane as my cameras and I used Stereo image calibrator in matlab to calibrate the two cameras.

This app can easily calibrate the two cameras by scan pairs of pictures taken by the cameras. Normally, the app expects at least ten pairs of pictures to calibrate. However, since I did this experiment with a lack of professional environment, I only use three pairs of pictures to calibrate the cameras.

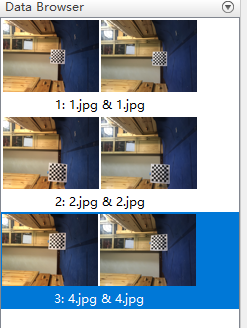


Fig.2-3 three pairs of pictures from two camera

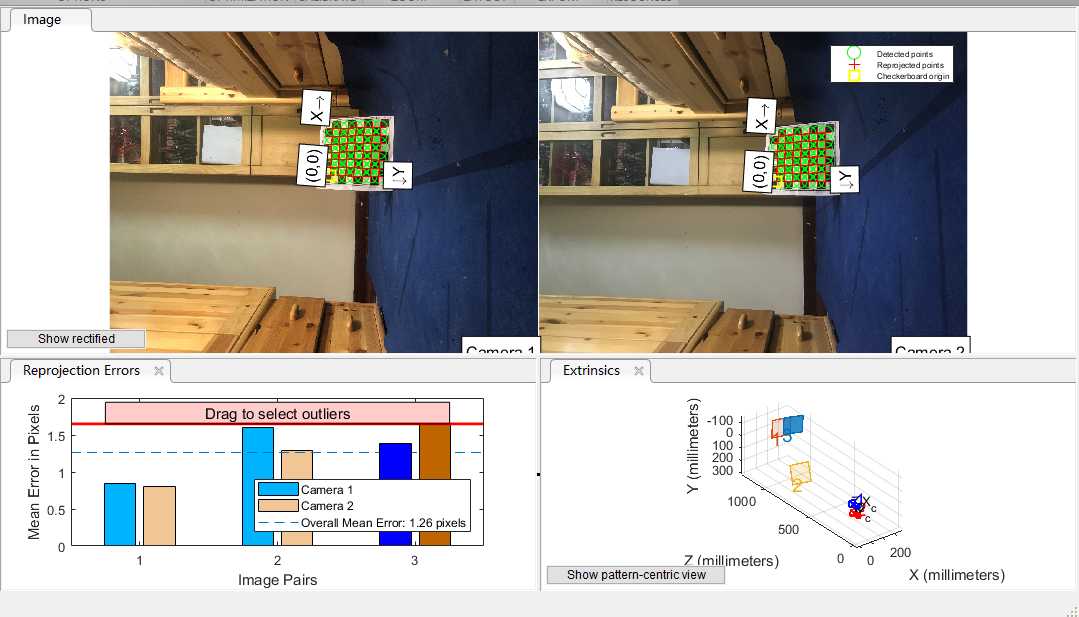


Fig.2-4 The result of calibration

As shown in Fig. ,Cameraparameters1 is a separate calibration parameter for camera1 and cameraparameters2 is a separate calibration parameter for camera2. StereoParams .Rotation of Camera 2 shows the rotation of camera 2 relative to camera 1, Translation of Camera 2 shows the translation of camera 2 relative to camera 1.

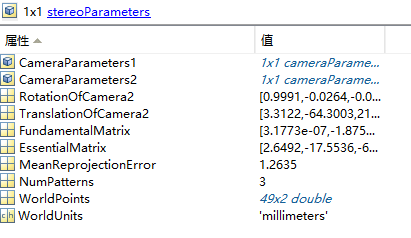


Fig.2-5 The result of calibration-stereoParameters

From Fig. ,we can find Radial Distortion,Tangential Distortion and Inreinsic Matrix for each camera.

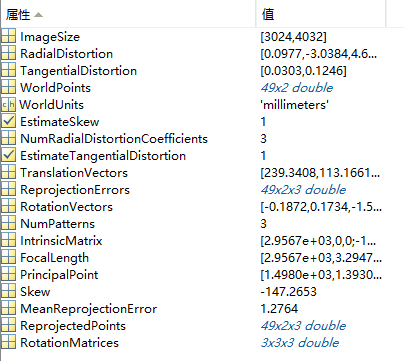


Fig.2-6 The result of calibration-CameraParameters1

2.2 Image rectification

At this time, although we have obtained the parameters of the camera, we still need to carry out distortion correction before matching. This is because we want the imaging planes of the two cameras to be coplanar so as to reduce the amount of calculation.

In my experiment, I called rectify StereoImages functions in matlab to rectify the two images, this function takes the two images as independent variables and uses the camera parameters we got from 2.1 as constraint condition.



Fig.2-7 Stereo image from camera1



Fig.2-8 Stereo image from camera2

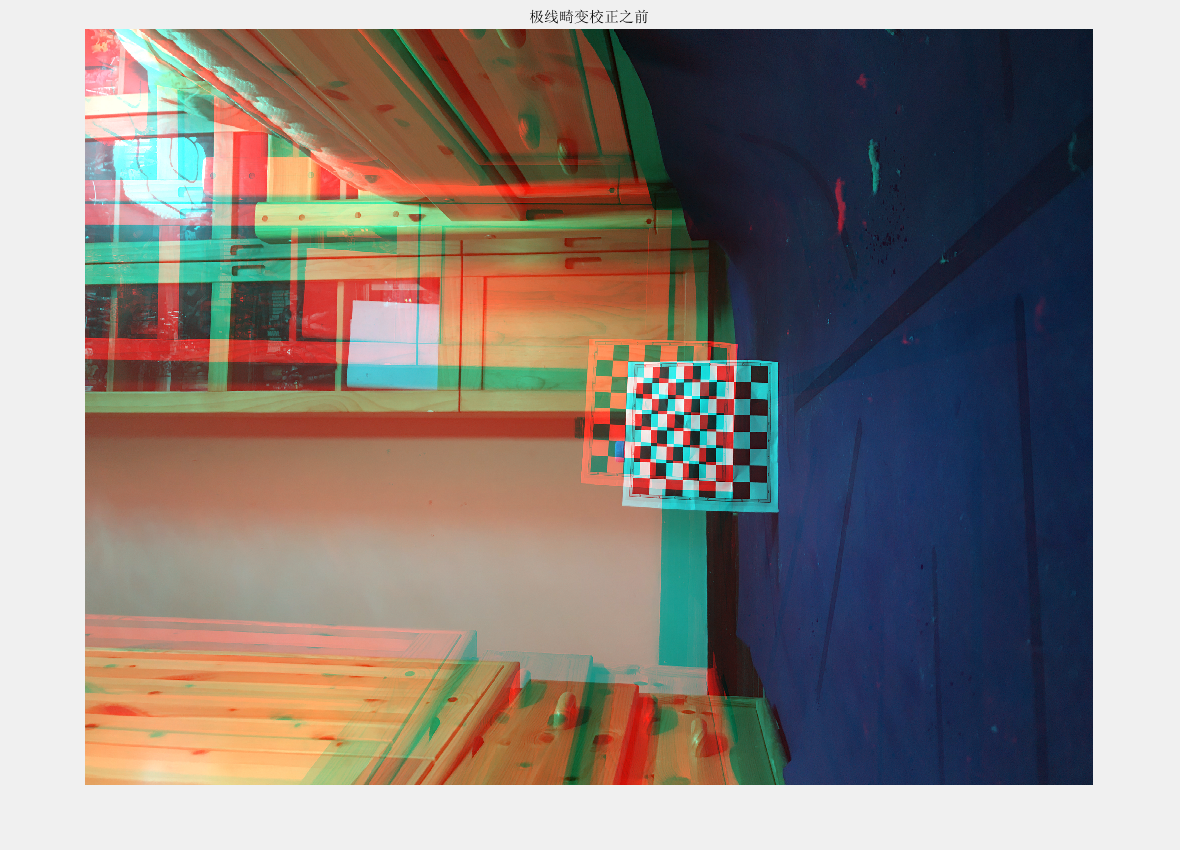


Fig.2-9 before rectification

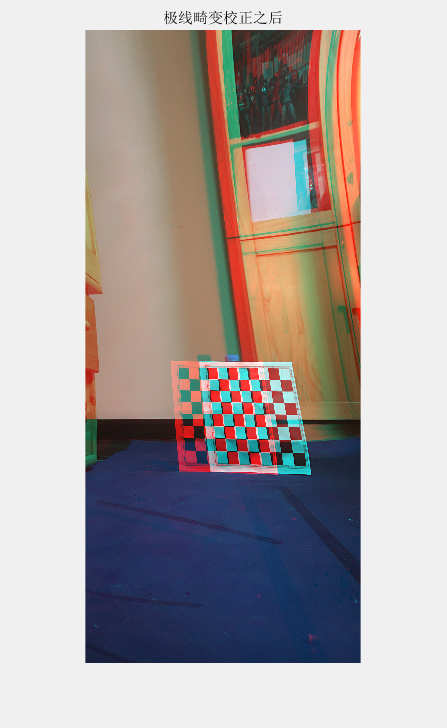


Fig.2-10 after rectification

As shown in Fig.2-10, now the corresponding lines in the two chessboards coincide with each other, indicating that the rectification is successful.

2.3 Block matching

Stereo matching is a key part of stereo vision research. The aim is to match the corresponding pixels in two or more views and calculate the parallax between two images. By establishing an energy cost function and minimizing it, the disparity of pixels is estimated and the depth is obtained. Calling Disparity function in matlab is a feasible way to do so, it allows user to generate disparity map. The function takes block matching as the principle and finds similar blocks by calculating the distance between pixel blocks in two images. The input of the function is the images J1 and J2 that are converted to grayscale images and the output is Disparity map. The Disparity map allows us to view the depth of our target. Also, thre are are few parameters we can adjust in the function, such as ‘BlockSize’, ’ContrastThreshold’ and ‘UniquenessThreshold’, all of which would influences the output.

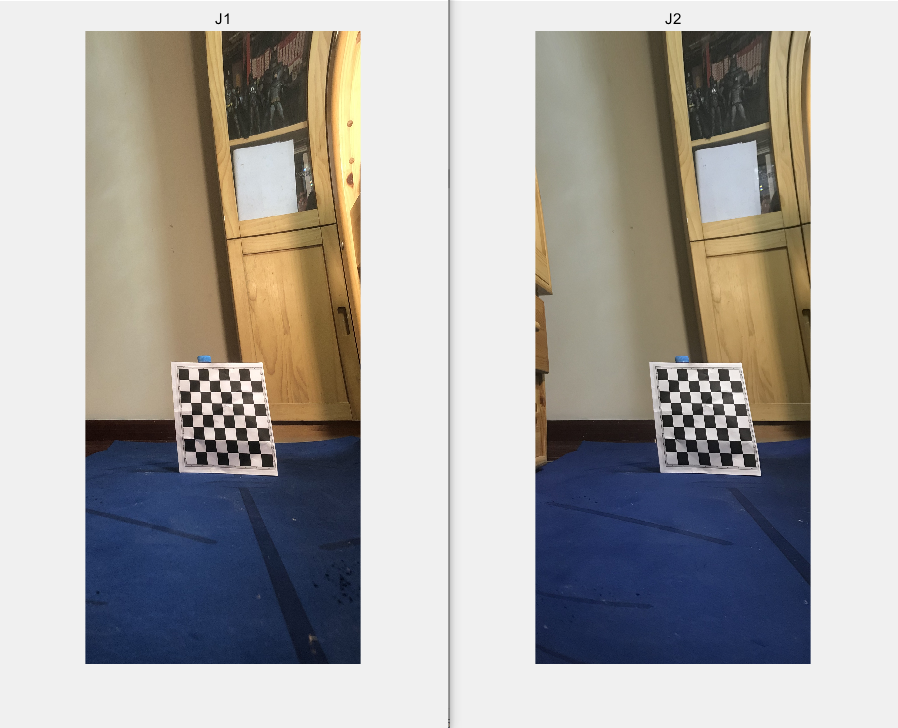


Fig.2-11 J1 and J2

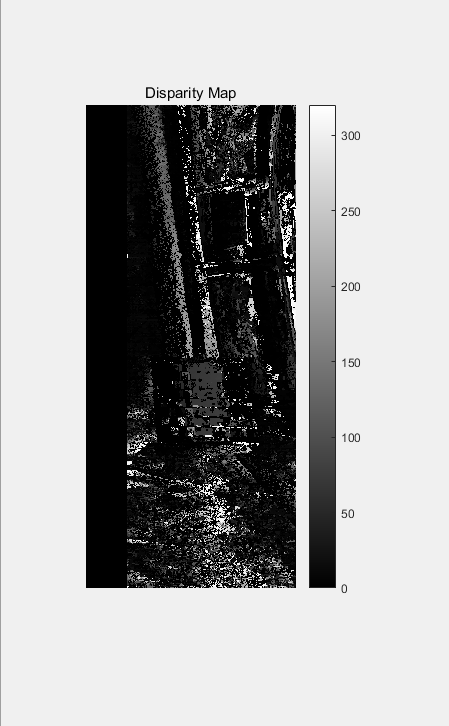


Fig.2-12 Disparity map

2.4 3D construction

After acquiring the disparity map, we can tell the occlusion relations in our target, which allows us to build 3D image. In this step, the reconstructionScene function in matlab helps build the image.

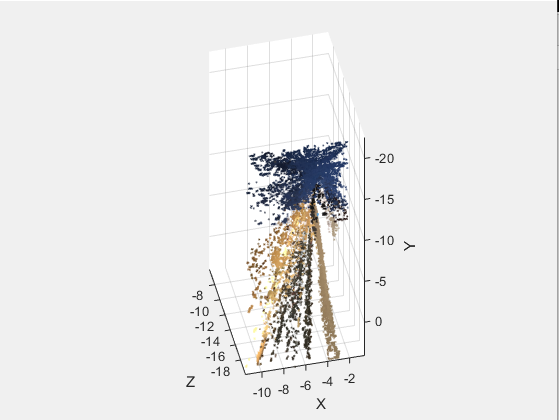


Fig.2-13 3D construction