COMP3065 Computer Vision
Coursework: Stereo Vision

Submitted May 6, 2022

Xinran Tang
20126518
shyxt1@nottingham.edu.cn

School of Computer Science University of Nottingham Ningbo China

Contents

Chapter 1

Introduction

This chapter introduces the background information and objectives of this coursework.

1.1 Background Information

The images used in the coursework were captured through Sony D7500 with a tripod. The original pixel size of the images is 2784 × 1856, which will take a long time for calculating. Therefore, the size of the input images is reduced to 500 × 333 pixels. By panning the camera left and right, three sets of pictures with left and right views are obtained as follows:

Figure 1.1: checkerboard grid images

1.2 Objective

The objective of this coursework is to develop an application that can allow users to choose the input image. Then, calculate the depth map of the chosen image and show the real distances from the image position to the real-world position.

The key features of this coursework are:

- Design a user interface to allow the user to choose the input image.
- Implement camera calibration to obtain the intrinsic matrix, distortion factor, and exterior parameters of the camera.
- Implement the Block Matching Algorithm to obtain the depth map of input images.
- Calculate the real distance from the image points to the left camera.

1.3 Code Structure

- **calibration.py**: Work for camera calibration. Use the 15 calibration images to obtain the camera parameters: intrinsic matrix, distortion factor, rotation vector, and translation vector.
- **camera_config.py**: Store the configurations of the camera which will be used in the image rectification and real distance calculation.
- **depth.py**: Generating depth map and calculating the distance in the real world.
- **main.py**: Start of the program. By running this file to start the program.

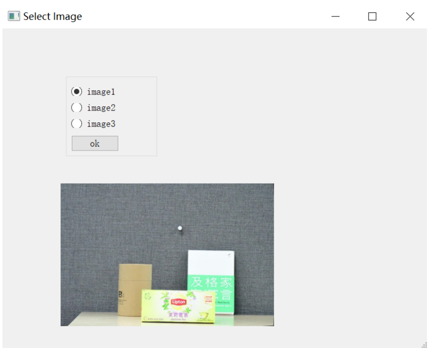
Chapter 2

Key Features and Results

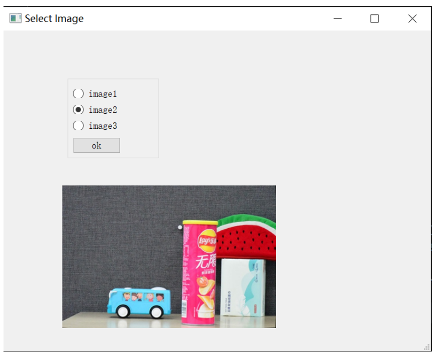
This chapter introduces the key features implemented and the corresponding results.

2.1 User Interface

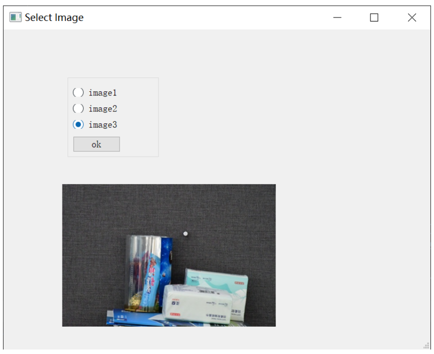
For the user interface, PyQt5 is used to design the interface. Users can choose the input image by clicking the buttons shown in Figure 2.1, different images will be shown on the interface if clicking different buttons.



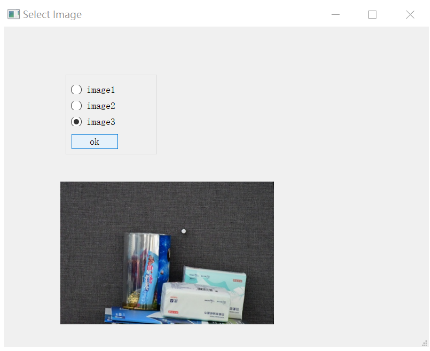
(a) Select Image 1



(b) Select Image 2



(c) Select Image 3



(d) Confirm Selection

Figure 2.1: User Interface

Figure 2.1(a) shows select the first image, Figure 2.1(b) shows select the second image, and Figure 2.1(c) shows select the third image. Finally, Figure 2.1(d) shows confirm the selected image by clicking the "ok" button. After confirming the selection, the selected image will be input to the program which can calculate the depth map of the image by the Block Matching (BM) algorithm and calculates the depth value in the real world by using the parameters obtained from the camera calibration process.

If choosing the second image, after confirming the selection. Wait a few minutes, the depth map will be shown in another window. By clicking the positions in the image, the distance from the position to the left camera in the real world will be shown on the terminal. As shown in Figure 2.7

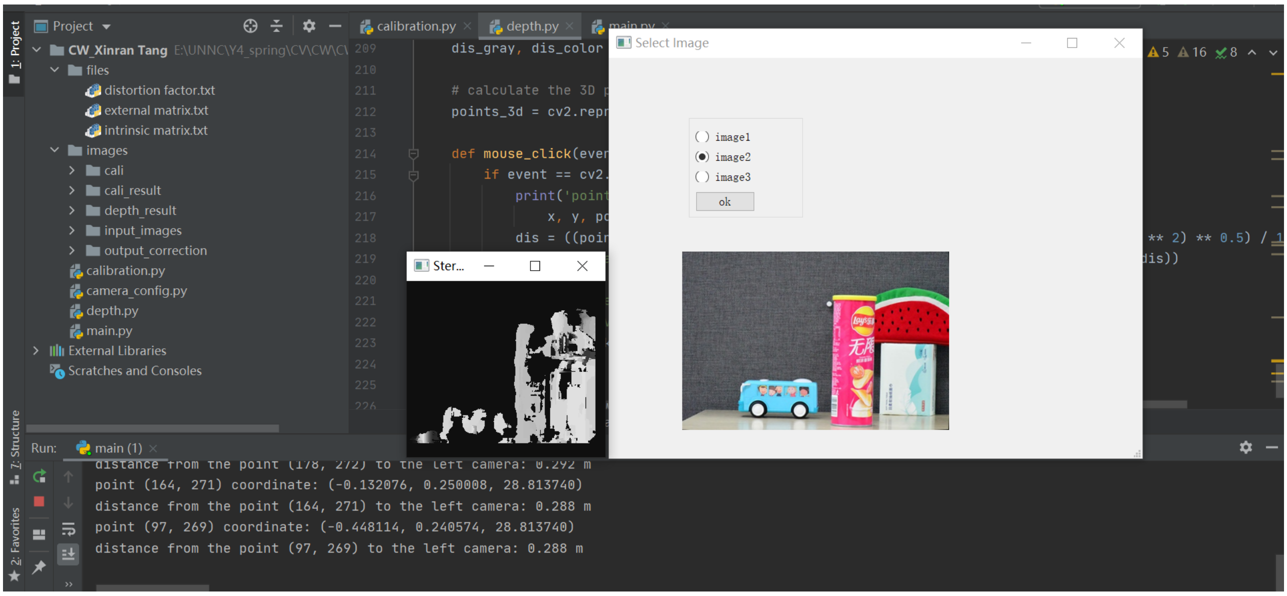


Figure 2.2: Depth Map Window and Depth Value in Real World

2.2 Camera Calibration

Camera calibration aims to rectify the input images. For camera calibration, Zhang's calibration method [1] is implemented. This method is between the photographic calibration method and the self-calibration method, which overcomes the shortcomings of the high-precision three-dimensional calibrations required by the photographic calibration method and solves the problem of poor robustness of the self-calibration method. The calibration process only requires the use of a printed checkerboard grid and several sets of pictures taken from different directions. Anyone can make their calibration pattern, which is practical, flexible, and convenient. Figure 2.3 shows the input checkerboard grid images:

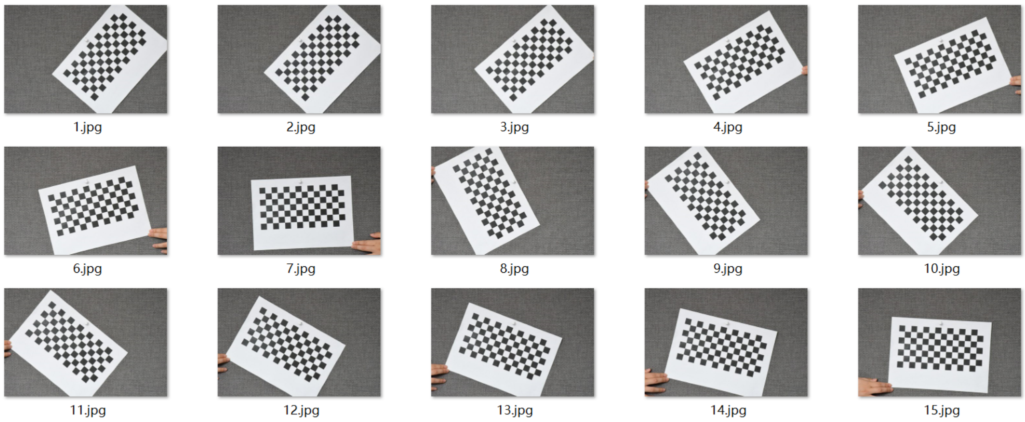


Figure 2.3: Checkerboard Grid Images

The results of the camera calibration are shown in Figure 2.4

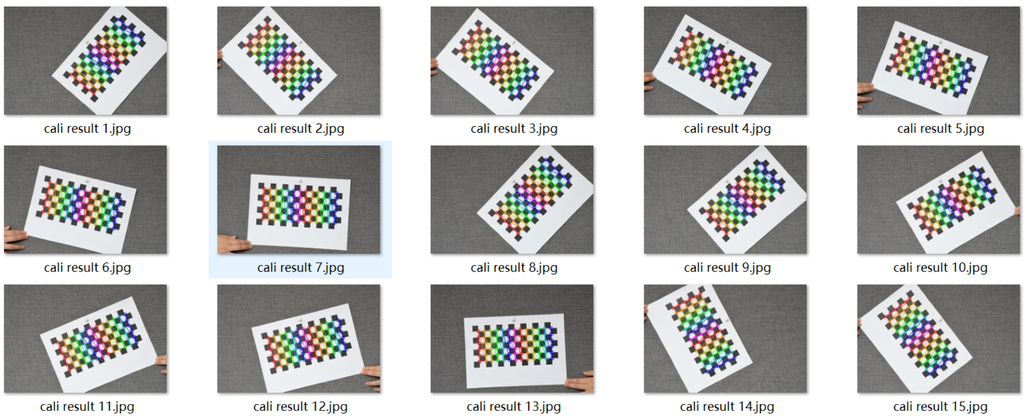


Figure 2.4: Calibration Result

Then, use the intrinsic matrix, distortion factor, rotation vector, and translation vector obtained in the calibration process to rectify the input image. For example, Figure 2.5 is the image captured by the camera, and Figure 2.6 shows the image after correction. The image after rectification will be used in the following steps.

Figure 2.5: Input Image

Figure 2.6: Output Image

The internal matrix and the distortion factor of the camera obtained from the calibration process are shown below:

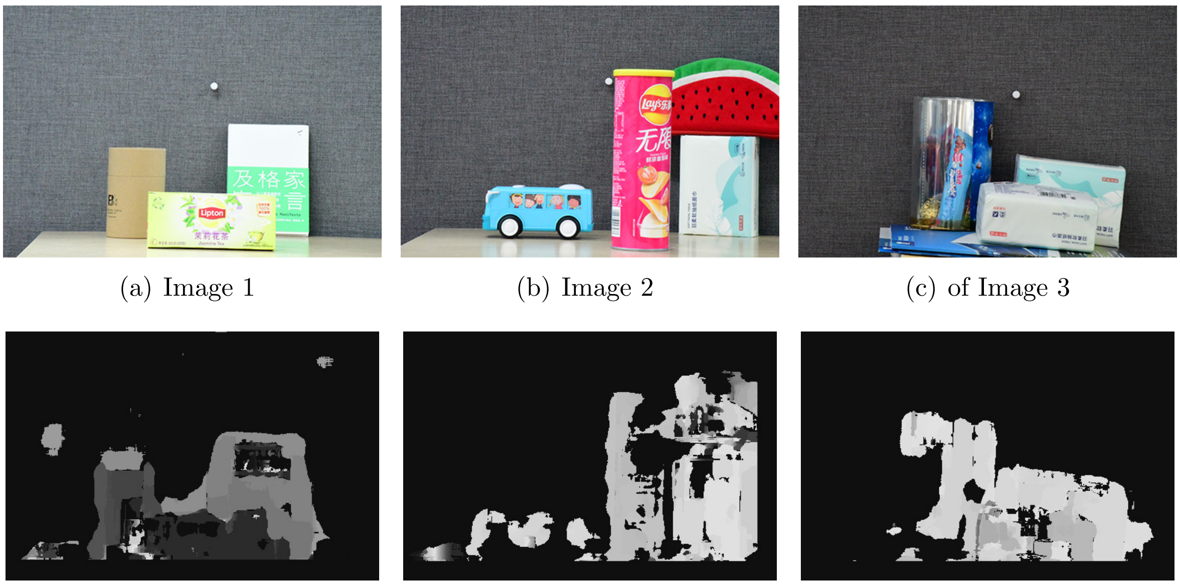
$$internal \quad matrix = \begin{bmatrix} 2.41e+03 & 0.00e+00 & 2.43e+02 \\ 0.00e+00 & 2.41e+03 & 1.60e+02 \\ 0.00e+00 & 0.00e+00 & 1.00e+00 \end{bmatrix} \tag{2.1}$$

$$distortion \quad factor = \begin{bmatrix} 1.42e+00 & 3.20e+01 & -8.17e-03 & 6.67e-04 & 7.81e+02 \end{bmatrix} \tag{2.2}$$

2.3 Block Matching Algorithm

BM algorithm will divide the current frame into many small pieces, and compare each small piece with the small piece collected by another camera. This process is obtained by moving the small piece. The process of moving is to create a vector that simulates the motion of the block as it moves from one location to another.

In Sum of Absolute Differences (SAD) parallax calculation, the original left and right images are pre-processed by histogram equilibrium and Sobel filter to reduce the impact of noise and light. Then, calculate the disparity of left and right images respectively. After removing the low texture visual handicap according to texture, produce the disparity image of the left and right image. The original images and the corresponding depth map of three sets of images are shown in Figure 2.7.



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

(d) Depth Map of Image 1 (e) Depth Map of Image 2 (f) Depth Map of Image 3

Figure 2.7: Depth Map

2.4 Depth Information in Real World

By converting the depth map derived from the BM algorithm to real-world distances, it can be used in a variety of projects, such as robotic vision, aerial mapping, military and medical imaging, and industrial inspection. To make this coursework more practical, the pixel positions of different squares in another image are found and combining the relative data of the two cameras (translate and rotation matrix in calibrated parameters) to calculate the actual depth of the object. By clicking the points in the shown depth map, the distances in the real world are shown like in Figure 2.8

```
point (362, 266) coordinate: (0.944444, 0.266676, 33.936180)
distance from the point (362, 266) to the left camera: 0.340 m
```

Figure 2.8: Depth in Real World

Chapter 3

Discussion

This chapter discusses the strength and weakness of this coursework.

3.1 Strengths

Firstly, one of the strengths of this coursework is that the user can choose the input image by clicking the button on the user interface. The program will input different sets of left and right images according to the user's choice.

Secondly, in order to achieve a fast calculation, the BM algorithm is implemented. The BM algorithm can produce a depth map in 5 minutes.

Finally, the real distance from the positions in the image to the left camera can be obtained. It can be used in some applications such as mobile robot autonomous navigation system, aviation and remote sensing measurement, industrial automation system by including the distance value in the real world.

3.2 Weakness

The weakness of this coursework is the accuracy of the BM algorithm is not good enough. Compared to Semi-Global Block Matching (SGBM) and Graph cuts (GC) algorithm, though the processing speed BM >SGBM >GC, the accuracy BM <SGBM <GC. And the BM algorithm can only calculate the depth map for 8-bit gray image. Therefore, in this coursework, some accuracy was sacrificed for speed.

Another weakness of this coursework is that the camera used is not the binocular camera, therefore, different sets of images captured have different number of disparity.

References

[1] Zhengyou Zhang. A flexible new technique for camera calibration. *IEEE Transactions on pattern analysis and machine intelligence*, 22(11):1330–1334, 2000.