NANYANG TECHNOLOGICAL UNIVERSITY

Depth calculation from an image pair for pedestrian safety system

JIN Linhao

Jin_linhao@hotmail.com

https://github.com/Jin-Linhao

2016

Introduction

The project is mainly about finding the depth information from stereo images. The depth of an object in the image can generally be found in following steps:

1. Calibrating stereo camera
2. Finding disparity map
3. Calculating depth from disparity map

The basic ideas come from the theory of computer vision and they can be found in my summary in Github. For further questions regarding the project, you can find me via my email (jin_linhao@hotmail.com).

Current Progress

So far step 1 and step 2 are done. The disparity can be found from a pair of rectified images. The codes are generally adapted from OpenCV samples and tested with real images in Xcode platform. Codes and other project files can be found in my Github StereoMatching repository where I have uploaded the entire workspace. I have rearrange the whole workspace before uploading so the file path might also be changed.

In Disp_Map.cpp, there are two major functions: calculating disparity map from an image pair and creating a tracking bar with changeable image processing parameters. The output and best parameters can be found in Figure 1.

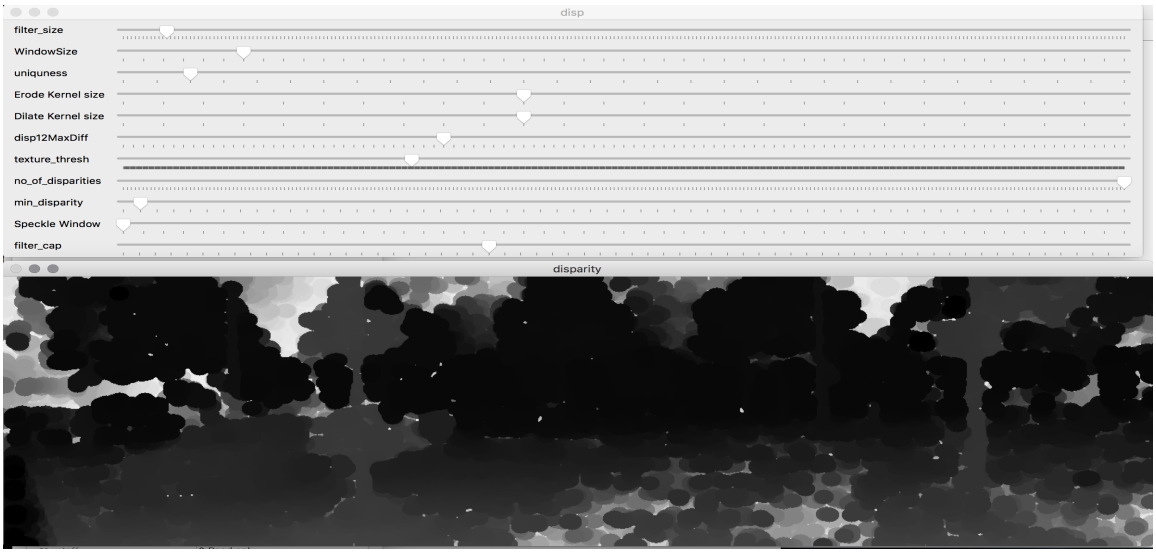

Figure 1.

or in Figure 2.

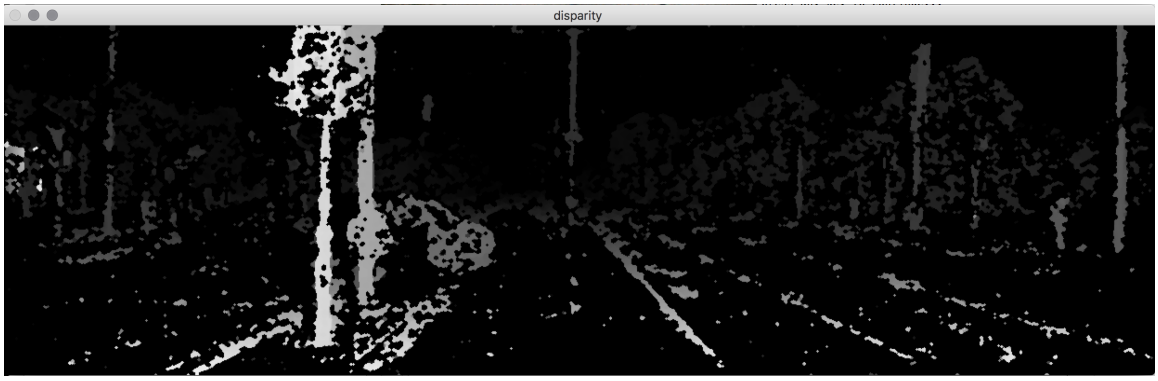

Figure 2.

Image processing techniques (dilation and erosion) are used here to are used to remove the noise and smooth the image border.

Stereo_Calib.cpp reads a pair of stereo chessboard images, calibrated the camera and write camera parameters in a xml file. Several batches of images from different devices are tested. iPhone are not recommended and a set of professional stereo camera is implemented. The outputs are shown in Figure 3.

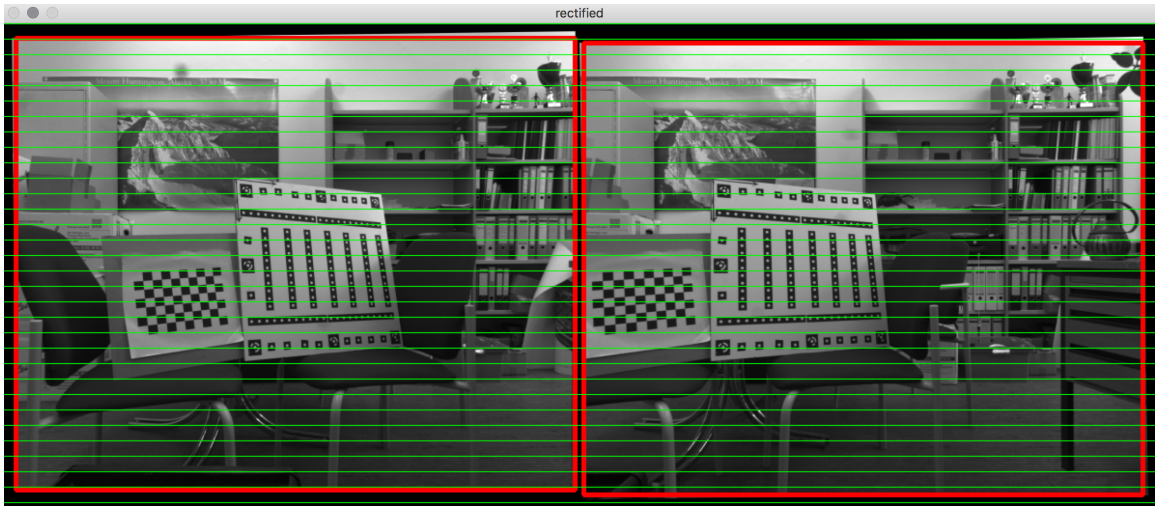

Figure 3.

Further suggestions

1. The preliminary challenge is the connection between camera calibration and finding disparity map. So far these codes are written in separate c++ files with individual main functions. The challenge is to revise the file and link it with other files of the project.

2. The disparity maps calculated from the set of images taken by stereo camera are still not desirable. In Figure 4, the contour of the objects is not clear and seems that there is a relative displacement between each horizontal of the image. One probably reason is the testing error. Here a pair of rectified images is saved as .png format, and used to find disparity. All the black pixel caused by rectification at the bolder of the image are assigned to be zero. This will significantly affect the disparity since block matching and semi global block matching algorithm analyze the same horizontal line in different images.

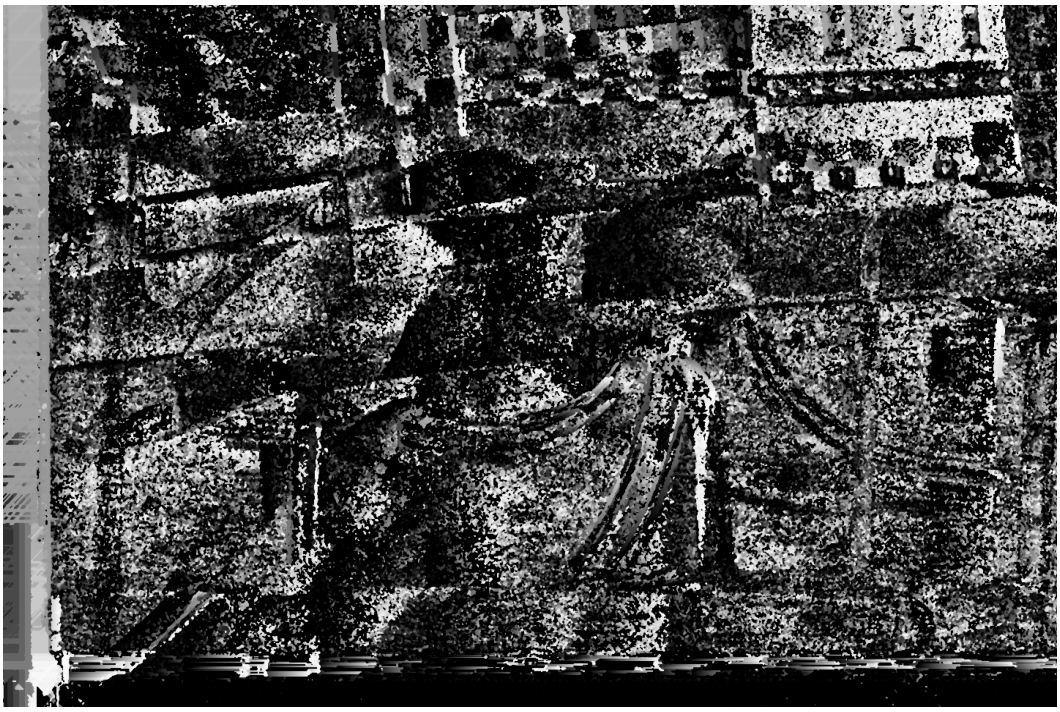

Figure 4.

In order to solve this, one way is to revise the code and directly take the two rectified images to find the disparity. If this doesn't work, the rectified images may need to be chopped to guarantee alignment of the horizontal lines. Different environment settings are also need to be tested and the algorithm may need to be changed in order to get an optimum result.

3. When everything is done, the next step is to find the average pixel value of a region of interest from disparity. These can be easily done by OpenCV in either C++ or Python.
4. I would also like to suggest to use ROS framework for the whole project. ROS will provide different nodes for stereo calibration and disparity calculation and also other functions for the project. It will be easier to manage the entire project in ROS framework.