Depth Estimation from Stereo Image

NYCU Computer Vision 110 Spring

Github link:

https://github.com/clashroyaleisgood/Course_ComputerVision/tree/main/Final_Project

309553018 林孟學

0716038 郭崇瑋

I. Introduction

It's hard to find the 3D position of an object with only one image. When the camera captures the image, it only preserves the 2D relation of the real world. But with two cameras, predicting depth becomes possible with triangulation, finding correspondence, and a lot of mathematical calculations.

Our goal is to implement this idea, from mathematical calculations to real things.

- find disparity from 2 well-rectified images, provided in middlebury
- image rectification from unrectified images, collected by ourselves
- experiment with different filter

Our goal is to implement this idea, from mathematical calculations to real things.

And we finally implement the disparity algorithm: Block matching, DP method, along with 2 post-processing methods and image rectification algorithm(finding pair part)on our own.

II. Method

1. Disparity from rectified images - **Block Matching**
   (1) for each pixel in the left image, bounding a window and scanning all the pixels in the right image.
   (2) use the SSD method to calculate the minimum for all windows scanned in the right image. find the pixel which has the minimum corresponding to the pixel in the left image.

(3) save the coordinate distance of two pixels in the left and right image respectively to disparity map.

2. Disparity map Post-processing - **Filtering**

   (1) smooth the border of different object

3. Disparity from rectified images - **DP method**

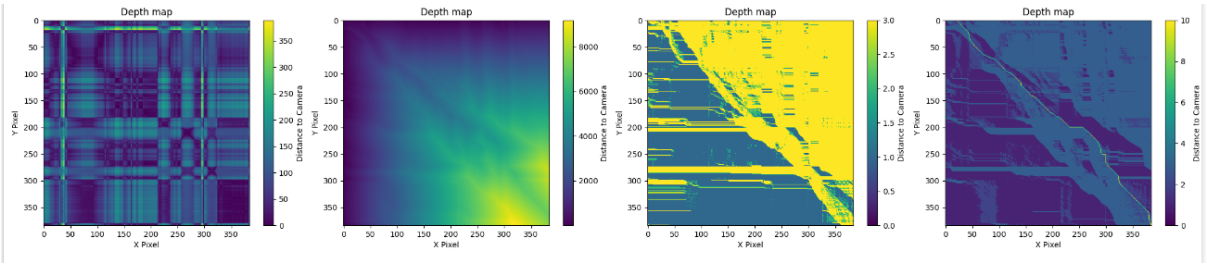
   for each row:

   (1) calculate **correlation map** for rows in left and right image

   (2) calculate **DP and directional map** to find minimum cost path from start to end(the matching method to get lowest cost)

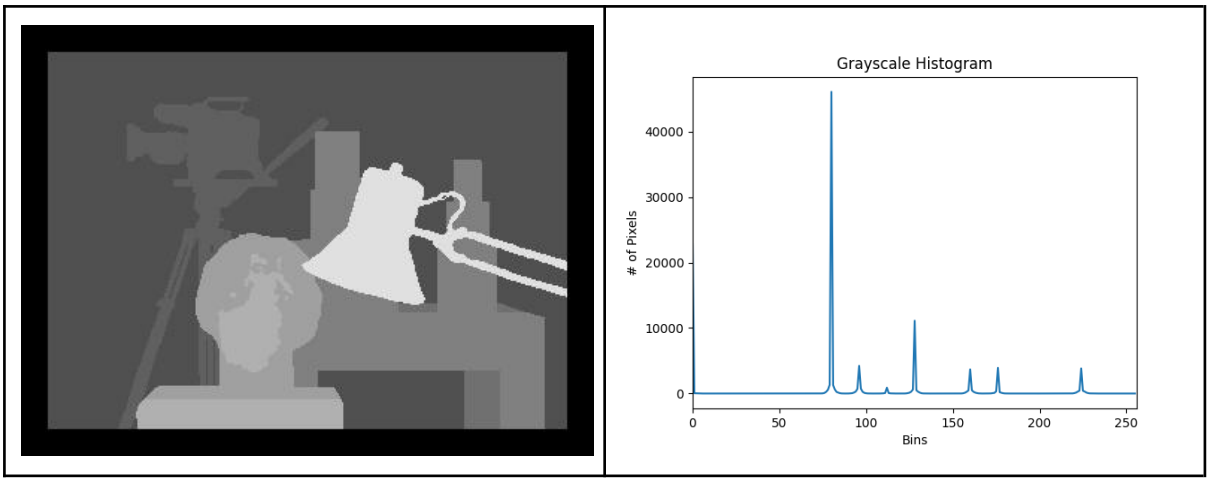
   (3) calculate disparity line according to the path

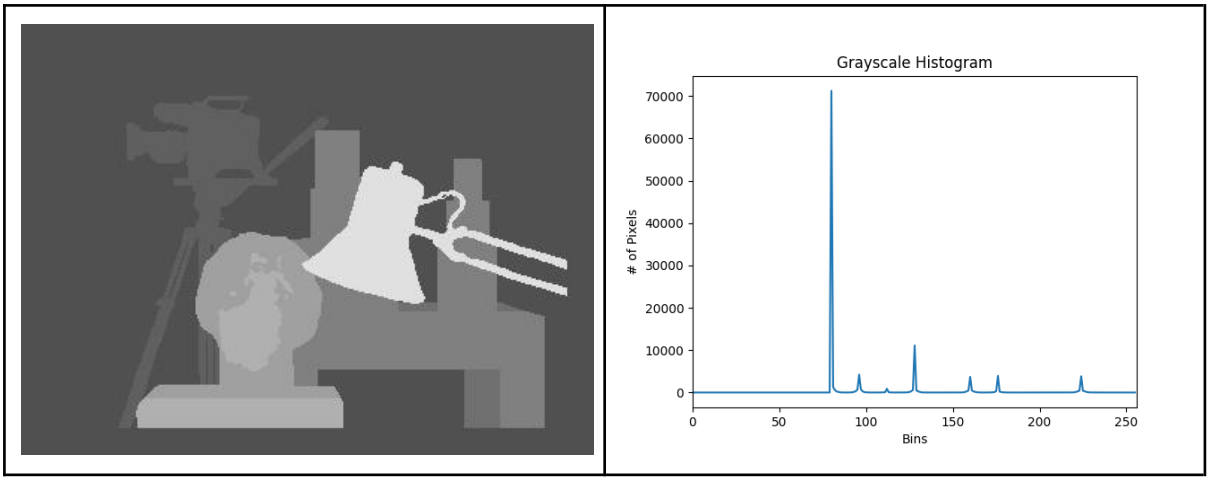
   (4) calculate the line disparity from the path



4. Disparity map Post-processing - **Histogram**

   (1) use Histogram to threshold the lower peak value(often with value 0) which is caused from disparity algorithm error, or the border of middlebury disparity map





two small peaks, one at 0, one at around 80. my algorithm, `getDisparityMin()`, will find the second small peak(80), treat it as a real background disparity number.

Then I can threshold it with `disp[disp < disp_min] = disp_min`

5. Image **Rectification**
   (1) find corresponding **point pairs** and their positions
   (by SIFT feature and kNN algorithm with Low's ratio test)
   (2) calculate the **fundamental** matrix by point pairs
   (3) calculate the H1 and H2, which mean the **homography** matrices to wrap two images, by fundamental matrix and point pairs
   (4) use homography matrices to **wrap** images, so that tow

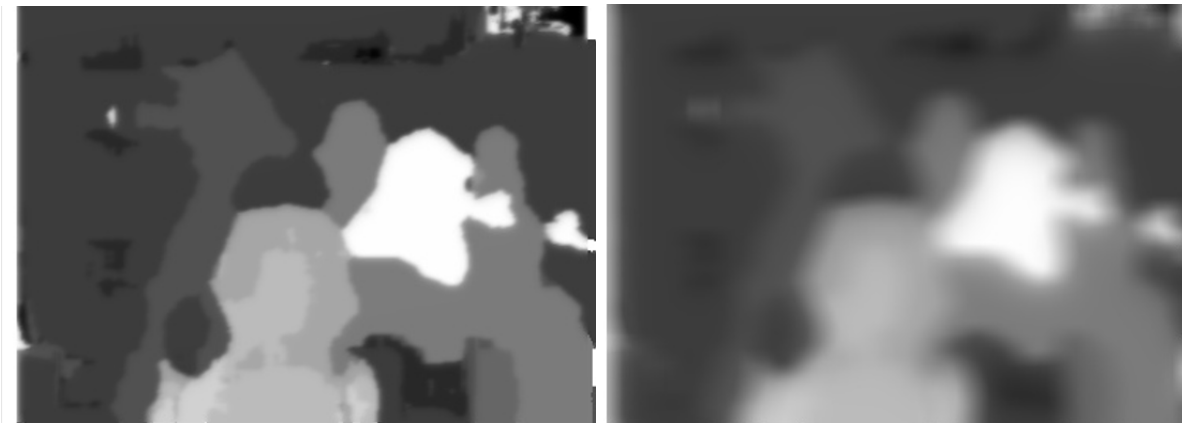
III. Results and Experiments

1. Disparity from rectified images - **Block Matching**
   (1) result

2. Disparity map Post-processing - **Filtering**

   (1) result

gaussian filtering | mean filtering

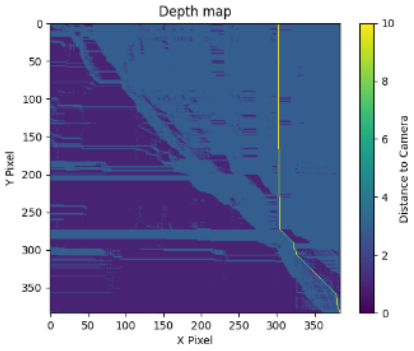


3. Disparity from rectified images - **DP method**

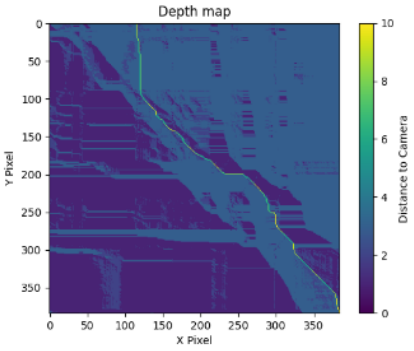
   (1) result

Disparity(norm to [0, 255]) | Depth map(norm to [0, 255])

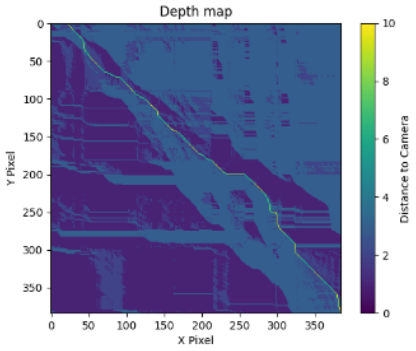
   (2) experiment with different occlusionConstant(path from in DP map)



occ: 10

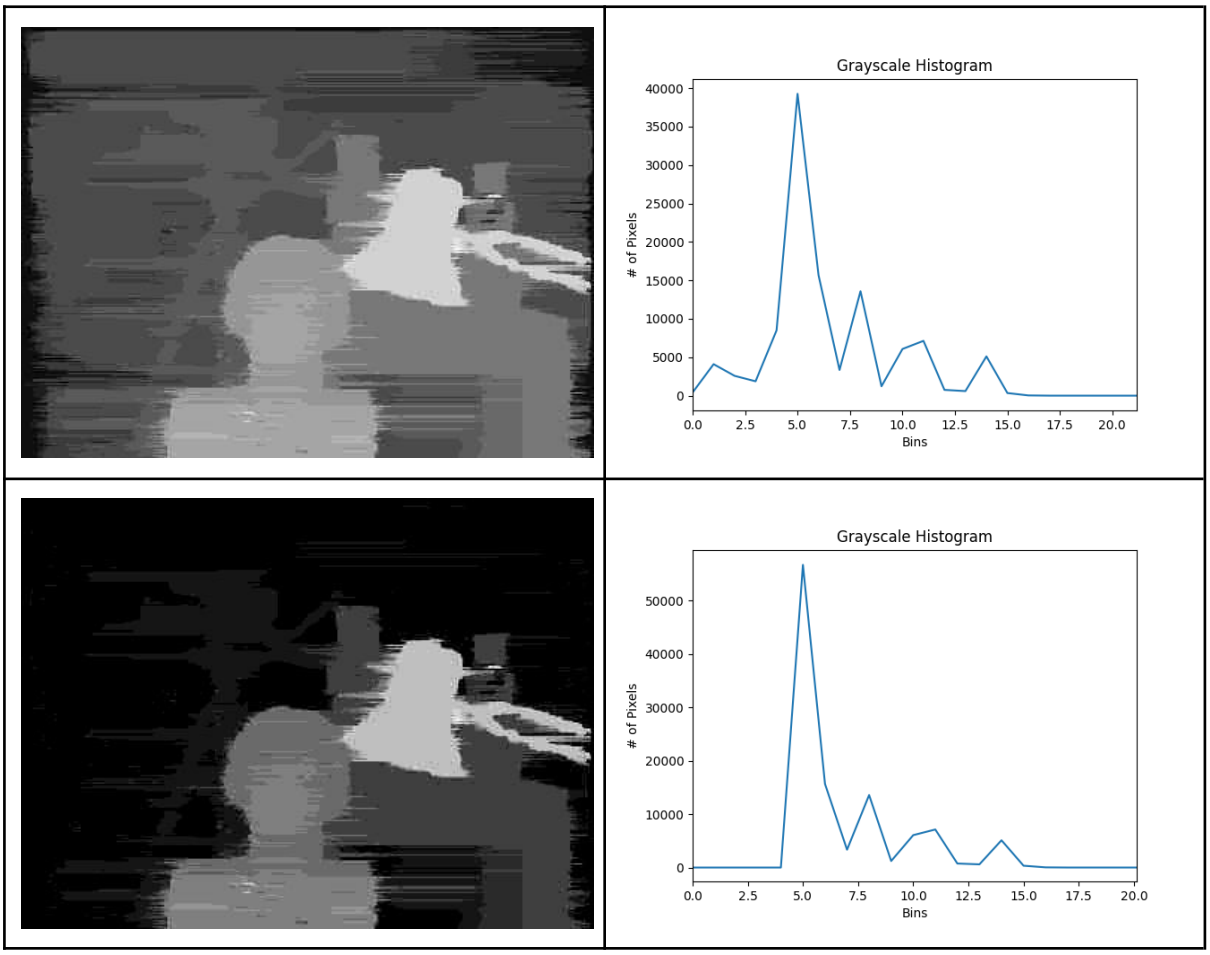


occ: 20



occ: 30

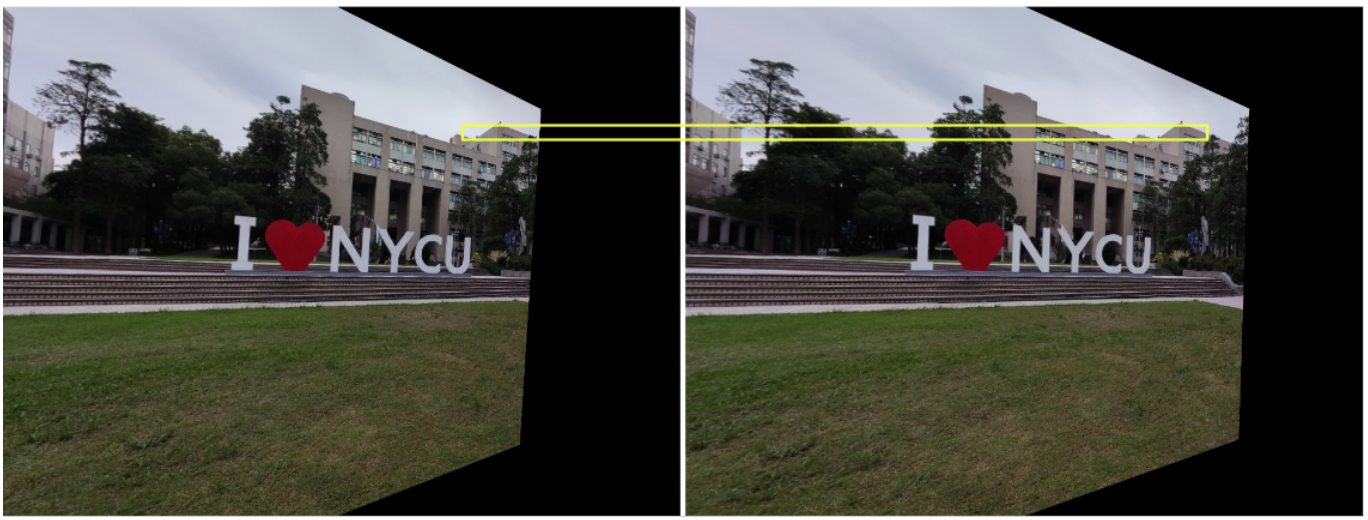
4. Disparity map Post-processing - **Histogram**



5. Image **Rectification**

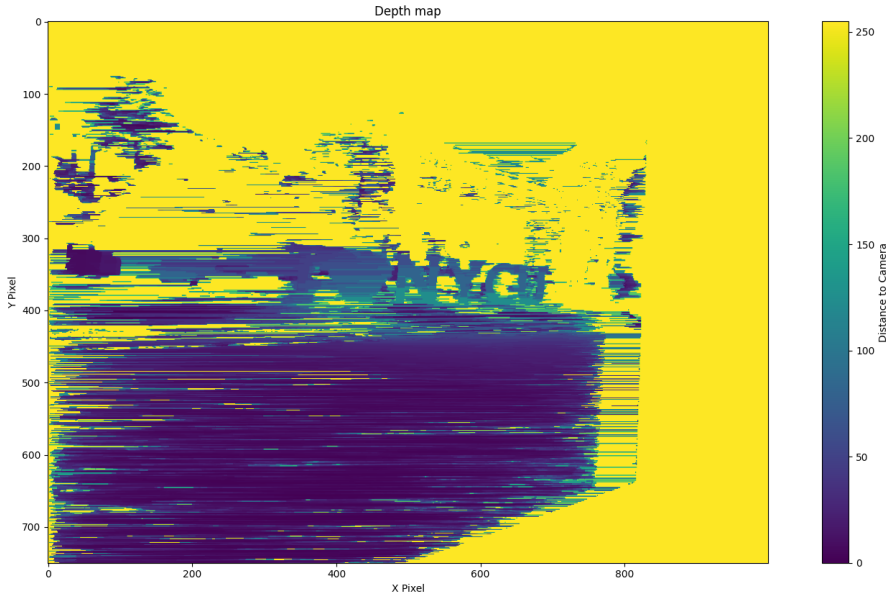
unrectified

rectified: the top of 工五 is well-aligned



6. calculate **disparity** from the **rectified images**

   On the presentation day, it seemed to be horrible to calculate disparity from our rectified images. And I think maybe the **resolution is the problem**: Because the disparity algorithm is really time-consuming, I resized our rectified image from (1000, 750) to (300, 225) which is similar to the size of images in middlebury:tsukuba (384, 288). But the (384, 288) sized image was taken in-door, the depth **isn't varying large**, and my (300, 225) sized image was taken out-door with a **large depth difference**, the pixel-shift may be **too small** if the depth is large. So I do it again with the original size(1000, 750), and get an acceptable result.



The "I ♡ NYCU" in the middle is separated from background buildings, and below it is the grass which is nearer to the camera.

IV. Contribution

- 林孟學: **DP method** in disparity.py, **Image rectification**, post-process: **histogram method** in rectification.py,
  self-collected data and data selector in datasets.py
- 郭崇瑋: **Block matching method** in disparity.py, **Filtering** in disparity.py

V. Reference

1. opencv: findFundamentalMat(), stereoRectifyUncalibrated(), SIFT_create(), warpPerspective()
2. Middlebury dataset: https://vision.middlebury.edu/stereo/data/scenes2001/
3. disparity DP method lecture: https://www.cse.psu.edu/~rtc12/CSE486/lecture09.pdf
4. image rectification: https://www.andreasjakl.com/understand-and-apply-stereo-rectification-for-depth-maps-part-2/