CS4186 Assignment 2 Report - Generating Stereo Disparity Map

Chan Tsz Fung (56228799)

Background

This assignment aims to produce three disparity maps given three pairs of stereo images. For every pair of images, the left view and right view of the object is provided for the reconstruction process.

Overview on Methodologies

In order to generate depth map, I adopted SGBM in OpenCV for calculation. Before the generation, preprocessing like extracting features points, calculating fundamental matrix, drawing epipolar lines and rectifying images are performed. After that, I used the rectified images for Semi-Global Block Matching. The parameters are tuned manually one by one with reference to the PSNR value given by comparing the generated map to the ground truth map. Finally, I also take an investigation on postprocessing the generated depth map for retrieving a better PSNR score. I have performed inpainting techniques and Gaussian filtering for reducing noise on the map. The detailed explanation for all the steps and experiments will be illustrated below.

Preprocessing for image rectification

I adopted SIFT for extracting all the features points on both images. Flann-based matcher is used to find out all the matches. After that, good points are selected from all the points using Lowe's ratio. The good points are then used for calculating the fundamental matrix, which represents the relationship between the two stereo images. Through using the mask obtained from calculating fundamental matrix, I filter the points from all the good points I have. Then I use those points for drawing the epipolar lines (Figure 1). Finally, the rectified images are obtained through warping (Figure 2).

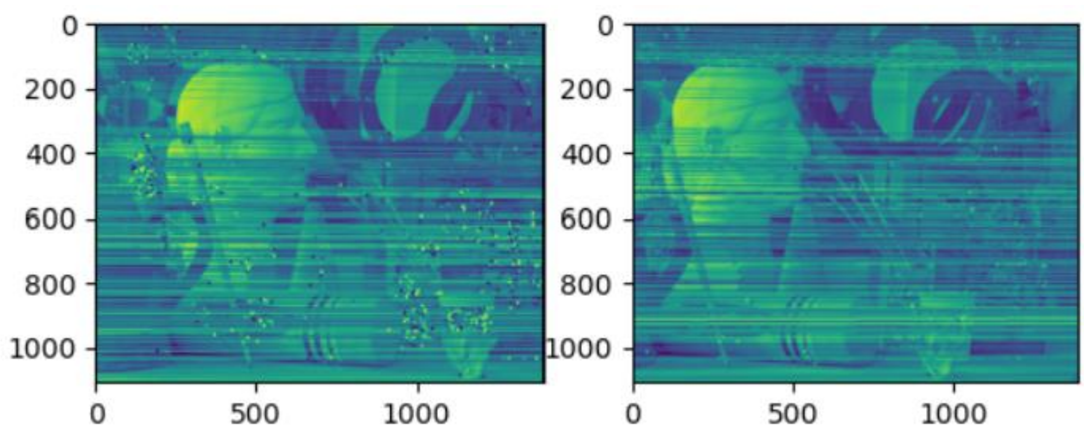

Figure 1 Drawing Epippolar Lines

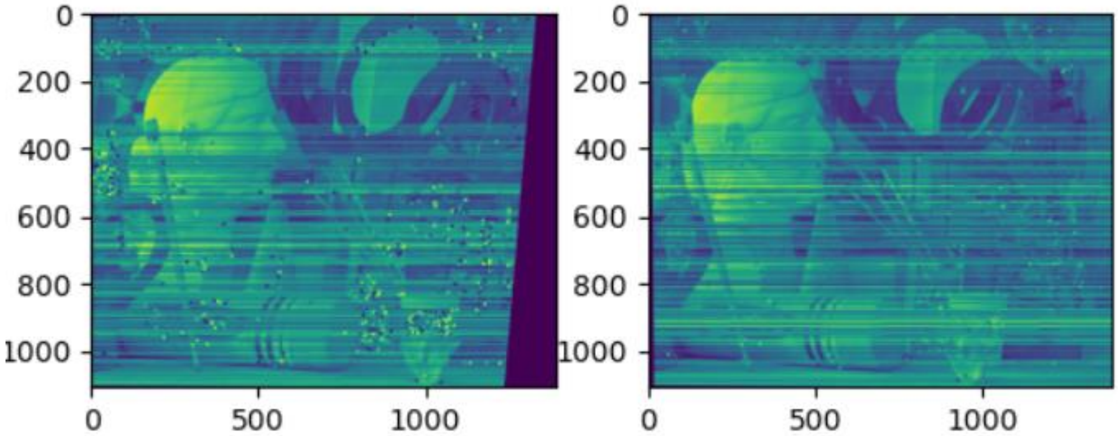

Figure 2 Rectified Image

<u>Stereo SGBM for disparity map generation</u>

Stereo SGBM is an algorithm I used to generate dense disparity map from two rectified images. It calculates a matching cost for every pixel on the left image and right image. The disparity value is given according to the minimum cost found. The algorithm searches all the 8 directions surrounding a pixel and aggregates the total cost for generating the dense disparity map. This is known as the “semi-global” approach.

<u>Result for the generated disparity map</u>

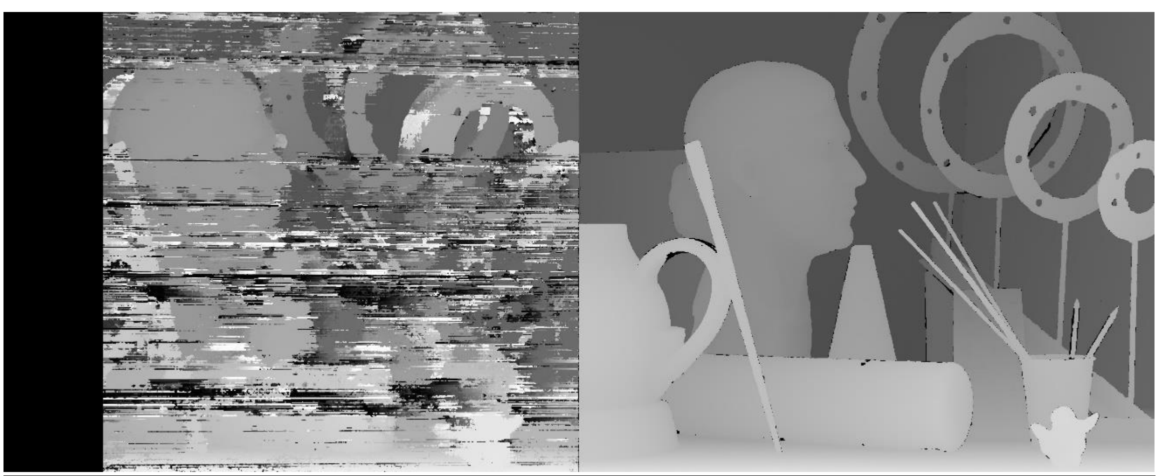

Figure 3 Disparity map generated using SGBM (left) , Ground truth (right)

It is shown that the generated disparity map is very similar to the ground truth (Figure 3). The PSNR score for artwork, doll and reindeer images are **9.978, 9.719 and 10.264** respectively. The average score is about **9.987.**

However, it is discoverable that the generated disparity map contains a **black empty region** on the left side. The reason for this is that some points on an image cannot be located on another image. As a result, I adopt inpainting and filtering techniques for giving a better result. This would be covered in the next section.

The second issue of the generated disparity map is that it is quite **noisy**. The reason for this is that I chose a very small window size (w=1) for implementing the SGBM algorithm. With smaller window size, the generated map can have more details, hence improving the PSNR score. The noise would be an inevitable tradeoff for this advantage. Nevertheless, I still experiment with some post-processing techniques for improving the result.

Post-processing- Inpainting and Filtering

To achieve a high PSNR score, I have applied inpainting and filtering to improve the result of the generated disparity map. I used Navier-stokes inpainting method in OpenCV, which is based on fluid dynamics and partial differential equations for estimating missing points in image. For every pixel that is under the threshold disparity value, they are converted to zero. To obtain the optimum threshold, I have experimented with different values and found out that the performance improves most when the threshold is 30. It means that I would in-paint all the pixels with disparity values less than 30. The result are as follows:

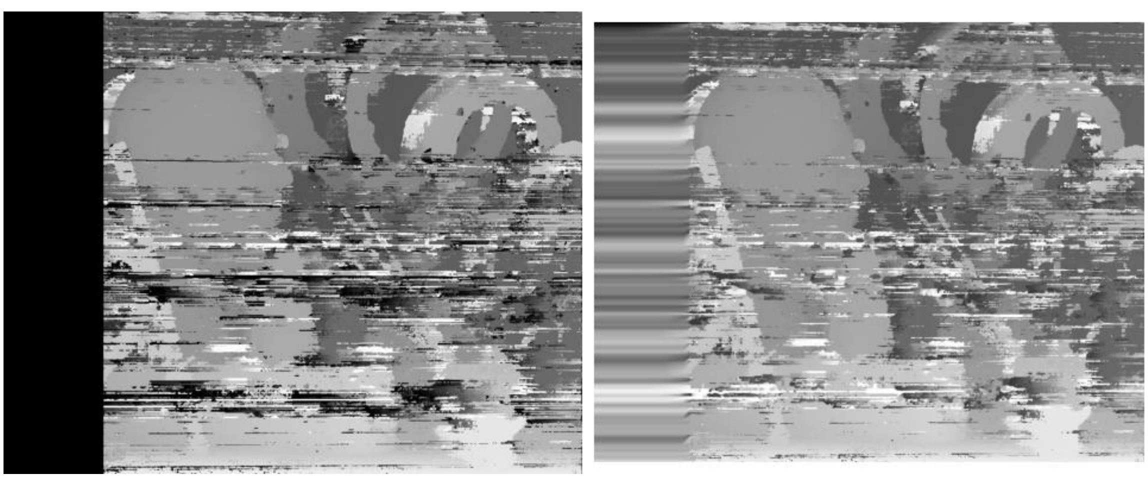

Figure 4 Before inpainting (Left), After inpainting (Right)

It is shown that the empty region is removed and some of the noisy pixels are being painted in white. When I evaluated the PSNR score again, it was found that the PSNR score of the three images are **14.52,12.97 and 13.25** respectively. On average, the PSNR score is **13.58**, which is about **3.593 higher** compared to the original disparity map.

After inpainting, I tried to figure out ways to further improve the PSNR score of our disparity map. One of the ways I adopt is image filtering using Gaussian filter. The gaussian filter is a common approach in smoothing the image and reducing the noise on image. I have tried out using different sizes of kernels. It is found out that **the larger the kernel size, the more noise I can reduce and the higher PSNR score I can achieve.** However, the tradeoff is that the disparity map would be much blurred if I make the window size too large. The image content in the original disparity map would be lost and the map would become meaningless. Therefore, I decided the window size to be 21x21, in order to achieve a good PSNR score, while retaining the details in the images. The resulting PSNR score for the three images after filtering are **16.07, 15.53 and 14.93** respectively. The average score is **15.51**, which is about **1.93** higher compared to those of the in-painted version. The resulted disparity map after inpainting and filtering are shown below:

Figure 5 Before Filtering (Left), After Filtering (Right)

<u>Conclusion</u>

The final PSNR score for the three images are 16.07, 15.53, and 14.93. The score achieved at each stage are summarized as below:

Image pair	PSNR for original disparity map	PSNR for in-painted disparity map	PSNR for in-painted disparity map with Gaussian filtering
Art	**9.978**	**14.52**	**16.07**
Dolls	**9.719**	**12.97**	**15.53**
Reindeer	**10.264**	**13.25**	**14.93**
Average	**9.987**	**13.58**	**15.51**

References

[1] H. Hirschmuller, “Accurate and efficient stereo processing by semi-global matching and Mutual Information,” *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR’05)*. doi:10.1109/cvpr.2005.56