Experiments On Different Stereo Matching Algorithms

Jiahao Chen
The University of Melbourne
Melbourne, Australia
jiahchen4@student.unimelb.edu.au

Mingyang Liu
The University of Melbourne
Melbourne, Australia
minliu2@student.unimelb.edu.au

***Abstract*—In this article, we propose a method of stereo matching which combines results from 3 disparity maps, including a normal one, an up-sampled one and a map with larger window size. A couple of experiments on basic algorithms are performed first to demonstrate our evolution. Our algorithm finally achieves an average root mean squared (rms) error of 27.53 and similarity of 0.89.**

***Keywords*—Stereo matching, disparity map**

I. Introduction

Stereo matching is a classical method of finding matching pixels from stereo image pairs which are two images taken from two cameras at the same time. Cameras are required to be parallel and separated along the x-axis and their configurations are known, which can provide a useful epipolar geometry constraint for matching correspondences [1]. It is widely used in the area of computer vision, such as autonomous vehicles and robotics. It faces a significant challenge that pixels only indicate spatial and color information and obtaining accurate depth information with reasonable speed is difficult [2].

The key element in stereo matching is disparity, which is the difference of corresponding pixels' x axis coordinate. It is inversely proportional to the pixel's depth (i.e., distant objects in the camera's scene have small disparities). The main purpose of this article is to put forward a method to calculate the disparity of each pixel given a pair of stereo images. Results are stored as a new image naming disparity map with each pixel's value represents the corresponding pixel's disparity in the left image.

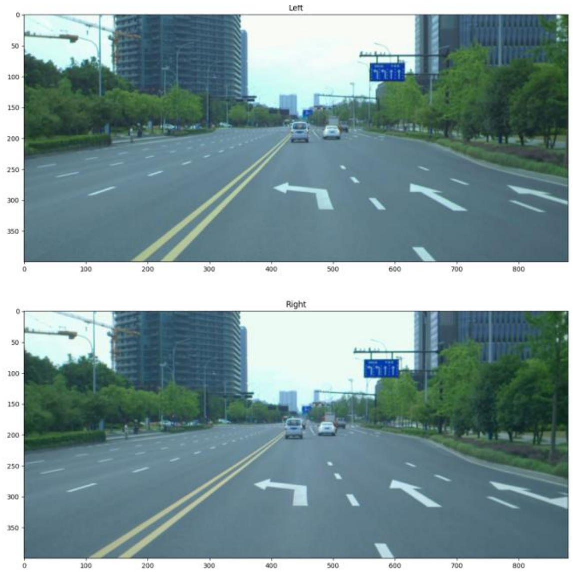

Fig. 1. Example of a pair of stereo images

The dataset used in this project contains 25 pairs of stereo images taken from moving vehicles, representing different kinds of possible driving scenarios [3] as figure 1 shows. In addition, Ground truth files of their disparity maps are provided as shown in figure 2, which are generated through LIDAR and help evaluating different methods. The ground truth file assigns each pixel with its true disparity value and missing ones are set as zeros.

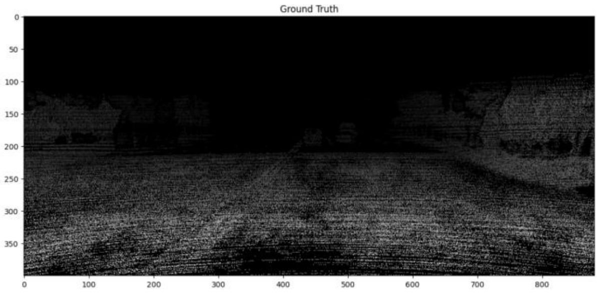

Fig. 2. Example of a ground truth file

The remainder of this article is organized as follows. The algorithms implemented and choices of design are explained in Section II. Then in Section III, experimentation steps and results are presented. Evaluation on these results and error analysis are performed in Section IV. Finally, concluding remarks are included in Section V.

II. Methods & Design Choices

To begin with, we first attempt to loop over all pixels in the left image to find their corresponding in the right image. Sum of squared differences (SSD) is applied as the similarity function. However, the speed is extremely slow that it takes about 10 minutes to generate a disparity map. Therefore, we come up with a better one as depicted in figure 3, which is the basic of our final algorithm.

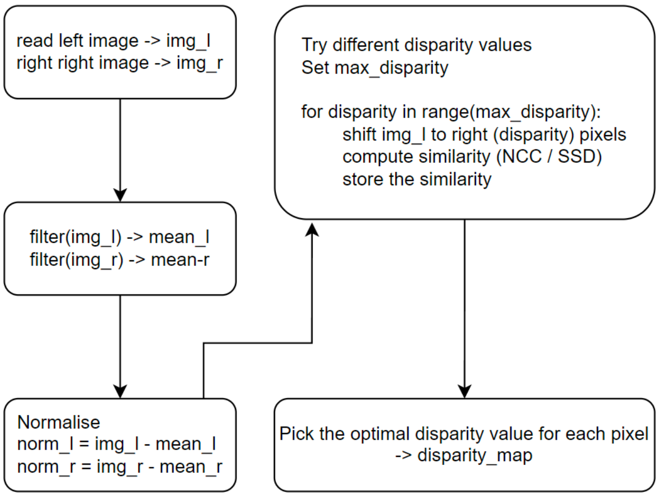
read left image -> img_l
right right image -> img_r
filter(img_l) -> mean_l
filter(img_r) -> mean-r
Normalise
norm_l = img_l - mean_l
norm_r = img_r - mean_r
Try different disparity values
Set max_disparity
for disparity in range(max_disparity):
shift img_l to right (disparity) pixels
compute similarity (NCC / SSD)
store the similarity
Pick the optimal disparity value for each pixel
-> disparity_map

Fig. 3. Peusdocode

There are 3 key components in the algorithm: max disparity, similarity function and filter.

A. *Max disparity*

This parameter determines how many disparity values will the algorithm explore. The larger it is, the longer time it will take to generate the disparity map. Therefore, it is better to estimate a reasonable max value of disparity at the start.

One possible solution is to first perform pixel matching and find the matching pair with the lowest depth. Their disparity may be considered as a good reference to setting the range. Moreover, the max value in ground truth files can be the optimal value, though this may not be rigorous since normally ground truth is not provided.

In brief, a large value of max disparity should normally generate acceptable results, at the expense of more time complexity.

B. *Pixel Matching*

There several similarity functions for pixel matching, such as sum of squared differences (SSD), sum of absolute differences (SAD) and normalized cross correlation (NCC). We select SSD and NCC to perform experiments since SAD and SSD draw the same results.

$$SAD\,(i,j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \left| I\,(i+m, j+n) - T\,(m,n) \right|$$

$$SSD\,(i,j) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \left(I\,(i+m, j+n) - T\,(m,n) \right)^2$$

$$NCC\,(i,j) = \frac{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I\,(i+m, j+n) \cdot T\,(m,n)}{\left(\sqrt{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} I\,(i+m, j+n)^2}\right) \cdot \left(\sqrt{\sum_{m=0}^{M-1} \sum_{n=0}^{N-1} T\,(m,n)^2}\right)}$$

Fig. 4. Similarity functions

C. *Filter*

The filter is applied so as to smooth the results. The larger the window size, the smoother the disparity map, but more details will be lost, vice versa. In addition, two types of filters are implemented, including uniform filter and gaussian filter. The former one assigns equal importance for all the pixels in the window while the latter one follows a gaussian distribution. The gaussian filter performs much better in general, which will be presented in later section. And it is applied in our final algorithm.

D. *Sub-pixel Accuracy*

To achieve sub-pixel accuracy, upsampling is added to the basic algorithm. In addition to the basic disparity map, another one with double resolution is also generated. The images are first resized using linear interpolation. Other interpolation strategies are tested as well and it seems the differences are negligible. The up-sampled disparity map is expected to give more details so as to help the result approaching the ground truth.

E. *Optimizations*

As mentioned above, larger window size can smooth the depth but lose more details. As shown in figure 5, results generated from small window sizes tend to have more noises in near scene. Also the color of roads are similar so that there is a high likelihood of mismatching. To reduce the noises, we attempt to combine different results, including a normal map, a doubly up-sampled map and a map using 3 times of the normal window size. Figure 6 presents an improved method. It firstly generates 3 disparity maps and compute a depth ratio that represents the pixels relative depth in the scene. If the ratio is smaller than the threshold, mean of the nearest pixels in the up-sampled map will be extracted as the final value, otherwise the original value will be kept.

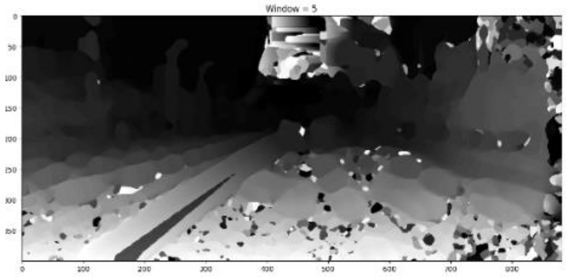

Fig. 5. Example of disparity map with low window size

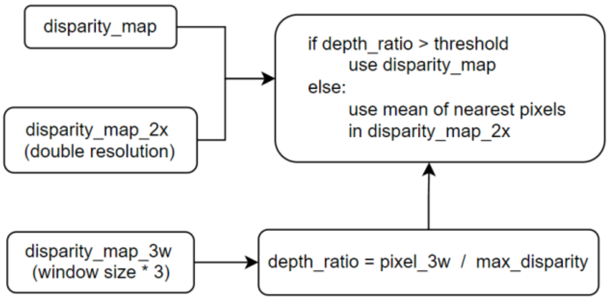

Fig. 6. Improved version

Besides, to accelerate the process, images are read as numpy arrays so that numpy functions can be applied to operate images. For example, numpy.roll() is used to shift the left image.

III. EXPEIMENTS & RESULTS

Firstly, combinations of different similarity functions and filters are tested as shown in figure 7. Obviously, gaussian filter with NCC (right top) produces the satisfactory result. Therefore, this combination is took as our final decision.

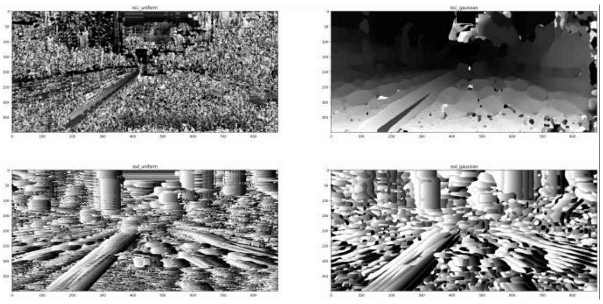

Fig. 7. Different similarity functions and filters

TABLE I. STATISTICS OF DIFFERENT WINDOW SIZES

Size	3	4	5	6	7	8	9	10
RMS	25.48	25.87	25.84	25.51	25.46	25.4	25.31	25.31
Similarity	0.9	0.89	0.89	0.89	0.9	0.9	0.9	0.9
% < 4	0.735	0.808	0.847	0.859	0.85	0.832	0.808	0.785
% < 2	0.604	0.655	0.678	0.677	0.662	0.644	0.624	0.604
% < 1	0.441	0.472	0.485	0.483	0.475	0.463	0.45	0.439
% < 0.5	0.182	0.194	0.199	0.198	0.196	0.193	0.189	0.185
% < 0.25	0.182	0.194	0.199	0.198	0.196	0.193	0.189	0.185
Runtime	1.55	1.75	2.05	2.05	2.34	2.47	2.57	3.00

Results generated from window sizes ranging from 3 to 10 are presented in figure 8. The best size is 6 and statistics are listed in table 1, including root mean squared error (rms error), similarity, fractions of pixels with errors less than 4, 2, 1, 0.5

and 0.25 pixels, and runtime. The runtime is based on i7-12700k.

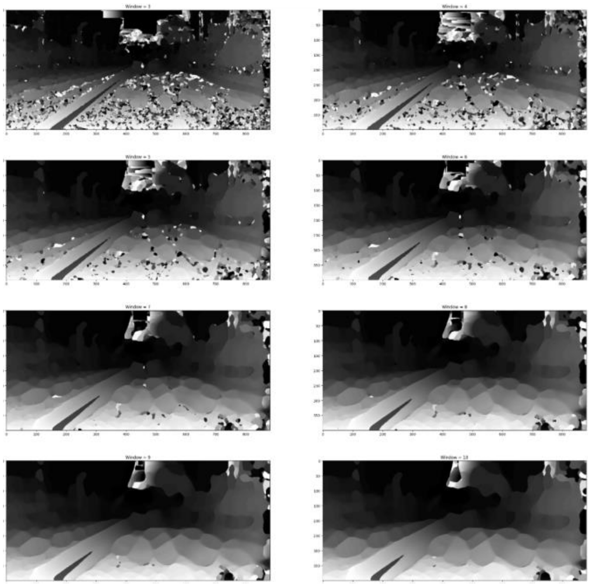

Fig. 8. Window size from 3 to 10

An example of result generated from the improved method is shown in figure 9, including comparison to the basic one and the ground truth. Table 2 presents its statistics.

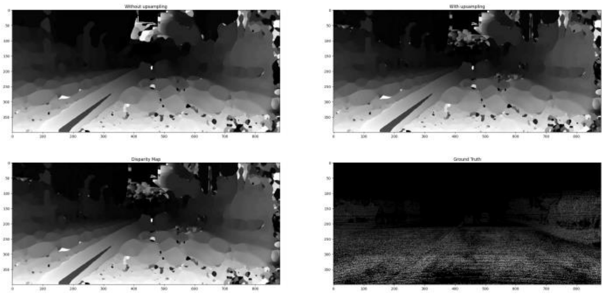

Fig. 9. Example final map

TABLE II. STATISTICS OF EXAMPLE FINAL MAP

• RMS Error	25.03
• Similarity	0.90
• % Error < 4	0.86
• % Error < 2	0.68
• % Error < 1	0.48
• % Error < 0.5	0.20
• % Error < 0.25	0.19
• Runtime	13.84 s

Table 3 presents the average performance of our algorithm over all 25 stereo image pairs.

TABLE III. STATISTICS OF AVERAGE PERFORMANCE

• RMS Error	27.53
• Similarity	0.89
• % Error < 4	0.82
• % Error < 2	0.67
• % Error < 1	0.50
• % Error < 0.5	0.21
• % Error < 0.25	0.19
• Runtime	14.83 s

IV. EVALUATION & ERROR ANALYSIS

Figure 10 shows our best result compare with ground truth(upper one is our result, lower one is ground truth by LIDAR), it has 88% of pixel with error less than 4, 64% of pixel with error less than 1. We can generare more accurate result in distant area than ground truth and also the nearby one. The ground truth distal end is blurred and indistinct as shown in red box of figure. Since ground truth images is generated by LIDAR, it will decrease performance once object is to far and cause uncleared result to produce. However our disparity algorithm is mainly based on image colour, therefore, it can generate a better result in distant area.

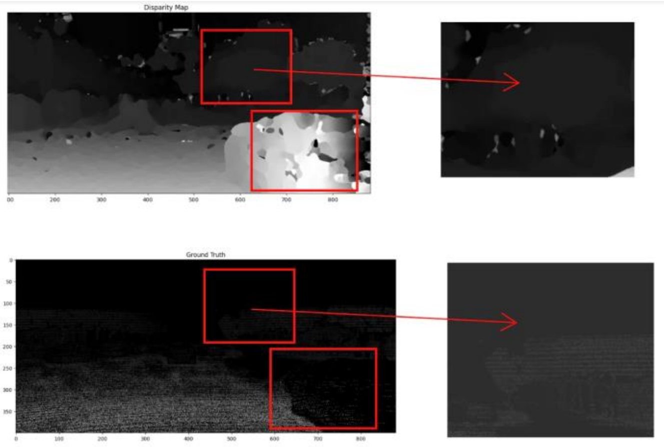

Fig. 10. Best output dataset compare with ground truth

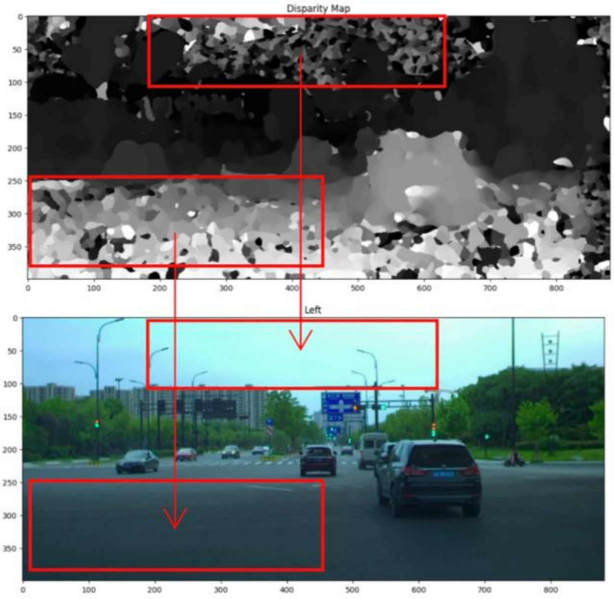

Fig. 11. Worst output dataset compare

Figure 11 shows the worst output of dataset. It has 68% of pixel with error less than 4, 42% of pixel with error less than 1. The unmatching pixels are mainly concentrated in left down corner and sky(shown in red boxs). These two areas generate to much noise to result. And these two areas is mainly formed by same color pixels. Since NCC or SSD are evaluated from nearest pixels, without much colour change around, filter is hard to generate edges and leading wrong disparity map.

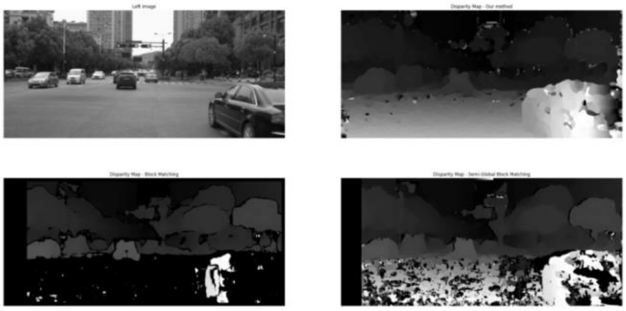
Left image
Disparity Map - Our method
Disparity Map - Block Matching
Disparity Map - Semi-Global Block Matching

Fig. 12. Compare to StereoBM & StereoSGBM

Figure 12 shows the comparison between our result with others algorithm, in the right corner is our result, left bottom is Block Matching, right bottom is semi-global block matching. And clearly we have the best result in nearby areas. Distant area of StereoBM and StereoSGBM is better than us, however, this dataset is based on autopilot, where near objects need to be given more weight.

V. Conclusion

In this paper, we have proposed an approach for disparity estimation based on the NCC algorithm and improved it using a sliding window(filter) and up-sampling images to a higher resolution. With this method we achieve average fraction of pixels with error less than 4 is 82%, less than 2 is 67%, less than 1 is 50%, less than 0.5 is 21%, less than 0.25 is 18%. Up-sampling is used primarily in distant views since the higher the image resolution it is, the more detailed the disparity map it will generate. Also, most of the data sets we processed are real road conditions, therefore, in the near distance, the image will most likely be all roads with pixels with the same colour, and an NCC model will produce noise only. That is why we use gaussian and no up-sampling in the near distance.

In future work, we will focus mainly on reducing incorrect disparity estimates of near-distance pixels with other techniques like deep learning and improving the time complexity of calculating the disparity map. Also, cuda tensor and multithread might be added to increase evaluation speed of our algorithm.

References

[1] Jian Sun, Nan-Ning Zheng and Heung-Yeung Shum, "Stereo matching using belief propagation," in IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 7, pp. 787-800, July 2003, doi: 10.1109/TPAMI.2003.1206509.J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[2] S. Mukherjee and R. M. R. Guddeti, "A hybrid algorithm for disparity calculation from sparse disparity estimates based on stereo vision," 2014 International Conference on Signal Processing and Communications (SPCOM), 2014, pp. 1-6, doi: 10.1109/SPCOM.2014.6983949.K. Elissa, “Title of paper if known,” unpublished.

[3] G. Yang, X. Song, C. Huang, Z. Deng, J. Shi and B. Zhou, "DrivingStereo: A Large-Scale Dataset for Stereo Matching in Autonomous Driving Scenarios," 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 899-908, doi: 10.1109/CVPR.2019.00099.