COMP90086 Computer Vision, 2022 Semester 2 Group Project Report: Stereo Disparity

Chenyang Dong
Student ID: 1074314
doncd@student.unimelb.edu.au

Xinhao Yan
Student ID: 1200713
xinhao1@student.unimelb.edu.au

***Abstract*—In order to address the issue of stereo disparity in an autonomous driving scenario, a local area-based matching method is presented with zero-mean normalized cross correlation as cost function. By balancing the performance on the fractions of disparity in small-error ranges and root-mean-squared error between generated disparity map and the ground truth, a relatively optimal window size is selected for the matching. Methods including smoothing, applying Gaussian filter and sub-pixel technique are used to enhance the algorithm.**

***Index Terms*—stereo matching, disparity estimation**

I. Introduction

Stereo matching is one of the fundamental techniques in computer vision for generating disparity map and recovering 3D structure of the scenes. It has been extensively applied in fields like autonomous driving, mobile robotics and augmented reality (AR). Unlike many recent research focusing on developing deep learning approaches, this work uses the traditional methods to address with the correspondence problem and compute the disparity map.

II. Dataset

In this project, the dataset is curated from [1] which gives existing stereo image pairs taken from a moving vehicle at the same time in an autonomous driving scenario. Also, with the ground truth disparity images, the performance evaluation of our algorithm can be implemented. An example image pairs used for experiment is shown in Fig. 1 which the ground truth image is brighten up for display.

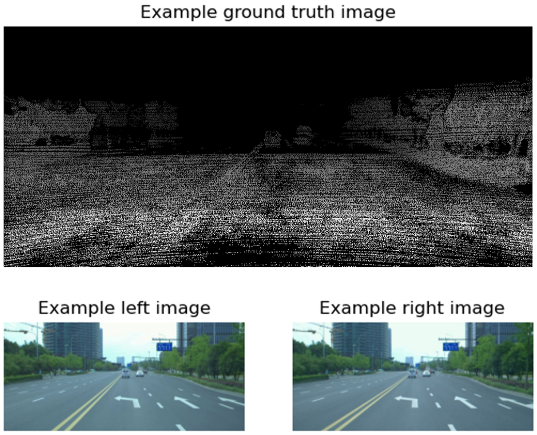


Fig. 1. The Example Images for Experiment

III. Method

The algorithm uses a local area-based stereo matching technique to obtain the correspondence. For stereo cameras, since the epipolar lines always coincide with the horizontal lines, the corresponding points in two images must also locate on the same horizontal line. By setting a window size and step distance, the surroundings of a pixel (depend on the window size) in the left image can be compared with corresponding windows on the same horizontal line in the right image. This general stereo matching process is shown in Fig. 2 [2]. Thus, the most correlated area can be found by using certain matching cost functions.

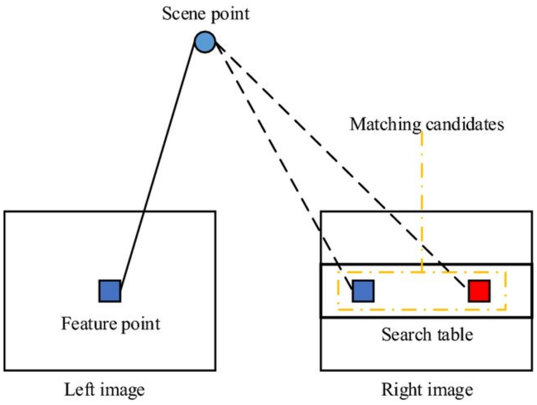


Fig. 2. Stereo Matching Process

A. Design choice

To have a good quality of disparity map, choosing a suitable matching cost function is essential. Common cost functions include sum of squared differences (SSD) [3] and normalised cross correlation (NCC). Studies such as [4] shows that NCC would have significant resistance to white noise interference and excellent matching accuracy with little gray-scale change or geometric distortion; however more likely to be impacted by local light variation.

To pursue better performance of our algorithm, we also attempt to develop zero-mean based costs such as ZSSD and ZNCC to consider the effect of global variations. For zero-mean based costs, they would have the mean first extracted from both the template window and matching window before computing the inner product of two images. Besides, ZNCC does not only inherit the advantage from NCC, but also provides better robustness on brightness variation [5]. Thus, the experiment starts with three cost functions, SSD, ZSSD and ZNCC and the further experiment will be mostly conducted on the one with best performance.

IV. Analysis

A. Experiment

a) Window size: In the area-based stereo matching, window size is one of the most significant factor to affect the result. In regions with inconsistent disparity, small windows usually calculate more reliable matching results. However, in low-texture region small windows will have large matching uncertainties compared to large windows.

In Fig. 3-5, the performance of applying different matching cost function (SSD, ZSSD and ZNCC) under a range of window sizes are presented. Each figure includes the fraction of pixels with certain error ranges (purple line as 'less than 4 pixels'; red line as 'less than 2 pixels'; green line as 'less than 1 pixel'; orange line as 'less than 0.5 pixel'; blue line as 'less than 0.25 pixel') and root-mean-squared error (RMSE) between the values in generated disparity map and those in the ground truth.

From the result, it first represents that ZNCC will outperform the matching over other matching cost functions. Also, since ZNCC performs the best on the small-error ranges at 25x25 and performs the best on the RMSE at 35x35, we assume that the matching achieves a balanced optimal result at window size around 30x30. And this window size is temporarily used for further experiment.

With the experiment going on, we will still continue exploring the effect of window size while attempting different techniques to improve the algorithm.

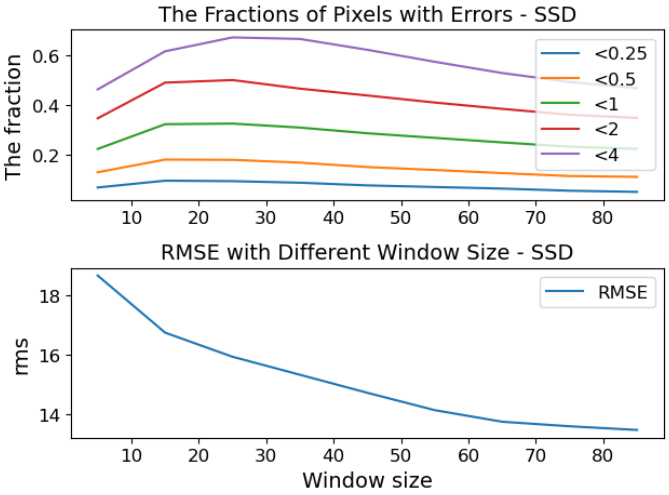


Fig. 3. The Performance of Using SSD under Different Window Size

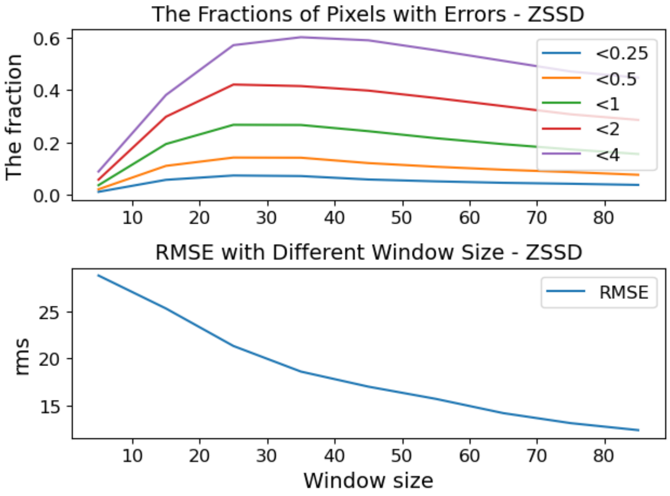


Fig. 4. The Performance of Using ZSSD under Different Window Size

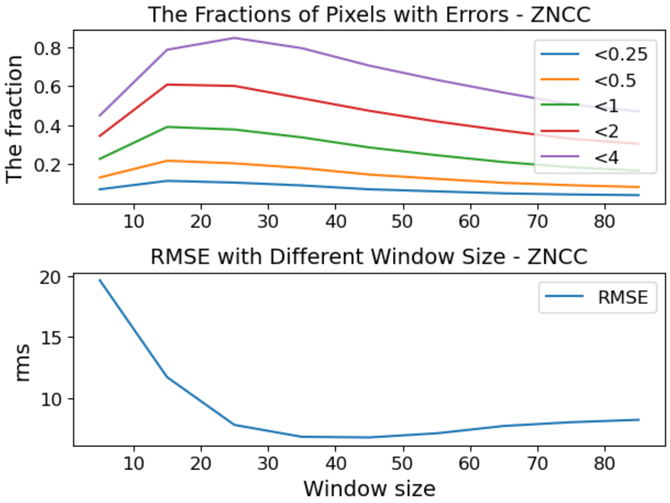
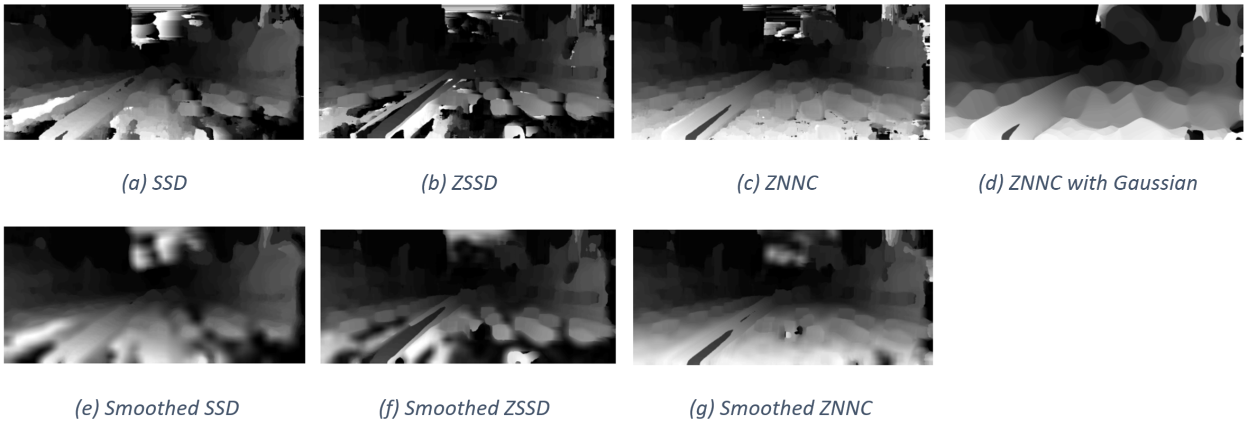


Fig. 5. The Performance of Using ZNCC under Different Window Size

b) Sub-pixel Refinement: Sub-pixels are a finer resolution representation of a parent pixel [6]. Since the disparity value obtained by the algorithm is in integer pixel, in order to obtain a higher sub-pixel accuracy, further sub-pixel refinement of the disparity value is required. The main thoughts of the sub-pixel improvement is that, for each epipolar line, after we find the best match window of the left image from the right image, we do some actions on the best match window. In this way, the space between two integer pixels are extended from 1 unit to n units ($n < 1$). We tried several ways of using sub-pixel methods to make the results better, but the results were not satisfactory.

One way is to enlarge the width of left and right images by the same factor, so that several new values were inserted between pixels. The idea is from a research that interpolated scan-lines by a factor of 4 using a cubic interpolate before computing the SSD score [7]. After getting the disparity map of the enlarged images,we resized it back to original size to see the changes. We tried all interpolation methods but the results are not as good as the original images. The other way



(a) SSD

(b) ZSSD

(c) ZNNC

(d) ZNNC with Gaussian

(e) Smoothed SSD

(f) Smoothed ZSSD

(g) Smoothed ZNNC

Fig. 6. The Disparity Map of using Different Matching functions

to do so is that is called Symmetric Sub-pixel Stereo Matching [8].we go through each pixel of the disparity map and get the difference of its left pixel and right pixel and divide it by a certain weight. In this way, we can make each pixel in the disparity map represents the sub pixel of its left or right. That's so far we tried for our sub-pixel part but they all give bad results in the end. However, we believe that With such sub-pixel technique, to a great extent it can improve the precision and effectiveness of image detection of the algorithm.

c) Importance of center pixels: We think the center pixels of a window should have greater weights and we apply Gaussian filter to our model. Gaussian filter reduces noise and will blur the edges more significantly. Gaussian filtering incorporates a weighting of the grey-scale information, for example, the closer the grey-scale value is to the center grey-scale value in the neighbourhood the greater the weight of the point, and the smaller the weight of the point with a large difference in grey-scale value. We determine the grey value of the center after convolving it with the image using a weighting coefficient kernel of spatial distance. As shown in the Fig.7, the root-mean-squared error returns a better result after we applying the Gaussian filter which proves our idea and we find out that ZNCC with Gaussian filtering can get the best results with even a small window size which will save more running time than bigger window sizes in other methods.

d) Smoothing: Apart from match quality, smoothness also defines a good stereo correspondence. Therefore, we used an algorithm based on existing research [9] and improve it by changing the smooth function to the formula $(f_p - I_p)^2 + \lambda * (\alpha - \beta)^2$ which can be represented as $E(f) = E_{data}(f) + E_{smooth}(f)$.

For the $E_{data}(f)$ part, it measures the goodness of a label f_p fits pixel p by using the formula $(f_p - I_p)^2$ (I_p is the observed intensity of p).

For the $E_{smooth}(f)$ part of the formula,λ is the weight of the smooth function. Lasso regression $|\alpha - \beta|$ (a.k.a L1 or absolute distance) and ridge regression $(\alpha - \beta)^2$ (a.k.a L2 or truncated quadratic) were talked about and they picked the lasso regression to do the smoothness for graphics. Both of the smooth methods are well-known penalty function to reduce the model complexity and over fitting problems. However, it proves that ridge regression for the smoothness of disparity map in our project has a better root-mean-squared error than the lasso regression as we have different scopes compared with the research.

Fig. 6 shows the effect of applying smoothed function onto disparity maps and the problem of discontinuity gets solved to certain extent. To further compare how the smoothing help with the matching quality on statistics, Fig. 7 shows that smoothing help decrease RMSE around 9%. Besides, from Fig. 8, the peak of the fractions of disparity in small-error ranges moves closer to a higher window size around 30x30 where we expected to have a balanced optimal performance with RMSE.

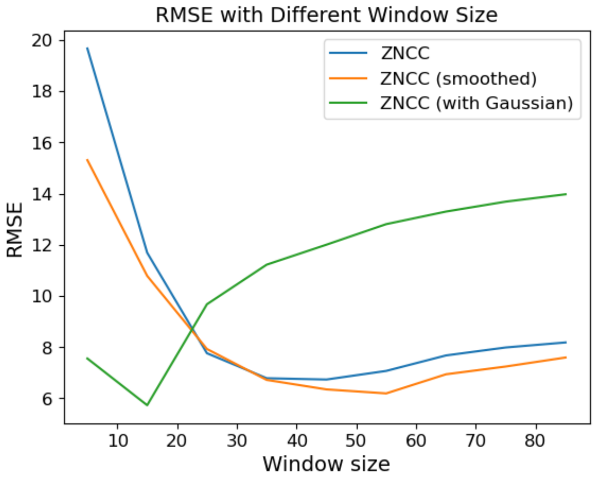


Fig. 7. The Performance Comparison on RMSE Between ZNCC, Smoothed ZNCC and ZNCC using Gaussian Filter under Different Window Size

B. Result

After a series of experiment, our final algorithm uses two set of designs, which both uses zero-mean normalised cross correlation as the matching cost function.

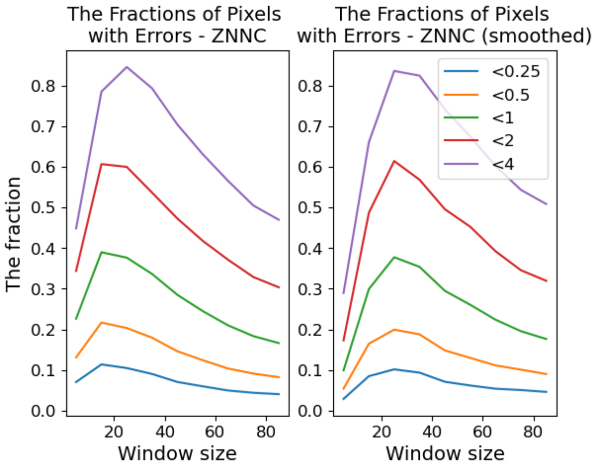


Fig. 8. The Performance Comparison on Fraction Between ZNCC and Smoothed ZNCC under Different Window Size

The first one uses a designed smoothed function to further improve its performance by 9% on average and sets a fixed window size at 30x30 to achieve a balanced optimal performance between RMSE and fractions of disparity in small-error ranges. The other one applies a Gaussian filter with a fixed window size at 15x15 instead.

By testing on all 15 sets of image pairs provided in the dataset, we compute the average performance of our algorithm shown in Table. I-II, which both give a satisfied result. By comparison, algorithm one using smoothed ZNCC gives better performance on fraction of pixels in small-error ranges, while algorithm two using ZNCC with Gaussian filter performs better on RMSE.

TABLE I
AVERAGE PERFORMANCE OF ALGORITHM 1 (SMOOTHED ZNCC) ON ALL SET OF IMAGES

Root Mean Squared Error	**Fraction of Pixels in Small-error Ranges**				
	< 0.25px	*< 0.5px*	*< 1px*	*< 2px*	*< 4px*
7.74	9.28%	17.95%	32.32%	51.52%	72.17%

TABLE II
AVERAGE PERFORMANCE OF ALGORITHM 2 (ZNCC WITH GAUSSIAN FILTER) ON ALL SET OF IMAGES

Root Mean Squared Error	**Fraction of Pixels in Small-error Ranges**				
	< 0.25px	*< 0.5px*	*< 1px*	*< 2px*	*< 4px*
7.25	8.16%	15.92%	28.61%	44.73%	61.68%

C. Error analysis

We have twenty five packs of image with left image, right image and ground truth in total. When we go through all packs of images, they are captured in great conditions. Even in this condition, some pack of images are showing better results than others just because some small noises. It may be less sun light condition or occlusion by cars which gives a bad depth of focus. Real-life situations may be worse than our data-set, the cloudy and rainy days may give less sun light conditions and make images obscure. Buildings with high reflections, sky which give less information should all be considered as bad effects. In the ground truth images, we can see that the building and sky are all set to zero as invalid information. This is a good point to apply in our project but our algorithms do not have abilities to recognize these information and invalid them.

Meanwhile, we put more attention on the root-mean-squared error more than the fractions of disparity in small-error ranges. root-mean-squared error is a good accuracy measurement but it is scale-dependent which is only good at model for a particular variable [10]. As we can see from the Fig.5, when we find the smallest root-mean-squared error by changing the window size, the fractions of disparity in small-error ranges tend to drop down a little bit. That means we may take some high noise value into our disparity map which causes over-fitting problem. There should be a more general range of window sizes to choose for all images when we consider both root-mean-squared error and the fractions of disparity in small-error ranges.

V. CONCLUSION AND FUTURE WORK

A local area-based matching algorithm is presented to approach the problem of stereo disparity in a autonomous driving scenario. By comparing the performance between different cost functions, zero-mean normalised cross correlation (ZNCC) is selected. The window size of the matching is also experimented by looking for a balanced performance on the root-mean-squared error (RMSE) and the fractions of disparity in small-error ranges. By applying a Gaussian filter on the matching window, it is found that for pixels closer to the center are treated more importantly than those are further away. A penalty function is also used to smooth the disparity map which further improves the algorithm on correspondence.

Since the algorithm only uses the local gray-scale information from the image, it is very sensitive to noise, low-texture object and occlusion. Especially while we are using a fixed window size, it is difficult to obtain a high matching accuracy in varying scenarios. In the future experiment, an adaptive window size can be implemented to cope with different situations. For instance, using a larger window in low-texture regions, while in the high-texture regions, using a smaller window to protect the detail information such as object edges.

REFERENCES

[1] 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," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[2] C. Tang, X. Zhao, J. Chen, L. Chen, and Y. Zhou, "Fast stereo visual odometry based on lk optical flow and orb-slam2," *Multimedia Systems*, 06 2020.

[3] S. Birchfield and C. Tomasi, "A pixel dissimilarity measure that is insensitive to image sampling," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 4, pp. 401–406, 1998.

[4] M. Hisham, S. N. Yaakob, R. Raof, A. A. Nazren, and N. Wafi, "Template matching using sum of squared difference and normalized cross correlation," in *2015 IEEE Student Conference on Research and Development (SCOReD)*, 2015, pp. 100–104.

[5] L. Di Stefano, S. Mattoccia, and F. Tombari, "Zncc-based template matching using bounded partial correlation," *Pattern Recognition Letters*, vol. 26, pp. 2129–2134, 10 2005.

[6] K. Mertens, L. Verbeke, and R. Wulf, "Sub-pixel mapping : a comparison of techniques," 01 2006.

[7] R. S. L. H. Matthies and T. Kanade, "Kalman filter-based algorithms for estimating depth from image sequences," 01 1989, pp. 209–236.

[8] R. Szeliski and D. Scharstein, "Symmetric sub-pixel stereo matching," 01 2002, pp. 525–540.

[9] O. V. Yuri Boykov and R. Zabih, "Fast approximate energy minimization via graph cuts," 01 2001, pp. 1222–1239.

[10] S. P. N. David Christie, "Measuring and observing the ocean renewable energy resource," 01 2022, pp. 149–175.