COMP90086 Computer Vision
Final Project: Stereo Disparity

Jun Li Chen
1043258

Tianli Cheng
1128434

I. INTRODUCTION

II. METHODOLOGY

III. EVALUATION

IV. IMPROVEMENTS

In digital image processing, different similarity measurement functions exist including the sum of absolute difference (SAD), the sum of square difference (SSD), etc. Six functions have been chosen among these to make a performance comparison in this report. They are SAD, SSD, the zero-mean sum of absolute difference (ZSAD), the zero-mean sum of square difference (ZSSD), and hamming distance of census transform (HDCT).

SAD of block W1 and block W2 was calculated by summing up the intensity differences between each corresponding pixel pair, as in:

$$SAD = \sum_{i=0}^{n} \sum_{j=0}^{m} |W1[i,j] - W2[i,j]|$$

Similarly, the SSD of block W1 and block W2 was calculated by summing up the square of intensity differences between each corresponding pixel pair, as in:

$$SSD = \sum_{i=0}^{n} \sum_{j=0}^{m} (W1[i,j] - W2[i,j])^2$$

SSD is more discriminative compared to SAD since the differences between image patches are amplified by squaring [1]. It means that SSD is more sensitive to large differences compared to SAD. In case there is one single pixel holding a value that is much larger than the corresponding pixel in the other image block compared to the rest, the dissimilarity score given by SSD can be much higher than what will be given by SAD. It means that those pixels holding small differences are more likely to be ignored by SSD.

Hence, two other similarity measurement functions, ZSAD and ZSSD, are chosen to perform a certain degree of normalization avoiding the result being biased by large values. ZSAD between blocks is calculated by summing up the deviation differences between each corresponding pixel pair, as in:

$$ZSAD = \sum_{i=0}^{n} \sum_{j=0}^{m} |(W1_{[i,j]} - \frac{1}{nm}\sum_{i=0}^{n}\sum_{j=0}^{m}W1_{[i,j]}) - (W2_{[i,j]} - \frac{1}{nm}W2_{[i,j]})|$$

It first finds the differences between each element and the mean intensity value in both image blocks, then finds the absolute differences of the corresponding deviation. Similarly, ZSSD was calculated by summing up the square of deviation differences between each corresponding pixel pair, as in:

$$ZSSD = \sum_{i=0}^{n} \sum_{j=0}^{m} ((W1_{[i,j]} - \frac{1}{nm}\sum_{i=0}^{n}\sum_{j=0}^{m}W1_{[i,j]}) - (W2_{[i,j]} - \frac{1}{nm}W2_{[i,j]}))^2$$

By only keeping the deviation, these functions are less sensitive to a general shift in intensity which can be caused by a change in lightning. However, normalization can be taken further by including the relation of a single pixel with the image block it belongs to into consideration. Normalized Cross-Correlation (NCC) is one of the most classic metrics including local relations when mapping similarity between images. It sums the cross-correlation between blocks and then divides it by the product of the square root of the sum of the square of both image blocks, as in:

$$NCC = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m}W1_{[i,j]} * W2_{[i,j]}}{\sqrt{\sum_{i=0}^{n}\sum_{j=0}^{m}W1_{[i,j]}^2} * \sqrt{\sum_{i=0}^{n}\sum_{j=0}^{m}W2_{[i,j]}^2}}$$

NCC is less sensitive to a linear change in lightning intensity but computationally expensive [3]. Different approaches, such as coarse-to-fine and multi-resolution search, exist to reduce the computational complexity for NCC, but these include much more complex methods like neural networks beyond the scope of this report. Hence, HDCT has been selected as an alternative to NCC and also normalized the image pattern by local correlation.

The census transform records each pixel holding a lower intensity compared to the center pixel as 0 and the rest as 1. Figure 1 gives an example of how census transform works in a 3 by 3 window.
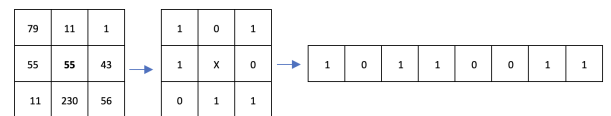

Fig. 1. Census transform

It represents each pixel's relation to the center pixel in a binary form. Therefore, it is non-parametric and insensitive to radiometric variations [2].

The Hamming distance of the census transform between two blocks is calculated by performing a XOR operation on the two 1-D binary representations. By counting the number of 1 existing in the outcome of XOR between blocks, the result represents the number of different patterns. Figure 2 gives an example of how it works.

| 0 | 1 | 1 | 1 | 0 | 1 | 0 | 0 |

XOR

| 1 | 0 | 1 | 1 | 0 | 0 | 1 | 1 |

=

| 1 | 1 | 0 | 0 | 0 | 1 | 1 | 1 |

Fig. 2. Hamming distance of Census transform

It is found that the computational time of pixel-wise disparity calculation is extremely long in this case since experiment photos all have 400 by 881 pixels. The function will iterate $400 * 881 * 88 = 31011200$ times if it searches every pixel on the left image and finds correspondence on the right image on the same line. The iteration must be deducted to improve the method's feasibility.

By setting a search size, the program will not iterate over all pixels laying on the same line on the right image but only within a defined searching window of fixed size. Meanwhile, the function will search those image blocks with center coordinates closer to the original coordinate of the block cropped from the left image first, since the correspondence is more likely to be found in those locations.

The number of iterations is further deducted by assuming each pixel in the image block holds the same disparity. By replacing pixel-wise correspondence searching with block-wise correspondence searching, the maximum number of iterations becomes:

$$\frac{400 * 881}{block\_width * block\_height * window\_size}$$

V. Comparison / Conclusion

References

[1] Schmidt, A., Kraft, M. and Kasiński, A., 2010. An Evaluation of Image Feature Detectors and Descriptors for Robot Navigation. Computer Vision and Graphics, pp.251-259.
[2] Sarika, S., Deepambika, V. and Rahman, M., 2015. Census Filtering Based Stereomatching under Varying Radiometric Conditions. Procedia Computer Science, 58, pp.315-320.
[3] Tsai, D. and Lin, C., 2003. Fast normalized cross correlation for defect detection. Pattern Recognition Letters, 24(15), pp.2625-2631.