# Modified SAD Using Adaptive Window Sizes for Efficient Stereo Matching

Yong-Jun Chang and Yo-Sung Ho
Gwangju Institute of Science and Technology(GIST)
123 Cheomdangwagi-ro, Buk-gu, Gwangju 500-712, Rep. of Korea
Email: {yjchang, hoyo}@gist.ac.kr

***Abstract*—If we use adaptive window sizes instead of a fixed size, the stereo matching algorithm can improve quality of the disparity map. However, it also increases the calculation of stereo matching. For this reason, we present the modified stereo matching method. The proposed algorithm uses two methods. First, we modify the sum of absolute differences (SAD) method. The second method exploits adaptive window sizes to acquire improved disparity maps. We use the modified SAD to overcome increase of computational complexity generated by using adaptive window sizes. Compared with the conventional local window matching, the proposed algorithm allows faster disparity map generation with a small loss in bad pixel rates.**

***Keywords—stereo matching; disparity map; local window; adaptive window; SAD***

## I. INTRODUCTION

Over the past few years, 3D entertainment systems have attracted great popularity in many areas, especially in 3D films. To generate 3D images, we need depth information of images. We can get depth information using two methods. First one is passive depth sensing and second one is active depth sensing. Passive depth sensing uses multi-view images to calculate depth information [1]. It is similar to human visual system which uses binocular disparities. Active depth sensing calculates depth information directly by using a time-of-flight (TOF) camera. It measures the depth through a laser or an infrared ray (IR) [2].

Each method has some advantages and disadvantages. Passive depth sensing does not constrained in place when it calculates depth information, because it uses multi-view images already captured. However, since depth information is obtained by using the disparity of images, it has lower accuracy than active depth sensing which calculates depth information directly. On the contrary, active depth sensing is hard to use in open spaces such as outdoors because this method uses the laser to obtain depth information.

The depth estimation using stereo matching methods is the typical way of passive depth sensing. In general, stereo matching methods are either local methods or global methods. Prior to stereo matching, image rectification of stereo images is required. Image rectification is an essential process to make stereo images be located on the same epipolar line [3]. We can do this by using the 2D homography matrix. The 2D homogaraphy matrix is made of original camera parameters and estimated camera parameters.

Local methods in the stereo matching mean that comparing the matching cost of each pixel between the left image and the right image. The matching cost can be calculated by similarity measures such as sum of squared differences (SSD), normalized cross correlation (NCC) and SAD, etc [4]. SSD and SAD find the minimum matching cost using the difference of two pixels to find the disparity. NCC calculates the correlation of two pixels therefore NCC finds the maximum matching cost to get the disparity. Local methods are fast but they have matching ambiguities at vertical lines and texture-less areas of stereo images.

Global methods can reduce those matching ambiguities. It is represented through an energy function of Markov Random Field (MRF), and the disparity value of global methods is determined to satisfy the minimum energy function. There are several ways to optimize energy function such as dynamic programming, belief propagation and graph cuts [5-7]. In the optimization process, global methods have the cost that increases the algorithm's complexity. Accordingly, it has longer execution time than local methods.

## II. LOCAL WINDOW MATCHING

When we use local methods in the stereo matching, comparing pixels one by one has the matching ambiguity. Because the pixel which has the same value at the same scan line may exist on the other positions. For this reason, we should apply the local window matching to local methods.

The local window matching calculates the matching cost considering for neighbor pixels including the current pixel. It increases the probability of matching accuracy. The window size can be determined by users 3×3, 5×5, etc. Fig. 1 shows the example of local window matching.

In Fig. 1, the window of left image is the reference window and the window of right image is the comparing window. At the same scan line, the reference window is in fixed state at position *(x, y)* and the comparing window moves within search range to check the matching cost. If the comparing window has the optimal cost at pixel position *(x-i, y)*, then the differences of each window's position *i* become the disparity value of pixel position *(x, y)* in the left image.

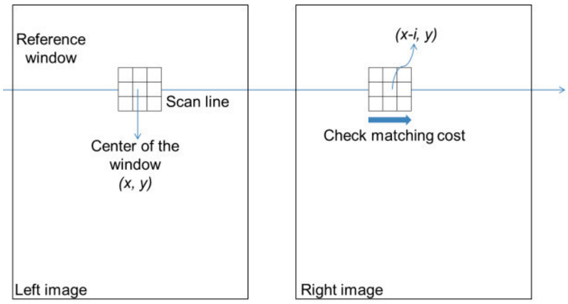



Fig. 1. Example of local window matching

In order to determine the optimal matching cost, SAD, one of the similarity measures, is calculated as in Eq. (1). SAD calculates the cost of color differences. In the case of using the SAD algorithm to measure the image similarity between stereo images, the optimal cost represents the lowest cost.

$$SAD(d) = \sum_i \sum_j |I_L(x+d+i, y+j) - I_R(x+i, y+j)| \quad (1)$$

In (1), *d* is the disparity between the left image and the right image. $x$ and $y$ are arbitrary positions of images in the 2D coordinate. $i$ and $j$ are search range. Each of $I_L$ and $I_R$ means the left image and the right image.

## III. Proposed Stereo Matching Algorithm

Disparity maps are obtained by using the local window matching. But it has some ambiguities depending on the window size. The small size of window can handle scene details but it has a weakness of image noises. On the contrary to this, the large size of window is strong at image noises, while weak at scene details. For this reason, many alternative methods have been proposed.

We can enhance the accuracy of disparity maps by using adaptive window sizes [8]. T. Kande et al. proposed the algorithm of adaptive window sizes by evaluating the local variance of the intensity and the disparity. However, if we use adaptive window sizes, they result in the increase of calculation complexity. It can be reduced if we use the modified SAD matching algorithm [9, 10]. J. Yi et al. reduce the calculation of SAD by selecting pixels in odd rows and odd columns. In this section, we propose the modified SAD using adaptive windows sizes for efficient stereo matching.

Fig. 2 represents the flowchart of the proposed method. First, we get stereo images to calculate two disparity maps using two different size windows. In this algorithm, we use 3×3 and 5×5 windows for SAD. After obtaining two disparity maps, calculate the disparity cost of each disparity map for a position *(x, y)*: *Cost_1* and *Cost_2*. If *Cost_1* is greater than *Cost_2*, then we choose the disparity value in the disparity map used the 5×5 window matching method at *(x, y)*. Conversely, if *Cost_2* is greater than *Cost_1*, then we choose the disparity value in the disparity map used the 3×3 window matching method. The final disparity map can be obtained by applying the same method to all pixels on the other positions.

This flowchart is composed of two parts. First, we explain the modified SAD algorithm. Second, we describe how to determine optimal disparity values.

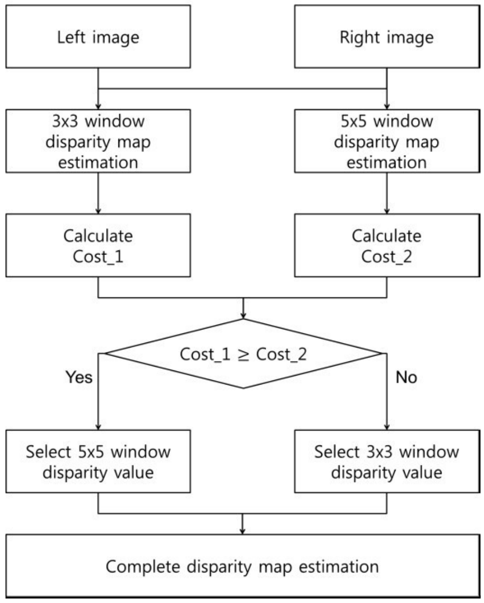
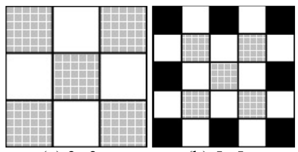



Fig. 2. Flowchart of the proposed method

### A. *Modified SAD Algorithm*

The modified SAD algorithm does not calculate all pixels in the matching window. It selects pixel candidates and uses only these pixels in the calculation process. We use outermost pixels as pixel candidates. Fig. 3 shows how to select the candidates in our algorithm. In Fig. 3 (a), we use the five pixels filled with plaid to calculate SAD. To reduce loss of matching accuracy as much as possible, we select the four outermost pixels including the center pixel. Likewise, pixel candidates in the 5×5 window are selected for the outermost eight pixels painted in black, including the five pixels of 3×3 window. It is described in Fig. 3 (b).



(a) 3×3 (b) 5×5

Fig. 3. Selecting pixel candidates

### B. *Disparity Map Estimation Using Adaptive Window Sizes*

After using modified SAD algorithm, we obtain two disparity maps. In order to get better disparity values, we compare the disparity cost of two disparity maps. The disparity cost is the degree of disparity correlation between the current pixel and neighbor pixels. The pixel value of an arbitrary point is likely to similar with neighbor pixel values. Therefore, we calculate the disparity cost using neighbor pixels based on the 3×3 window. The disparity cost is calculated as in Eq. (2).

$$Cost = \sum_{i=-1}^{1} \sum_{j=-1}^{1} |D(x,y) - D(x+i, y+j)| \quad (2)$$

*D* is the disparity values of disparity maps. *D(x, y)* is the center pixel and *D(x+i , y+j)* is the neighbor pixel.

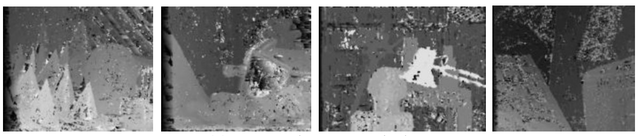
## IV. Experimental Results

In order to implement our algorithm, we use four stereo images, *Cones*, *Teddy*, *Tsukuba* and *Venus*. We experimented on these test images using the conventional SAD stereo matching. We used 3×3 and 5×5 windows to implement the local stereo matching. In the end of all experimental process, we use the median filter to remove image noises.

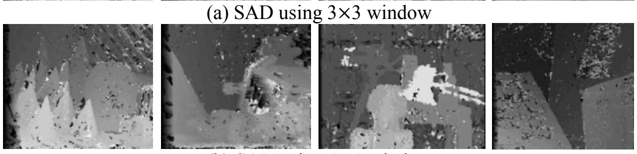
Fig. 4 and Fig. 5 represent ground truth data and disparity maps which are resulted in the conventional window stereo matching methods, respectively. In Fig. 5, we can check that the result of SAD using the 3×3 window has a lot of image noises. On the other hand, when we use the 5×5 window, the result of SAD has less image noises than the previous result.

(a) *Cones* (b) *Teddy* (c) *Tsukuba* (d) *Venus*

Fig. 4. Ground truth



(a) SAD using 3×3 window



(b) SAD using 5×5 window

Fig. 5. Results of conventional SAD algorithm

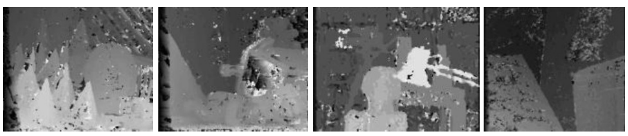


Fig. 6. Results of proposed algorithm

Fig. 6 is the result of proposed algorithm. For more detailed comparison of each result, we calculated execution time and bad pixel rate (BPR) of disparity maps.

Table I shows the performance comparison. In Table I, the conventional SAD algorithms using 5×5 windows has better BPR than using 3×3 windows. However using 5×5 windows for SAD takes longer time than using 3×3 windows. The proposed algorithm has less execution time than using only 5×5 windows and it also has almost same BPR values comparing with them. J. Yi et al. got 0.4% BPR loss in the modified SAD method and 1.8% BPR gain in the adaptive window method. On the contrary, our proposed algorithm has 0.25% BPR loss on average, using both the modified SAD and the adaptive window algorithm.

TABLE I. Performance Comparison

| Test Images | 3×3 SAD | | 5×5 SAD | | Proposed Algorithm | |
|---|---|---|---|---|---|---|
| | Time (sec.) | BPR (%) | Time (sec.) | BPR (%) | Time (sec) | BPR (%) |
| **Cones** | **1.38** | **39.34** | **3.80** | **31.53** | **2.93** | **31.51** |
| **Teddy** | **1.38** | **39.8** | **3.80** | **33.48** | **2.96** | **33.95** |
| **Tsukuba** | **0.25** | **16.86** | **0.70** | **12.88** | **0.57** | **13.13** |
| **Venus** | **0.50** | **32.12** | **1.30** | **22.88** | **0.93** | **23.17** |

## V. Conclusion

The local window matching methods acquire disparity maps by using the SAD algorithm. Depending on the window size, the results of local window matching are various. It also increases the calculation complexity when we use the large size windows. From the experimental results, we can check that our proposed algorithm reduces the execution time with less loss in quality of disparity maps compared to conventional SAD algorithms. If we use bigger and more variable size windows in our algorithm, then we expect that it would get better results.


## Acknowledgment

This research was supported by the 'Cross-Ministry Giga KOREA Project' of the Ministry of Science, ICT and Future Planning, Republic of Korea (ROK). [GK13C0100, Development of Interactive and Realistic Massive Giga-Content Technology]


## References


[1] M. Okutomi and T. Kanade, "A Multiple-Baseline Stereo," IEEE Trans. on Pattern Analysis and Machine Intelligence, pp. 353-363, April 1993.

[2] D. Scharstein and R. Szeliski, "High-Accuracy Stereo Depth Maps Using Structured Light," IEEE Conference on Computer Vision and Pattern Recognition, pp. 195-202, June 2003.

[3] Y.S. Kang and Y.S. Ho, "Efficient Stereo Image Rectification Method Using Horizontal Baseline," Lecture Notes in Computer Science, vol.7087, pp. 301-310, Nov. 2011.

[4] R. C. Gonzales and R. E. Woods, "Digital Image Processing," Prentice Hall, 3rd edtion, Aug. 2007.

[5] O. Veksler, "Stereo Correspondence by Dynamic Programming on A Tree," IEEE Conf. on Computer Vision and Pattern Recognition, vol. 2, pp. 384-390, June 2005.

[6] J. Sun, N. N. Zheng, and H. Y, Shum, "Stereo Matching Using Belief Propagation," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 25, no. 7, pp. 787-800, July 2003.

[7] Y. Boykov, O. Veksler, and R. Zabih, "Fast Approximate Energy Minimization via Graph Cuts," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 23, no. 11, pp. 1222-1239, Nov. 2001.

[8] T. Kanade and M. Okutomi, "A Stereo Matching Algorithm with An Adaptive Window: Theory and Experiment," IEEE Trans. on Pattern Analysis and Machine Intelligence vol. 16, no. 9, pp. 920-932, Sep. 1994.

[9] D. K. Hoa, L. Dung, and N. T. Dzung, "Efficient Determination of Disparity Map from Stereo Images with Modified Sum of Absolute Differences(SAD) Algorithm," IEEE Conf. on Advanced Technologies for Communications, pp. 657-660, Oct. 2013

[10] J. Yi and J. S. Kim, "Heuristic Designs of SAD Correlation Algorithm for Vision System," The Institute of Electronics and Information Engineers Trans. On Computer and Information vol. 47, no. 5, pp. 61-66, Sep. 2010.