

Stereo Matching Using Fusion of Spatial Weight Variable Window and Adaptive Support Weight

Phuc Nguyen Hong and Chang Wook Ahn

Abstract—In blocking-matching algorithms, a local window is used to measure the similarity (or dissimilarity) between pixels of a stereo pair. Although some area-based stereo matching methods have been developed and work well in many types of regions, such as textureless or object boundary regions, there are still cases where these methods are weak. In this paper, we propose an enhanced blocking-matching method to solve the correspondence problem in stereo matching. The proposed algorithm is an improved adaptive support weight method, which first classifies the pixels in the reference image and then uses the spatial weight variable window method and adaptive support weight method for each type of pixel. We also analyze and show the advantages and disadvantages of the variable window method and the adaptive support weight method and suggest solutions for cases where the two methods may not be ideal. The experimental results using the Middlebury images show that the proposed method outperform local stereo methods.

Index Terms—Stereo matching, spatial weight, adaptive window.

I. INTRODUCTION AND RELATED WORK

Stereo matching methods can be categorized in several ways [1]. Categories based on the type of disparity map can be separated into dense disparity maps, which have disparity values specified for all or nearly all pixels, and sparse disparity maps, in which only chosen points have specified disparity values. Another matching technique classification considers the matching measurement computation input format: direct techniques depend on intensity values without applying changes to the images, while indirect techniques rely on the transformation of intensity values or on features extracted from the images. Finally, stereo matching algorithms can also be classified into local and global algorithms. Local algorithms compute each pixel's disparity value from the intensity values within a window of finite size, whereas global algorithms explicitly assume an energy function with data and smoothness energy terms.

A commonly used assumption, known as photo consistency assumption, is that corresponding pixels have the same intensity value in the reference and target images. Thus for each pixel p in the reference image, the costs for p are typically computed by moving a window of pixels around pixel p along the corresponding epipolar-line in the target image and computing the differences between the pairs of windows. Pixel p is then assigned a disparity with a minimum

cost. Some popular methods for computing window-based matching costs include sum of absolute differences (SAD), sum of squared differences (SSD), and normalized cross correlation (NCC). Some methods for computing matching costs, which can work robustly in radiometric conditions, consist of adaptive normalized cross correlation (ANCC) [2], rank and census transforms [3], mutual information [4], or their hybrid [5].

Another commonly used assumption, known as smoothness assumption, in area-based stereo matching is that pixels within a support window have the same (or similar) disparity. While this smoothness assumption is typically true, it does not hold near disparity discontinuities or disparity boundaries. Furthermore, choosing the size of the matching window is very important because block-matching methods often mismatch in textureless regions. A window should therefore be small enough to include only pixels that have the same (or similar) disparity and large enough to contain pixels having sufficient variation in intensity. For that reason, many local methods have been proposed.

In the shiftable window (SW) method [6], the reference point can be located at any position in the support window, and the position with minimum cost is selected. The multiple window (MW) method [7] chooses the optimal support window from a number of windows having the best cost. In the MW method, support windows have various shapes, but the size is unchanged. The support windows in the SW have a fixed size and shape, while in the MW they have a fixed size and a limited number of shapes; as a result, these methods can be mismatched with real-world images, since objects usually have arbitrary sizes and shapes. The variable window (VW) method [8] enhances the above two methods by using a useful range of window sizes and shapes and creating a window cost that is suitable for comparing windows of different sizes. Even though a number of different sizes and shapes are deployed using the variable window method, the number is still very small in order to satisfy the smoothness assumption that all pixels in a support window have the same (or similar) disparity.

Adaptive support weight methods [9], [10] implicitly deploy a type of segmentation to estimate the support window for each pixel. One previously described method [9] assumes that pixels that are similar in color and close in Euclidean distance are more likely to lie in similar disparities and then uses photometric information and geometric information to build a weighted window for each pixel, while another method [10] computes a weighted window based on the color information and geodesic distance. While the adaptive support weight methods work very well with pixels at depth discontinuities, they sometimes fail to find the correct

Manuscript received December 4, 2013; revised February 19, 2014.

The authors are with the Computer Engineering Department, Sungkyunkwan University, Suwon, Republic of Korea (e-mail: {hongphuc, cwan}@skku.edu).

disparities for pixels in textureless regions. These methods also work poorly with noise-pixel areas, where the color is often much different compared to the colors of their neighbor pixels, as well as with mixed-pixel areas, where the color is a combination of the foreground and background pixel colors, and hence the color is different from those of the adjacent pixels.

In this paper, we present an improved adaptive support weight (IASW) algorithm. The IASW algorithm takes advantage of the strengths of the adaptive support weight (ASW) method [9], but applies a spatial weight variable window (SWVW) algorithm for pixels where the ASW method may not be ideal. We also analyze the advantages and disadvantages of the variable window method and the ASW method and suggest solutions that address their weaknesses.

The post-processing step is an important step in local stereo matching which can make disparity maps much more accurate. Some commonly used post-processing techniques include left-right consistency check [11], sub-pixel interpolation [12], and image filtering techniques such as median filtering or bilateral filtering [13]. However, in this study we do not use any post-processing techniques in testing our method so that the tests will be fair and so that we might highlight the superior performance of our proposed methods. The remainder of this paper is structured as follows. In Section II, we present details of the SWVW method. In Section III, we present the IASW algorithm. The experimental results of our algorithms are reported in Section IV. Finally, conclusions are presented in Section V.

II. SPATIAL WEIGHT VARIABLE WINDOW

In this section, we present the spatial weight variable window (SWVW) algorithm, introduced in [14]. We also review the variable window method and show why the SWVW method works better than the original method.

As mentioned above, a smoothness assumption in local stereo matching is that all pixels within a support window have the same (or similar) disparity. The VW algorithm attempts to satisfy this smoothness assumption by using a range of shapes and sizes for the support windows. The shape set can be constrained to square shapes without much appreciable decrease in the accuracy of the result [8]. The window cost in the variable window algorithm can be defined as:

$$C_d(N) = \bar{e} + \alpha \cdot \text{var}(e) + \frac{\beta}{\sqrt{|N|} + \gamma} \quad (1)$$

with

$$\bar{e} = \frac{\sum_{(x,y) \in N} e_d(x,y)}{|N|} \quad (2)$$

$$\text{var}(e) = \frac{\sum_{(x,y) \in N} e_d(x,y)^2}{|N|} - \left(\frac{\sum_{(x,y) \in N} e_d(x,y)}{|N|} \right)^2 \quad (3)$$

where N is a rectangular set of pixels, $|N|$ is a window size, d is some disparity of $e_d(x,y)$ is the measurement error pixel (x,y) in the reference image with disparity d , e is the average measurement error in the support window, $\text{var}(e)$ is the

variance of the errors in the support window, and are parameters. The last term in equation (1) is larger for smaller support windows [8]. Therefore, in textureless regions where the first and the second terms of equation (1) are similar for all support windows, larger support windows are more likely to be selected.

The variable window method uses the measurement error developed in [15] to compute the window cost. Suppose $I_r(x, y)$ is the intensity value of pixel (x, y) in the reference image and $I_t(x, y)$ is the intensity value of pixel (x, y) in the target image. The measurement error [15] can be expressed as follows:

$$\left(\begin{array}{l} \left| I_r(x, y) - I_r(x - d - \frac{1}{2}, y) \right|, \left| I_r(x, y) - I_r(x - d + \frac{1}{2}, y) \right|, \\ \left| I_r(x - \frac{1}{2}, y) - I_r(x - d, y) \right|, \left| I_r(x + \frac{1}{2}, y) - I_r(x - d, y) \right|, \\ \left| I_r(x, y) - I_r(x - d, y) \right| \end{array} \right) \quad (4)$$

After the computation of window costs, each pixel is assigned a disparity for which the window cost is at minimum. The advantage of the variable window method is that all pixels in the support window, not only the center pixel, are updated by the window cost, as shown in Fig. 1. In other words, the reference pixel updated by the cost function changes in different support windows. This strategy is very useful for pixels that are near depth discontinuities, as shown in Fig. 2.

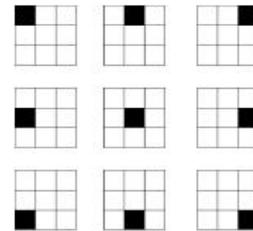


Fig. 1. When the support window size is (3×3) , the number of windows is 9. The updated position changes for each window.

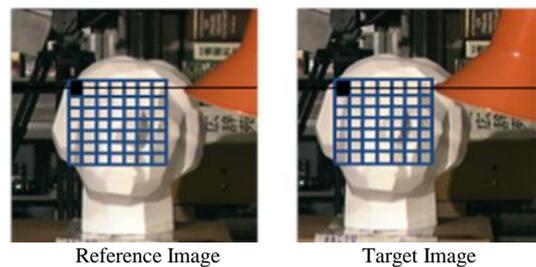


Fig. 2. The advantage of the variable window method at disparity discontinuities. All pixels in the support windows have similar disparities when the reference pixel is in the top-left.



Fig. 3. In all three cases, the support windows do not fit the lamp object, so there are some pixels in the support windows which have different disparities compared to the disparity of the lamp.

However, because objects in real-world images can have arbitrary sizes and shapes, the smoothness assumption may be frequently violated, as illustrated in Fig. 3. In order to reduce these violations, we proposed the spatial weight variable window (SWVW) method, which applies a spatially weighted window to each support window. Different support windows have different spatially weighted windows due to the position of the reference pixel in the support window. Each pixel in the support window receives a spatial weight computed by the Euclidean distance between the pixel and the reference pixel, as shown in Fig. 4. A spatial weight $w_s((x_r, y_r), (x, y))$ at pixel (x, y) can be computed as:

$$w_s((x_r, y_r), (x, y)) = e^{-\frac{d((x_r, y_r), (x, y))}{\lambda_s}} \quad (5)$$

where (x_r, y_r) is the reference pixel, $d((x_r, y_r), (x, y))$ is the Euclidean distance between pixels (x_r, y_r) and (x, y) , and λ_s is a constant that the smaller it is, the faster the spatial weight value of pixel (x, y) reduces from the spatial weight value of the reference pixel. By adding a spatial weight for each pixel in the support window, we implicitly assume that pixels which are spatially closer to the reference point have a greater probability of having the same (or similar) disparity.

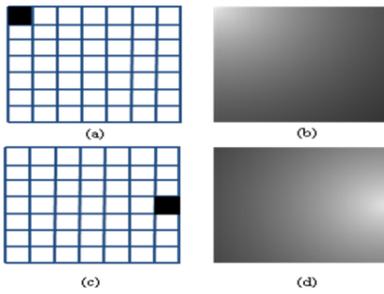


Fig. 4. Spatially weighted windows when the reference point is at two different positions in the support window. (a) shows the position of the reference pixel in the top-left, and the corresponding spatially weighted window is as shown in (b). Similarly, when the reference pixel is located at the middle-right, as shown in (c), the corresponding spatially weighted window is as shown in (d).

Unlike the original variable window method [8], the SWVW method works with color images and uses absolute difference as the measurement error. Therefore, we redefine the measurement error from equation (4) for the SWVW method as follows:

$$e_d(x, y) = w_s((x_r, y_r), (x, y)) \times \sum_{k \in \{r, g, b\}} |I_r^k(x, y) - I_t^k(x, y)| \quad (6)$$

where $I_r^k(x, y)$ and $I_t^k(x, y)$ are the intensity values of the color band k at pixel (x, y) in the reference image and the target image, respectively.

The SWVW method uses the window cost from equation (1), which was developed by [8], and each pixel is simply assigned a disparity which has the minimum window cost. By applying a spatially weighted window for each support window, we bias for pixels spatially closer to the reference pixel, while pixels that lie far from the reference pixel have

small effect on the window cost. This can make the SWVW method more robust than the variable window method when working in depth discontinuity regions.

III. IMPROVED ADAPTIVE SUPPORT WEIGHT METHOD

In this section, we present an improved adaptive support weight (IASW) algorithm, which is basically a combination of the SWVW method and the adaptive support weight (ASW) method [9]. We also show and analyze cases where the ASW algorithm may not be ideal and suggest solutions to overcome these cases.

The ASW method successfully deploys color and spatial proximity to determine the support weight for each pixel. Suppose p_c is the reference pixel in the support window in the reference image, and $I_r(p)$ is the intensity value of pixel p in the reference image. The ASW algorithm computes the reference support weight $w_r(p_c, p_i)$ of pixel p_i in the reference weighted window as follows:

$$w_r(p_c, p_i) = e^{-\frac{d_p(p_c, p_i)}{\gamma_p}} e^{-\frac{d_c(I_r(p_c), I_r(p_i))}{\gamma_c}} \quad (7)$$

where $d_p(p_c, p_i)$ is Euclidean distance between pixels p_c and p_i , $d_c(I_r(p_c), I_r(p_i))$ is the absolute difference between intensity values $I_r(p_c)$ and $I_r(p_i)$, and p and c are the proximity constant and color constant, respectively.

Similarly, suppose q_c is the reference pixel in the support window of the target image and $I_t(q)$ is the intensity value of pixel q in the target image. The target support weight $w_t(q_c, q_i)$ of pixel q_i in the target weighted window can then be computed in the same manner:

$$w_t(q_c, q_i) = e^{-\frac{d_p(q_c, q_i)}{\gamma_p}} e^{-\frac{d_c(I_t(q_c), I_t(q_i))}{\gamma_c}} \quad (8)$$

The window cost of pixel p_c and pixel q_c can then be computed as:

$$C(p_c, q_c) = \frac{\sum_{p_i \in S_r, q_i \in S_t} w_r(p_c, p_i) \times w_t(q_c, q_i) \times e_t(p_i, q_i)}{\sum_{p_i \in S_r, q_i \in S_t} w_r(p_c, p_i) \times w_t(q_c, q_i)} \quad (10)$$

where S_r and S_t are support windows in the reference image and target image, respectively, while $e_t(p_i, q_i)$ is the truncated absolute difference of pixels p_i and q_i . The $e_t(p_i, q_i)$ can then be computed as:

$$e_t(p_i, q_i) = \min \left(\sum_{k \in \{r, g, b\}} |I_r^k(p_i) - I_r^k(q_i)|, T \right) \quad (11)$$

where T is the truncation value that controls the limit of the matching cost, and $I_r^k(p_i)$ and $I_r^k(q_i)$ are the intensity.

The ASW generally work very well, particularly near the

depth discontinuities. However, the ASW method is weak in two cases. The first case is that of very small regions having a much different color compared to regions around them or that of noise or mixed-pixel areas where the color of a pixel is a combination of the foreground and background pixel colors, as shown in Fig. 5. A pixel in those regions or areas has a small sum for the reference weighted window, which is computed as:

$$Sw_r(p_c) = \sum_{p_i \in \mathcal{S}_R} w_r(p_c, p_i) \quad (12)$$

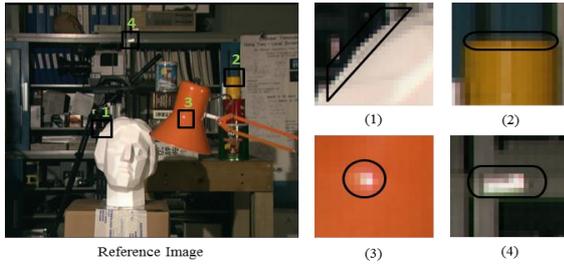


Fig. 5. Four examples of noise, small object, and mixed pixel areas in the Tsukuba image. (1) and (2) show the mixed-pixel areas, (3) presents the noise area, and (4) shows the small object. Values of the color band k at pixel p_i in the reference image and at pixel q_i in the target image, respectively.

In this case, the window cost from equation (9) is controlled by a very small set of pixels. In other words, a very large percentage of pixels in the support window at reference pixel p_c have very small reference support weights, which implicitly eliminate those pixels from the window cost, while the remaining number of pixels with enough large reference support weights and that are understood to have the same (or similar) disparity as p_c is too small, as shown in Fig. 6.

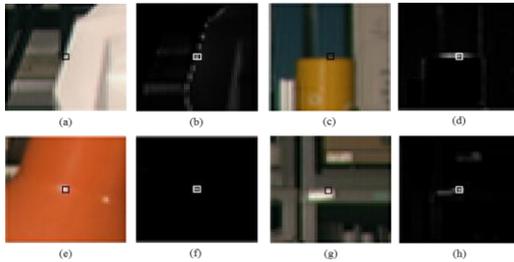


Fig. 6. The pixels having small sums of the reference weighted windows are illustrated in the Tsukuba image. The pixels in mixed-pixel areas (a) and (c), a noise area (e), or a very small object (g) in the reference image are shown, and their corresponding reference weighted windows are presented in (b), (d), (f), and (h), respectively.

Therefore, the matching result, which depends on a small set of pixels, can be unreliable. More-over, in real world situations, it is uncommon to have a tiny group of pixels with a different disparity compared to the neighboring pixels. That is why we use the SWVW method instead of the ASW algorithm for these types of pixels. The second case is that the ASW method sometimes fails to correctly find disparity in textureless regions. In a textureless region, all pixels in the region have similar intensity values, so the reference support weight from equation (7) can be rewritten as:

$$w_r(p_c, p_i) = e^{-\frac{d_p(p_c, p_i)}{\gamma_p}} e^{-\frac{d_c(I_r(p_c), I_r(p_i))}{\gamma_c}} \quad (13)$$

It can be seen from equation (13) that the reference support weight $w_r(p_c, p_i)$ at pixel p_i depends significantly on the spatial proximity to determine its value and that pixels which are spatially closer to the reference pixel p_c have more support weight value. If the textureless regions are large enough, the amplitude of the window costs of reference pixel p_c with a disparity range is small, so the disparity of p_c is unreliable. Support p_c is the reference pixel in the reference image, and q_c^d is the reference pixel in the target image, which is shifted some disparity d from p_c in the same epipolar line as p_c . The amplitude of window costs $A(p_c)$ at reference pixel p_c with the disparity range D can be expressed as:

$$A(p_c) = C_{\max}(p_c) - C_{\min}(p_c) \quad (14)$$

where

$$C_{\max}(p_c) = \max(C_{d \in D}(p_c, q_c^d)) \quad (15)$$

And

$$C_{\min}(p_c) = \min(C_{d \in D}(p_c, q_c^d)) \quad (16)$$

The amplitude of the window costs should be large enough so that the matching result becomes more reliable. Now, we make the assumption that pixels in the reference support window of a reference image having a reference support weight value in the reference weighted window greater than a constant λ_w have the same (or similar) disparity as reference pixel p_c . Based on what we assume, a weight-changing process is performed for the reference weighted window as follows: from the reference weighted window, we group pixels with support weights larger than λ_w into area(s), then, for each area, pixels lying on the boundary are given a support weight which is equal to the support weight of the reference pixel p_c , as shown in Fig. 7. By doing so, the amplitude of the window costs at pixel p_c can become larger, and the matching result can be more reliable. Furthermore, from our observations of those pixels in the texture regions, with the exception of pixels having small sums of the reference weighted windows; our above assumption is still not violated. Therefore, in the IASW method, the weight-changing process is performed for all pixels except the pixels having the small sums.

0.02	0.09	0.02	0.08	0.07	0.02	0.09	0.02	0.08	0.07
0.24	0.63	0.88	0.65	0.32	1.0	1.0	1.0	1.0	1.0
0.21	0.65	1.0	0.52	0.21	1.0	0.65	1.0	0.52	1.0
0.26	0.60	0.92	0.51	0.26	1.0	0.60	0.92	0.51	1.0
0.24	0.32	0.32	0.19	0.24	1.0	1.0	1.0	1.0	1.0
(a)					(b)				

Fig. 7. An example of a weight-changing process is performed for a reference weighted window with $w=0.15$. The positions of the weight value of 1.0 in (a) and (b) are the positions of the reference pixel. (a) weight values which are in bold are greater than λ_w , and grouped into a area. (b) positions lying on the boundary of the λ_w area are given a weight of 1.0.

The procedure for our improved adaptive support weight method is as follows:

Algorithm 1. Improve Adaptive Support Weight Algorithm

```

1: For each pixel  $S_{w_r}(p)$  in the reference image
2: Compute  $S_{w_r}(p)$ 
3: if  $S_{w_r}(p) \leq th$  then
4: Find disparity of pixel  $p$  by SWVW method
5: else
6: Weight-changing process for reference weighted window
7: Find disparity of pixel  $p$  by ASW method with the
   above processed reference weighted window
8: end if
9: Set disparity of pixel  $p$  to disparity map

```

In the ISWW method, for each pixel p used as a reference pixel in the support window of a reference image, we first compute the sum of the reference weighted window at pixel p . If the sum is less than or equal a threshold th , the SWVW method is applied to find the disparity of pixel p . On the other hand, if the sum is greater than a threshold th , the reference weighted window at pixel p is performed by the weight-changing process, and the IASW method is used to find the disparity of pixel p using the updated reference weighted window.

IV. EXPERIMENT

In this section we report the experimental result of the SWVW method and the ISWW method using Middlebury stereo images [11]. For all experiments, we fixed the parameters of the SWVW method as follows: $\lambda = 25$, $\alpha = 0.7$, $\beta = 18$, $\gamma = -2.0$ with a window size range from (5×5) to (35×35) . For the IASW method, we set $T = 45$, $th = 10$, $\lambda_w = 1.5$, $\gamma_c = 4$, $\gamma_c = 25$ and a support window size $= (35 \times 35)$. We found the above optimal parameter values for our approaches empirically.

We tested the performance of our proposed algorithms using images with ground truth, and then compared the performance of the proposed method with those of three other local methods: shiftable window [6], multiple window [7], variable window [8], and adaptive support weight (ASW) [9]. In order to make the testing fair, we set the parameters for the three above methods to the values described in the original papers, and we do not use any post processing technique for any of the methods tested, including our proposed method

A. Spatial Weight Variable Window Method

Fig. 8 depicts the advantage of the SWVW algorithm when working in depth discontinuity regions. In all three cases, the support windows do not fit the object ‘‘lamp’’. This means that some pixels in the support windows have different disparities from the disparity of the reference pixel. In these cases, the SWVW algorithm can work better than the variable window method, mainly because the SWVW algorithm employs the spatial support weight for each pixel, which is smaller for pixels that are far from the reference pixel. As seen in Fig. 8, the spatially weighted window for each support window can

change due to the position of the reference pixel in the support window.

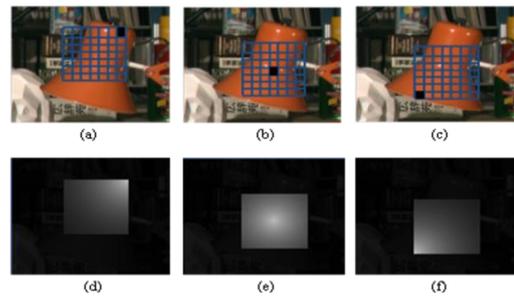


Fig. 8. Spatially weighted windows in the SWVW method. (a), (b), and (c) show examples where the support windows do not fit the lamp and their spatially weighted windows, where the brighter pixels having larger weights, are shown in (d), (e), and (f), respectively.

B. Improved Adaptive Support Weight Method

Fig. 9 depicts pixels in the reference image where the sums of the reference weighted windows are smaller than the threshold th of the Tsukuba image. Almost all of the pixels lie on object boundaries, and the colors of these boundary pixels are mixture of the colors of their neighboring pixels.



Fig. 9. Image of pixels having reference weighted window sums that are less than a threshold $th = 10$.

Table I presents the total number of pixels having reference weighted window sums smaller than the threshold th . When $th = 10.0$, the corresponding total number of pixels are 3016. Table II summarizes the performance of the test methods, except for the IASW method, for the above pixels. Each number in Table II represents the proportion of mismatching pixels in Tsukuba, Venus, Teddy, and Cones images when compared to the results of the test methods in these pixels with the corresponding ground truth images. As we can see, in the case where $th = 10.0$, SWVW method showed the best performance, with an average error percentage of mismatching pixels of 57.685, while the ASW method had the worst performance with the average error proportion of 63.8025.

TABLE I: TOTAL NUMBER OF EXCEL

	Tsukuba	Venus	Teddy	Cones
Total Pixel ($th=10.0$)	3016	1951	2845	2347

Fig. 10 shows two examples of the weight-changing process at pixels in the textureless regions. Table III summarizes the performance of the test stereo methods for the test images. We use the root-mean-square (RMS) error method [6] to compute the percentage of bad matching pixels for all pixels using depth maps and the ground truth. As shown in Table III, the IASW algorithm demonstrates superior

performance among the local test algorithms. Fig. 11 depicts the result of the test methods for the Tsukuba stereo images. Fig. 11b- Fig. 11g show the disparity maps of the test methods for the stereo image pair.

TABLE II: PERCENTAGE OF MISMATCHING PIXELS OUT OF THE TOTAL PIXELS

	SW	MW	VW	SWVW	ASW
Tsukuba (%)	22.94	18.80	17.61	16.98	30.01
Venus (%)	70.73	72.78	70.58	69.91	73.66
Teddy (%)	76.94	76.70	77.29	76.87	79.75
Cones (%)	68.13	68.30	66.94	67.02	71.76
Average Error	59.645	59.145	58.105	57.695	63.803

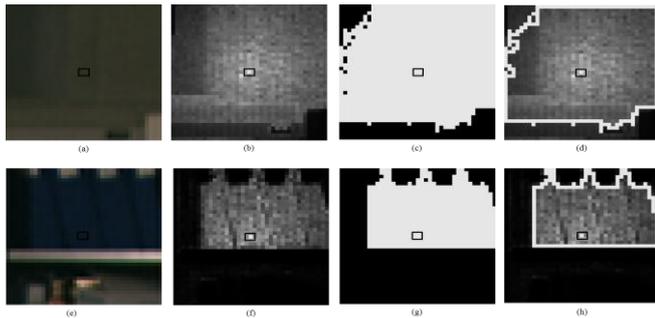


Fig. 10. Two examples of the weight-changing process in the textureless regions with $w = 1.5$. (a) shows the area that the reference support window covers, (b) shows the corresponding reference weighted window of (a), the brighter pixels have more support weights, (c) shows the binary image of (b) with threshold w , and (d) shows the pixels lying on the boundary of the region in (c) that are assigned weights equal to the weight of the reference pixel. Similarly, (e) shows the area that the reference support window covers, (f) shows the corresponding reference weighted window of (e), (g) shows the binary image of (f) with threshold w , and (h) shows pixels lying on the boundary of the region in (g) that are assigned weights equal to the weight of the reference pixel.

TABLE III: PERFORMANCE COMPARISON OF OUR PROPOSED ALGORITHMS WITH OTHER LOCAL TEST ALGORITHMS

	SW	MW	VW	SWVW	ASW	LASW
Tsukuba (RMS)	1.564	1.400	1.363	1.182	1.308	1.072
Venus (RMS)	2.360	2.009	1.723	1.675	1.743	1.621
Teddy (RMS)	8.778	8.215	8.110	7.596	7.176	7.024
Cones (RMS)	9.513	8.346	8.581	8.250	7.339	7.254
Average Error	5.5538	4.9925	4.9443	4.6758	4.3915	4.2428

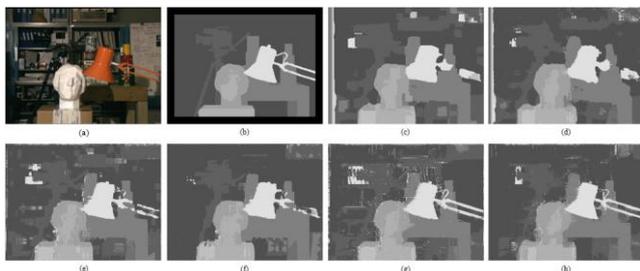


Fig. 11. Results of the test stereo methods on the Tsukuba image pair. (a) shows the reference image. (b)-(g) are the test method disparity maps: (b) shiftable window, (c) multiple window, (d) variable window, (e) spatial weight variable window, (f) adaptive support weight, and (g) improved adaptive support weight. (h) show the bad pixel map of the improved adaptive support weight method (error = 1)

Fig. 11h shows the bad pixel maps of the IASW method computed by comparing the depth map of the IASW method with the Tsukuba ground truth image.

V. CONCLUSION

In this paper, we presented an enhanced blocking-based matching method for stereo matching: the improved adaptive support weight method. The advantages of the proposed method is that they both work robustly and overcome the weaknesses of the original methods. We also analyzed the variable window algorithm and the adaptive support weight algorithm to demonstrate cases where they may not be ideal and suggesting solutions for those cases. We did not use any post-processing techniques for any of the test methods in our experiment, and the experimental results show that our algorithms outperform the original methods.

REFERENCES

- [1] B. Cygane and J. P. Siebert, "An introduction to 3-D computer vision techniques and algorithms," *James Clerk Maxwell, a Treatise on Electricity and Magnetism*, 3rd ed., New York: Wiley-Blackwell, 2009, vol. 2, pp. 68-73.
- [2] Y. S. Heo, K. M. Lee, and S. U. Lee, "Robust stereo matching using adaptive normalized cross correlation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 33, no. 4, pp. 807-822, 2011.
- [3] R. Zabih and J. Woodfill, "Non-parametric local transforms for computing visual correspondence," in *Proc. European Conf. on Computer Vision*, 1994, pp. 151-158.
- [4] G. Egna, "Mutual information as a stereo correspondence measure," Technical Report MS-CIS-00-20, Computer and Information Science, University of Pennsylvania, 2000.
- [5] V. Q. Dinh, D. D. Nguyen, V. D. Nguyen, and J. W. Jeon, "Local stereo matching using an variable window, census transform and an edge-preserving filter," in *Proc. the International Conference on Control, Automation and Systems*, 2012, pp. 523-528.
- [6] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. Journal of Computer Vision*, vol. 47, pp. 7-42, 2002.
- [7] H. Hirschmuller, P. Innocent, and J. Garibaldi, "Real-time correlation based stereo vision with reduced border errors," *Int. Journal of Computer Vision*, vol. 47, pp. 229-246, 2002.
- [8] O. Veksler, "Fast variable window for stereo correspondence using integral images," in *Proc. the International Conference on Computer Vision and Pattern Recognition*, 2003, pp. 556-561.
- [9] K. J. Yoon and I. S. Kweon, "Locally adaptive support-weight approach for visual correspondence search," in *Proc. the International Conference on Computer Vision and Pattern Recognition*, 2005, pp. 924-931.
- [10] A. Hosni, M. Bleyer, M. Gelautz, and C. Rhemann, "Local stereo matching using geodesic support weights," in *Proc. the 16th IEEE International Conference on Image Processing*, Nov. 7-11, 2009, pp. 2093-2096.
- [11] P. Fua, "A parallel stereo algorithm that produces dense depth. maps and preserves image features," *Machine Vision and Applications*, vol. 6, pp. 35-49, 1993.
- [12] Q. X. Yang, R. G. Yang, J. Davis, and D. Nist, "Spatial depth super resolution for range images," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.
- [13] C. Tomasi and R. Manduchi, "Bilateral Filtering for gray and color images," in *Proc. International Conference on Computer Vision*, 1998, pp. 839-846.
- [14] H. P. Nguyen, T. D. Tran, and V. D. Quang, "Enhancement of local methods by using spatial support weight," in *Proc. IEEE International Symposium on Signal Processing and Information Technology*, 2012, pp. 245-250.
- [15] S. Birchfield and C. Tomasi, "A pixel dissimilarity measure that is insensitive to image sampling," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 4, pp. 401-406, 1998.



Phuc Nguyen Hong received the B.S in science service from Auckland University of Technology (AUT), New Zealand in 2013. She is currently working toward the master degree at the Computer Engineering Department, Sungkyunkwan University, Suwon, Republic of Korea.

Her research interests include genetic algorithms and computer vision.



Chang Wook Ahn received Ph.D. degree at the Department of Information and Communications, Kwang-Ju Institute of Science and Technology, Kwang-ju, Republic of Korea.

From 2007 to 2008, he was a research professor in Gwangju Institute of Science & Technology (GIST). He is currently an assistant professor in Sungkyunkwan University (SKKU), Republic of Korea.

His research interests include genetic algorithms, multiobjective optimization, neural networks, and the applications of evolutionary techniques to wireless networks.