Adaptive Support-Weight Approach for Correspondence Search

Kuk-Jin Yoon, *Student Member*, *IEEE*, and
In So Kweon, *Member*, *IEEE*

Abstract—We present a new window-based method for correspondence search using varying support-weights. We adjust the support-weights of the pixels in a given support window based on color similarity and geometric proximity to reduce the image ambiguity. Our method outperforms other local methods on standard stereo benchmarks.

Index Terms—Stereo, 3D/stereo scene analysis.

1 Introduction

The crux of correspondence search is image ambiguity, which results from the ambiguous local appearances of image pixels due to image noise and insufficient (or repetitive) texture. When the local structures of image pixels are similar, it may be very difficult to find their correspondences in other images without global reasoning. However, most global correspondence methods are computationally expensive and sometimes need many parameters that are hard to determine. To properly deal with the image ambiguity problem, area-based local methods generally use some kind of statistical correlation between color or intensity patterns in local support windows. By using the local support windows, the image ambiguity is reduced efficiently while the discriminative power of the similarity measure is increased. In this approach, it is implicitly assumed that all pixels in a support window are from similar depth in a scene and, therefore, that they have similar disparities. Accordingly, pixels in homogeneous regions get assigned the disparities inferred from the disparities of neighboring pixels. However, the support windows located on depth discontinuities represent pixels from different depths, and this results in the "foreground-fattening" phenomenon. Therefore, to obtain accurate results not only at depth discontinuities but also in homogeneous regions, an appropriate support window should be selected for each pixel adaptively. To this end, many methods have been proposed. They can be roughly divided into several categories according to their techniques.

Adaptive-window methods [1], [2], [3], [4] try to find an optimal support window for each pixel. Kanade and Okutomi [1] presented a method to select an appropriate window by evaluating the local variation of intensity and disparity. This method is, however, highly dependent on the initial disparity estimation and is computationally expensive. Moreover, the shape of a support window is constrained to a rectangle, which is not appropriate for pixels near arbitrarily shaped depth discontinuities. On the other hand, Boykov et al. [2] tried to choose an arbitrarily shaped connected window. They performed plausibility hypothesis testing and computed a correct window for each pixel. Veksler [3], [4] found a useful range of window sizes and shapes to explore while evaluating the window cost, which works well for comparing windows of different sizes. However, the shapes of support windows used are not general and this method needs many user-specified parameters for the window cost computation.

Multiple-window methods [5], [6], [7] select an optimal support window among predefined multiple windows, which are located at different positions with the same shape. Fusiello et al. [5] performed the correlation with nine different windows for each pixel and retained the disparity with the smallest matching cost. Kang et al. [7] also presented a multiple-window method that examines all windows containing the pixel of interest.

Although the methods mentioned above improve the performance of correspondence search, they have a limitation in common: The shape of a local support window is not general. In fact, finding the optimal support window with an arbitrary shape and size is very difficult. For this reason, the methods limit their search space by constraining the shape of a support window. Rectangular and constrained-shaped windows, however, may be inappropriate for pixels near arbitrarily shaped depth discontinuities. To resolve this problem, segmentation-based methods [8], [9] use segmented regions with arbitrary sizes and shapes as support windows. In this approach, it is also implicitly assumed that the disparity varies smoothly in each region. However, these methods require precise color segmentation that is very difficult when dealing with highly textured images.

Some methods [10], [11], [12] try to assign appropriate support-weights to the pixels in a support window while fixing the shape and size of a local support window. Prazdny [10] proposed a new function to assign support-weights to neighboring pixels iteratively. In this method, it is assumed that neighboring disparities, if corresponding to the same object in a scene, are similar and that two neighboring pixels with similar disparities support each other. Xu et al. [12] also presented an algorithm that determines adaptive support-weights by radial computations. They used the certainties of the initial disparity distribution to determine support-weights. These methods, however, are dependent on the initial disparity estimation, which may be erroneous.

In this paper, we propose a new correspondence search method to get accurate results at depth discontinuities as well as in homogeneous regions. We compute the support-weights of the pixels in a given support window using color similarity and geometric proximity. The proposed method is composed of three parts: adaptive support-weight computation, dissimilarity computation based on the support-weights, and disparity selection. We give a detailed explanation for each part in Sections 2, 3, and 4 show some experimental results in Section 5. We then discuss the proposed method in Section 6 and conclude the paper in Section 7.

2 Support Aggregation in the Human Visual System

When aggregating support to measure the similarity between image pixels, the support from a neighboring pixel is valid only when the neighboring pixel is from the same depth—it has the same disparity—with the pixel under consideration. However, we do not know the disparities of the pixels beforehand because the disparities are what we want to compute. For this reason, some

• *The authors are with the Department of Electrical Engineering and Computer Science, KAIST, 373-1, Guseong-dong, Yuseong-gu, Daejeon, Korea. E-mail: kjyoon@rcv.kaist.ac.kr, iskweon@ee.kaist.ac.kr.*

Manuscript received 4 Apr. 2005; revised 7 July 2005; accepted 23 Aug. 2005; published online 14 Feb. 2006.
Recommended for acceptance by M. Pollefeys.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0178-0405.

困境

methods [1], [10], [12] iteratively update support windows or support-weights. The iterative methods, however, are very sensitive to the initial disparity estimation and are computationally expensive. To resolve this dilemma, we observed the mechanism performed in the human visual system for correspondence search. In fact, the proposed method originated from the observation that the pixels in a support window are not equally important in the support aggregation step in the human visual system.

2.1 Gestalt Grouping

Visual grouping is very important to form a support window and to compute support-weights and, therefore, the gestalt principles can be used to compute support-weights. There are many visual cues used for perceptual grouping [13], [14]. Among them, similarity and proximity are the two main grouping concepts in classic gestalt theory. The gestalt rule of organization based on similarity (or smoothness) and proximity is one of the most important ones and has been widely used in vision research [15], [16], [17].

相似性 邻近性

The gestalt principles of similarity and proximity are also used to compute support-weights. We compute the support-weight of a pixel based on the strength of grouping by similarity and proximity—the support-weight is in proportion to the strength of grouping. The more similar the color of a pixel, the larger its support-weight. In addition, the closer the pixel is, the larger the support-weight is. The former is related to the grouping by similarity and the latter is related to the grouping by proximity. Although these two rules are usually stated separately, they must be treated as a single rule in an integrated manner to get reasonable grouping.

2.2 Support-Weight Based on the Gestalt Grouping

Based on the gestalt principles, the support-weight of a pixel can be written as

$$w(p,q) = f(\Delta c_{pq}, \Delta g_{pq}), \tag{1}$$

where Δc_{pq} and Δg_{pq} represent the color difference and the spatial distance between pixel p and q, respectively. $f(\Delta c_{pq}, \Delta g_{pq})$ represents the strength of grouping by similarity and proximity when Δc_{pq} and Δg_{pq} are given. Here, Δc_{pq} and Δg_{pq} can be regarded as independent events and the strength of grouping by similarity and proximity can be measured separately. Then, $f(\Delta c_{pq}, \Delta g_{pq})$ can be expressed as

$$f(\Delta c_{pq}, \Delta g_{pq}) = f_s(\Delta c_{pq}) \cdot f_p(\Delta g_{pq}), \tag{2}$$

where $f_s(\Delta c_{pq})$ and $f_p(\Delta g_{pq})$ represent the strength of grouping by similarity and proximity, respectively.

As shown in (2), the core of the support-weight computation is how to model the strength of grouping by color similarity, $f_s(\Delta c_{pq})$, and the strength of grouping by proximity, $f_p(\Delta g_{pq})$. These should be modeled based on the perceptual difference measures.

3 Locally Adaptive Support-Weight Computation

3.1 Strength of Grouping by Similarity

The difference between pixel colors is measured in the CIELab color space because it provides three-dimensional representation for the perception of color stimuli. As the distance between two points in the CIELab color space increases, it is reasonable to assume that the perceived color difference between the stimuli that the points represent increases accordingly. Especially, short Euclidean distances correlate strongly with human color discrimination performance. When Δc_{pq} represents the Euclidean distance between two colors, $\mathbf{c}_p = [L_p, a_p, b_p]$ and $\mathbf{c}_q = [L_q, a_q, b_q]$, in the CIELab color space, the perceptual difference between two colors is expressed as

$$D(\mathbf{c}_p, \mathbf{c}_q) = 1 - \exp\left(-\frac{\Delta c_{pq}}{\gamma}\right), \tag{3}$$

where γ is 14. Based on (3), the strength of grouping by color similarity is defined as

$$f_s(\Delta c_{pq}) = \exp\left(-\frac{\Delta c_{pq}}{\gamma_c}\right). \tag{4}$$

3.2 Strength of Grouping by Proximity

The strength of grouping by proximity is defined in the same manner. According to the gestalt principle of proximity, the support-weight of a pixel decreases as the spatial distance to the reference pixel increases. Here, as in the color difference, only small spatial distances strongly correlate with the human discrimination performance. Therefore, the strength of grouping by proximity is defined using the Laplacian kernel as

$$f_p(\Delta g_{pq}) = \exp\left(-\frac{\Delta g_{pq}}{\gamma_p}\right), \tag{5}$$

where Δg_{pq} is the Euclidean distance between p and q in the image domain and γ_p is determined according to the size of the support-window as $\gamma_p \propto$ (window size). In fact, γ_p is related to the field-of-view of the human visual system.

3.3 Support-Weight Based on the Strength of Grouping

According to (4) and (5), (1) can be rewritten as

$$w(p,q) = \exp\left(-\left(\frac{\Delta c_{pq}}{\gamma_c} + \frac{\Delta g_{pq}}{\gamma_p}\right)\right). \tag{6}$$

Here, it is worthy of notice that the proposed method does not depend on the initial disparity estimation at all because the adaptive support-weight computation is entirely based on the contextual information within a given support window.

4 Dissimilarity Computation and Disparity Selection

The dissimilarity (i.e., the matching cost) between pixels is measured by aggregating raw matching costs with the support-weights in both support windows. In this step, unlike existing methods, we take into account the support-weights in both reference and target support windows. When considering only the reference support window, the computed dissimilarity can be erroneous when the target support window has pixels from different depths. To minimize the effect of such pixels, we compute the dissimilarity between pixels by combining the support-weights in both support windows. The combined support-weights favor the points likely to have similar disparities with the centered pixels in both images. The dissimilarity between pixel p and $\bar{p}_d$, $E(p, \bar{p}_d)$, can be expressed as

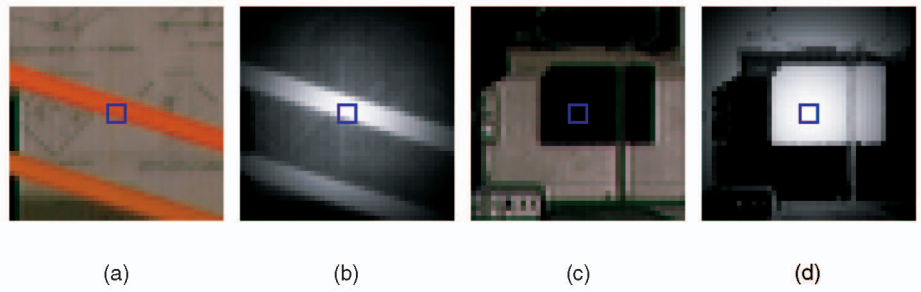


Fig. 1. Support-weight computation (1). The center pixels marked by rectangles are the pixels under consideration. The brighter pixels have larger support-weights in (b) and (d). (a) Reference window. (b) Weights of (a). (c) Target window. (d) Weights of (c).

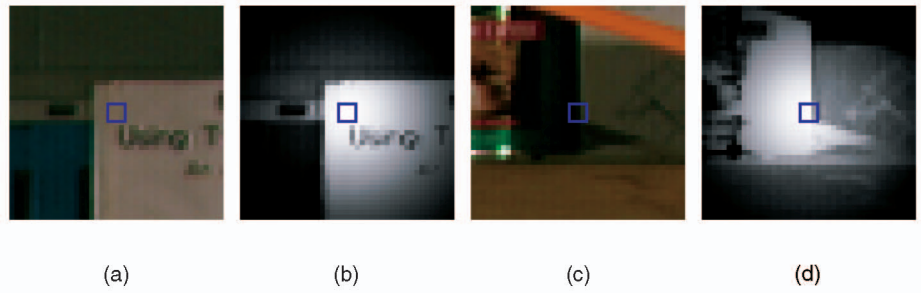


Fig. 2. Support-weight computation (2). The center pixels marked by rectangles are the pixels under consideration. The brighter pixels have larger support-weights in (b) and (d). (a) Reference window. (b) Weights of (a). (c) Target window. (d) Weights of (c).

$$E(p,\bar{p}_d)=\frac{\sum_{q\in N_p,\bar{q}_d\in N_{\bar{p}_d}} w(p,q)w(\bar{p}_d,\bar{q}_d)e(q,\bar{q}_d)}{\sum_{q\in N_p,\bar{q}_d\in N_{\bar{p}_d}} w(p,q)w(\bar{p}_d,\bar{q}_d)}, \tag{7}$$

where $\bar{p}_d$ and $\bar{q}_d$ are the corresponding pixels in the target image when the pixel p and q in the reference image have a disparity value d. $e(q,\bar{q}_d)$ represents the pixel-based raw matching cost computed by using the colors of q and $\bar{q}_d$. When using the truncated AD (absolute difference), it can be expressed as

$$e(q,\bar{q}_d)=\min\left\{\sum_{c\in\{r,g,b\}}|I_c(q)-I_c(\bar{q}_d)|,T\right\}, \tag{8}$$

where I_c is the intensity of the color band c and T is the truncation value that controls the limit of the matching cost.

After the dissimilarity computation, the disparity of each pixel is simply selected by the WTA (Winner-Takes-All) method without any global reasoning as

$$d_p=\arg\min_{d\in\mathbf{S}_d} E(p,\bar{p}_d), \tag{9}$$

where $\mathbf{S}_d=\{d_{min},\cdots,d_{max}\}$ is the set of all possible disparities.

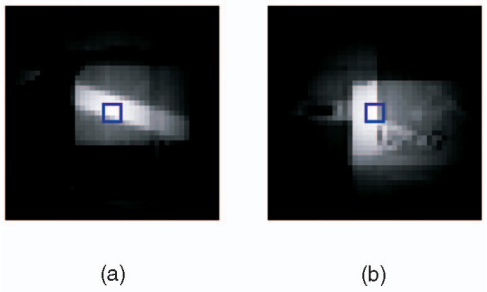


Fig. 3. Combined support-weights used for the similarity computation. (a) Weights for Fig. 1. (b) Weights for Fig. 2.

5 Experiments

5.1 Support-Weight Computation

Fig. 1 and Fig. 2 show the results of support-weight computation for the reference and target support windows. The small rectangles indicate the pixels under consideration. The support-weights in each support window are computed independently and combined as shown in Fig. 3. We can see that the local structures of support windows are reflected in the combined support-weights. The similarity between pixels is then computed by (7) using the combined support-weights.

5.2 Correspondence Search for Synthetic and Real Images

We evaluated the performance of the proposed method using the images with ground truth. We then compared the performance of the proposed method with those of other area-based local methods [3], [4], [7], [19], [20] that perform well.

Fig. 4 shows the result for a synthetic image. The proposed method produces an accurate disparity map, as shown in Fig. 4f. In particular, depth discontinuities are preserved very well. On the other hand, other methods fail to preserve depth discontinuities.

More matching results for real images, which are often used for the performance comparison of various methods [18], are shown in Fig. 5. The proposed method is run with a constant parameter setting across all four images: $T=40$, the size of a support window = (35×35), $\gamma_c=5$, and $\gamma_p=17.5$ (radius of the support window). As shown in Fig. 5, the proposed method yields accurate results at the depth discontinuities as well as in the homogeneous regions for the testbed images.

The performance of the proposed method for the testbed images is summarized in Table 1 to compare the performance with others. The numbers in Table 1 represent the percentage of bad pixels (i.e., a pixel whose absolute disparity error is greater than 1) for all pixels, pixels in untextured areas (except for the "Map" image), and pixels near depth discontinuities. Only nonoccluded

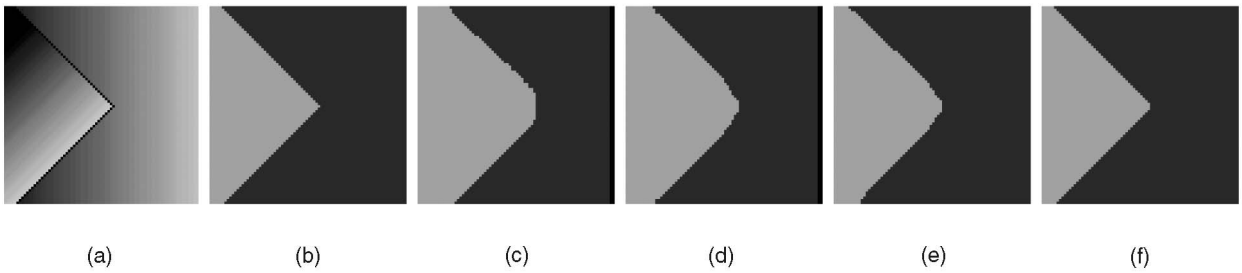
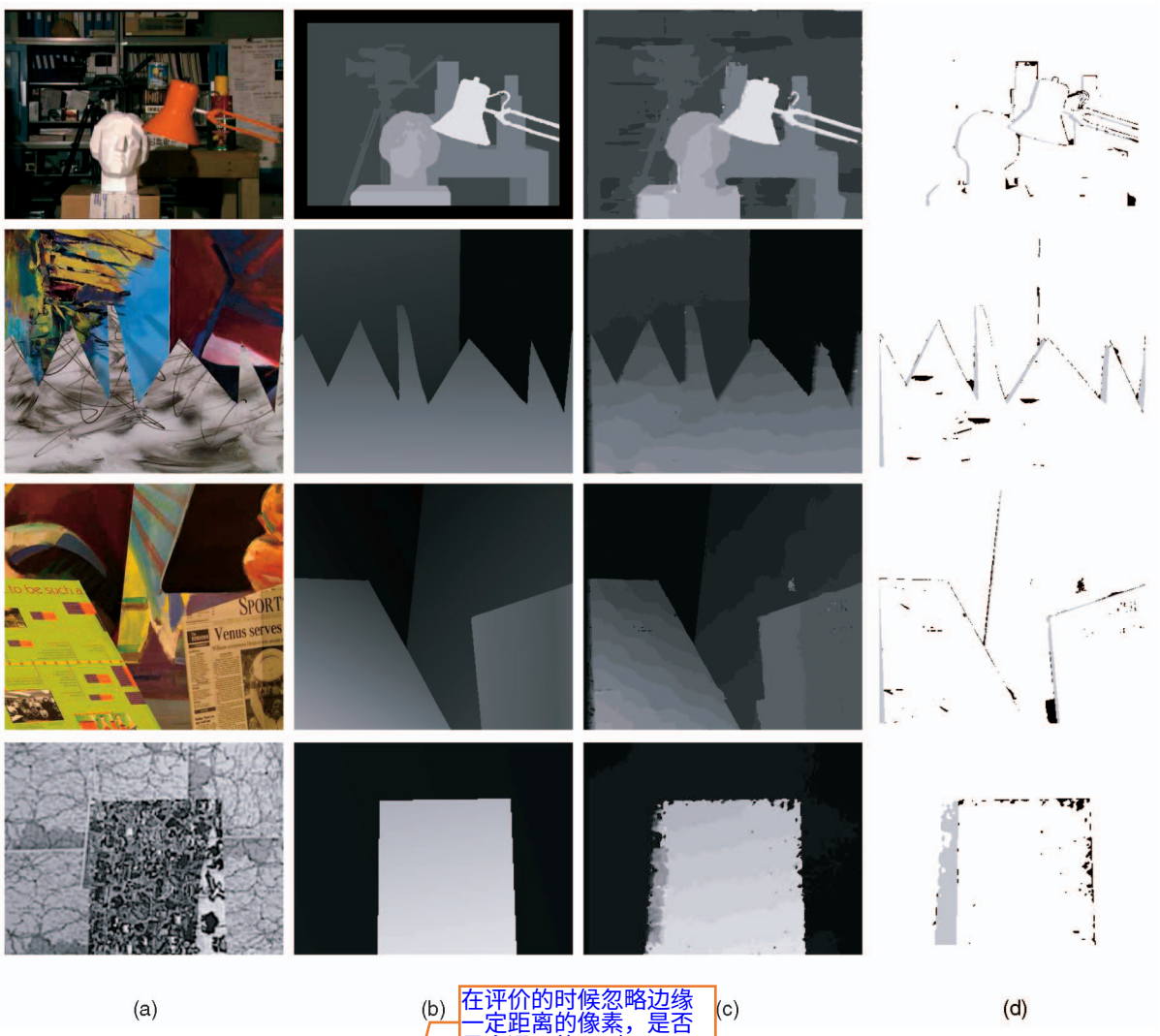


Fig. 4. Dense disparity map for a synthetic image. (a) Left image. (b) Ground truth. (c) SAD. (d) Shiftable window [7]. (e) Bay. Diff. [19]. (f) Proposed.



在评价的时候忽略边缘一定距离的像素，是否是个通用的原则

Fig. 5. Dense disparity maps for the "Tsukuba," "Sawtooth," "Venus," and "Map" images. (a) Left image. (b) Ground truth. (c) Our result. (d) Bad pixels (error >1).

pixels are considered in all three cases and we ignore a border of 10 (18 for the "Tsukuba" image) pixels when computing statistics. As shown in Table 1, the proposed method is generally the best among the state-of-the-art area-based local methods. Particularly, the performance near depth discontinuities is much better than the others because the proposed method can preserve arbitrarily shaped depth discontinuities well, while the methods using rectangular or constrained-shaped windows cannot.

However, the result on the "Map" data set is worse than other local methods. This is because the images are highly textured and there is repetitive texture. When the images are highly textured, the amount of aggregated support for each pixel may be insufficient. In that case, the discriminative power of the proposed similarity measure is reduced, which results in false matches. Moreover, the proposed method may produce inaccurate results for the regions with repetitive textures because the proposed method simply selects disparities using the WTA method locally.

More results for the synthetic and real images are shown in Fig. 6, Fig. 7, and Fig. 8. For these images, we performed the left-right consistency check to correct the disparities in the half-occluded regions and the borders of images. Because the proposed method uses the WTA algorithm in the disparity selection stage, the left-right consistency check can be simply achieved. We can see that, although there are some errors due specular highlights in the disparity map for

猜测是只在一张图片中拍摄到的区域

镜面高光

TABLE 1
Performance Comparison of the Proposed Method

	Tsukuba			Sawtooth			Venus			Map	
	nonocc.	untex.	disc.	nonocc.	untex.	disc.	nonocc.	untex.	disc.	nonocc.	disc.
Adapt. Weights	**1.29**	**0.61**	**6.72**	**0.97**	**0.34**	**4.82**	**0.99**	**0.89**	**6.66**	**1.13**	**11.55**
variable win. [4]	2.35	1.65	12.17	1.28	0.23	7.09	1.23	1.16	13.35	0.24	2.98
compact win. [3]	3.36	3.54	12.91	1.61	0.45	7.87	1.67	2.18	13.24	0.33	3.94
cooperative [20]	3.49	3.65	14.77	2.03	2.29	13.41	2.57	3.52	26.38	0.22	2.37
Bay. diff. [19]	6.49	11.62	12.29	1.45	0.72	9.29	4.00	7.21	18.39	0.20	2.49
shiftable win. [7]	5.23	3.80	24.66	2.21	0.72	13.97	3.74	6.82	12.94	0.66	9.35

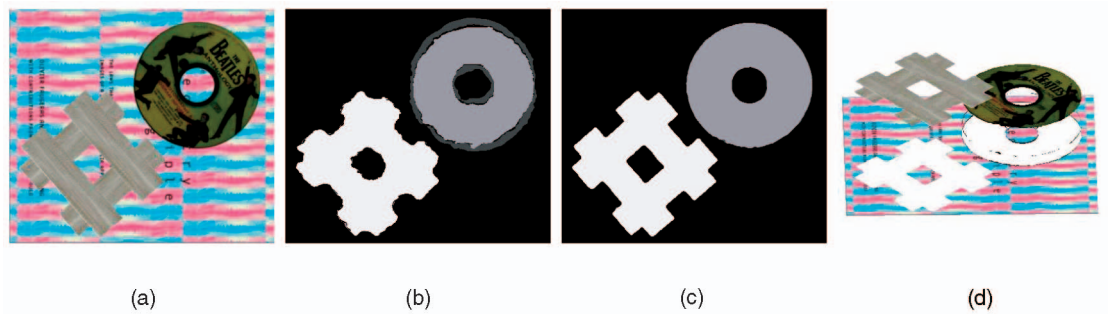


Fig. 6. Dense disparity map for the "CD-Book" image. (a) Left image. (b) Shiftable window [7]. (c) Our result. (d) Novel view synthesized by using (c).

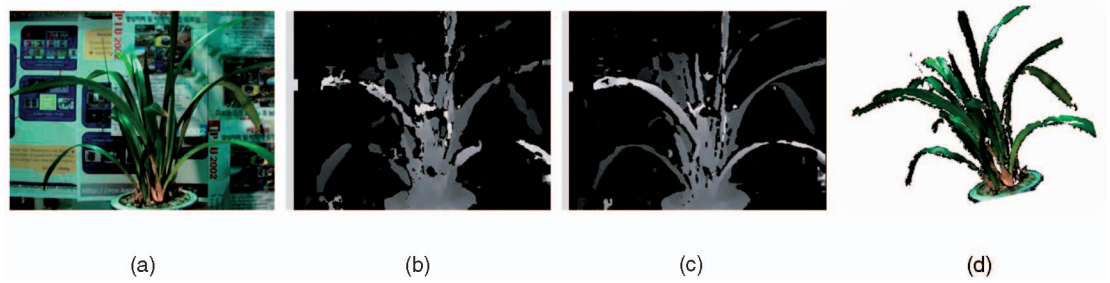


Fig. 7. Dense disparity map for the "Weed" image. (a) Left image. (b) Shiftable window [7]. (c) Our result. (d) Novel view synthesized by using (c).

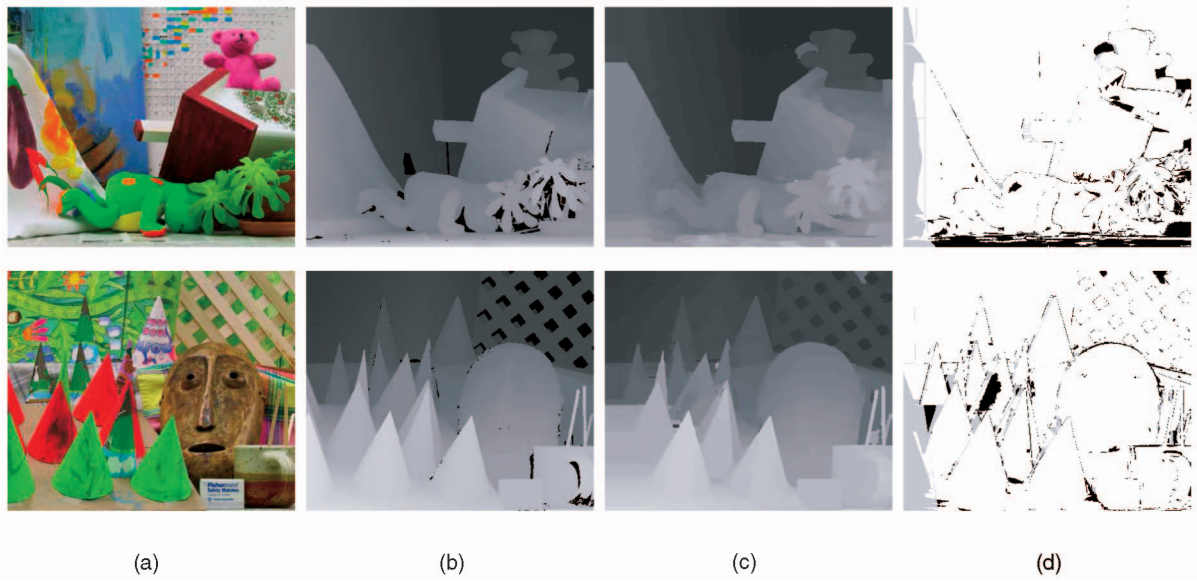


Fig. 8. Dense disparity maps for the "Teddy" and "Cone" images. (a) Left image. (b) Ground truth. (c) Our result. (c) Bad pixels (error > 1).

TABLE 2
Performance of the Proposed Method for the New Testbed Images

	Tsukuba			Venus			Teddy			Cone		
	nonocc.	all	disc.	nonocc.	all	disc.	nonocc.	all	disc.	nonocc.	all	disc.
Segm+visib [21]	1.30	1.57	6.92	0.79	1.06	6.76	5.00	6.54	12.3	3.72	8.62	10.2
SymBP+occ [22]	0.97	1.75	5.09	0.16	0.33	2.19	6.47	10.7	17.0	4.79	10.7	10.9
Adapt. Weights	**1.38**	**1.85**	**6.90**	**0.71**	**1.19**	**6.13**	**7.88**	**13.3**	**18.6**	**3.97**	**9.79**	**8.26**
SemiGlob [23]	3.26	3.96	12.8	1.00	1.57	11.3	6.02	12.2	16.3	3.06	9.75	8.9

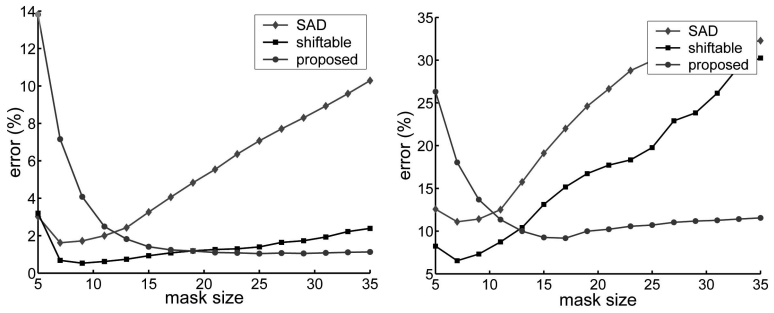


Fig. 9. Performance according to the window size for the "Map" image. Note that γ_p in (6) is increased according to the size of the support window for fair comparison. (a) Nonoccluded. (b) Depth discontinuities.

自适应权重计算公式和一些保留结构的噪声过滤方法中的自适应权重计算公式类似

the "Weed" image, the proposed method produces very accurate disparity maps and preserves arbitrary shaped depth discontinuities well. Table 2 shows the performance of the proposed method with the consistency check for the new testbed images that can be found at http://www.middlebury.edu/stereo.[1] The performance of the proposed method is comparable to the global methods.

6 Discussion

6.1 Sensitivity to the Size of a Support Window and Parameter Values

Fig. 9 shows the performance of the proposed method, the SAD (sum of absolute difference), and the shiftable window method [7] for the "Map" image, which is textured almost everywhere, according to the size of a support window. In this case, we increased γ_p in (6) according to the size of the support window for fair comparison. We can see that the proposed method is fairly robust against different sizes of a support window, whereas the others are not. This is because the effect of outliers (i.e., pixels from different depths) does not increase in the proposed method even though the size of a support window increases.

Fig. 10 shows the performance of the proposed method for the "Sawtooth" and "Venus" images according to γ_c. In this case, we kept the size of the support window and γ_p constant. Our method also appears to be fairly robust against different values of γ_c—the accuracy is almost constant for γ_c between 4.0 and 7.0. Although the determination of parameter values seems to be difficult, the parameter values are not so critical in the proposed method because the effect of outliers is suppressed by using the combined form of support-weights.

1. Results for the "Tsukuba" and "Venus" images with the consistency check also can be found at http://www.middlebury.edu/stereo.

6.2 Connection with Structure-Preserving Noise Filters

Equation (6) used for support-weight computation is very similar to the functions used for computing adaptive weights in structure-preserving noise filters such as those in [24], [25. This is because the structure-preserving filters also use the adaptive weights based on the intensity similarity and distance between pixels to remove image noise. As a result of using adaptive support-weights, the structure-preserving filters produce noise-removed smooth images while preserving the image structures (i.e., the intensity edges) well; the proposed method produces smooth disparity maps while preserving depth discontinuities well. In fact, the adaptive support-weight approach is applicable to any applications aimed at getting discontinuity-preserving smooth results.

7 Conclusions

In this paper, we have proposed a new area-based local method for correspondence search that focuses on the dissimilarity computation. Instead of finding an optimal support window, we adjusted

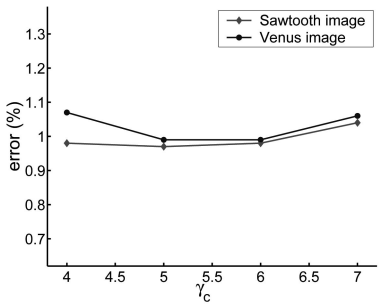


Fig. 10. Performance according to the γ_c for the "Sawtooth" and "Venus" images while the sizes of the support window and γ_p are kept constant.

the support-weight of the pixel in a given support window based on the color similarity and geometric distance to the reference pixel. We then computed the dissimilarity between pixels using the support-weights in both support windows. Experimental results show that the proposed method produces accurate piecewise smooth disparity maps. Particularly, the performance near the depth discontinuities is much better than that of other methods because the proposed method can preserve arbitrarily shaped depth discontinuities well, whereas the methods using rectangular or constrained-shaped windows cannot.

The proposed method has some advantages. First, the proposed method does not depend on the initial disparity estimation because the adaptive support-weight is noniteratively computed based on the contextual information within a given support window. Second, the proposed method is fairly robust against different sized support windows. The proposed method, however, is computationally a little more expensive than other area-based local methods because of the pixel-wise adaptive support-weight computation. For instance, the running time for the "Tsukuba" image with a (35×35) support window is about one minute on the AMD 2700+ machine. Fortunately, however, the runtime of the proposed method can be reduced by using parallel processors because the support-weights can be computed in parallel.

ACKNOWLEDGMENTS

This research was supported by the Korean Ministry of Science and Technology for the National Research Laboratory Program (grant number M1-0302-00-0064).

REFERENCES

[1] T. Kanade and M. Okutomi, "A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiments," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 16, no. 9, pp. 920-932, Sept. 1994.
[2] Y. Boykov, O. Veksler, and R. Zabih, "A Variable Window Approach to Early Vision," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 12, pp. 1283-1294, Dec. 1998.
[3] O. Veksler, "Stereo Correspondence with Compact Windows via Minimum Ratio Cycle," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 12, pp. 1654-1660, Dec. 2002.
[4] O. Veksler, "Fast Variable Window for Stereo Correspondence using Integral Images," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 556-561, 2003.
[5] A. Fusiello, V. Roberto, and E. Trucco, "Efficient Stereo with Multiple Windowing," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 858-863, 1997.
[6] A.F. Bobick and S.S. Intille, "Large Occlusion Stereo," *Int'l J. Computer Vision,* vol. 33, no. 3, pp. 181-200, 1999.
[7] S.B. Kang, R. Szeliski, and C. Jinxjang, "Handling Occlusions in Dense Multi-View Stereo," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 103-110, 2001.
[8] H. Tao, H.S. Sawhney, and R. Kumar, "A Global Matching Framework for Stereo Computation," *Proc. Int'l Conf. Computer Vision,* vol. 1, pp. 532-539, 2001.
[9] L. Wang, S.B. Kang, and H.-Y. Shum, "Cooperative Segmentation and Stereo Using Perspective Space Search," *Proc. Asian Conf. Computer Vision,* vol. 1, pp. 366-371, 2004.
[10] K. Prazdny, "Detection of Binocular Disparities," *Biological Cybernetics,* vol. 52, pp. 93-99, 1985.
[11] T. Darrel, "A Radial Cumulative Similarity Transform for Robust Image Correspondence," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp. 656-662, 1998.
[12] Y. Xu, D. Wang, T. Feng, and H.-Y. Shum, "Stereo Computation using Radial Adaptive Windows," *Proc. Int'l Conf. Pattern Recognition,* vol. 3, pp. 595-598, 2002.
[13] S.E. Palmer, *Vision Science: Photons to Phenomenology.* Bradford Books, MIT Press, 1999.
[14] D. Marr, *Vision.* WH Freeman and Company, 1982.
[15] S.W. Zucker and R.A. Hummel, "Toward a Low-Level Description of Dot Clusters: Labeling Edge, Interior and Noise Points," *Computer Graphics and Image Processing,* vol. 9, pp. 213-233, 1979.
[16] N. Ahuja and M. Tuceryan, "Extraction of Early Perceptual Structure in Dot Patterns: Integrating Region, Boundary, and Component Gestalt," *Computer Vision, Graphics, and Image Processing,* vol. 48, pp. 304-356, 1989.
[17] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 8, pp. 888-905, Aug. 2000.
[18] D. Scharstein and R. Szeliski, "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithm," *Int'l J. Computer Vision,* vol. 47, no. 1, pp. 7-42, 2002.
[19] D. Scharstein and R. Szeliski, "Stereo Matching with Nonlinear Diffusion," *Int'l J. Computer Vision,* vol. 28, no. 2, pp. 155-174, 1998.
[20] C.L. Zitnick and T. Kanade, "A Cooperative Algorithm for Stereo Matching and Occlusion Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 7, pp. 675-684, July 2000.
[21] M. Bleyer and M. Gelautz, "A Layered Stereo Algorithm Using Image Segmentation and Global Visibility Constraints," *Proc. Int'l Conf. Image Processing,* pp. 2997-3000, 2004.
[22] J. Sun, Y. Li, S.B. Kang, and H.-Y. Shum, "Symmetric Stereo Matching for Occlusion Handling," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 399-406, 2005.
[23] H. Hirschmüeller, "Accurate and Efficient Stereo Processing by Semi-Global Matching and Mutual Information," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 2, pp. 807-814, 2005.
[24] C. Tomasi and R. Manduchi, "Bilateral Filtering for Gray and Color Images," *Proc. Int'l Conf. Computer Vision,* pp. 839-846, 1998.
[25] S.M. Smith and J.M. Brady, "SUSAN—A New Approach to Low Level Image Processing," *Int'l J. Computer Vision,* vol. 23, no. 1, pp. 45-78, 1997.