Locally Adaptive Support-Weight Approach for Visual Correspondence Search

Kuk-Jin Yoon and In-So Kweon
Robotics and Computer Vision Laboratory
Dept. of Electrical Engineering and Computer Science, KAIST
373-1 Guseong-dong Yuseong-gu Daejeon, 305-701, Korea
kjyoon@kaist.ac.kr, iskweon@kaist.ac.kr

Abstract

In this paper, we present a new area-based method for visual correspondence search that focuses on the dissimilarity computation. Local and area-based matching methods generally measure the similarity (or dissimilarity) between the image pixels using local support windows. In this approach, an appropriate support window should be selected adaptively for each pixel to make the measure reliable and certain. Finding the optimal support window with an arbitrary shape and size is, however, very difficult and generally known as an NP-hard problem. For this reason,unlike the existing methods that try to find an optimal support window, we adjusted the support-weight of each pixel in a given support window. The adaptive support-weight of a pixel is computed based on the photometric and geometric relationship with the pixel under consideration. Dissimilarity is then computed using the raw matching costs and support-weights of both support windows, and the correspondence is finally selected by the WTA (Winner-Takes-All) method. The experimental results for the rectified real images show that the proposed method successfully produces piecewise smooth disparity maps while preserving sharp depth discontinuities accurately.

1. Introduction

The crux of visual correspondence search is the image ambiguity, which results from the ambiguous local appearances of the image pixels due to image noise and insufficient (or repetitive) texture. When the local structures of the image pixels are similar, it may be very difficult to find their correspondences in other images without any global reasoning.

To properly deal with the image ambiguity problem, local and area-based methods generally use some kind of statistical correlation between color or intensity patterns in the local support windows. By using local support windows, 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, support windows that are located on depth discontinuities represent pixels from different depth, and this may result in the 'foreground-fattening' phenomenon.

To obtain more 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.

to have bord

Adaptive-window methods [1], [2], [3], [4] try to find an optimal support window for each pixel by changing the size and shape of a window adaptively. 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 the 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. In [3] and [4], Veksler 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 the 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 the pre-defined 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 the multiple-window method that examines all windows containing the pixel of interest.

Although the methods mentioned above improve performance, 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 and generally known as an NP-hard problem. 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 the pixels near arbitrarily shaped depth discontinuities. To resolve this problem, segmentation-based methods [8], [9] use segmented regions with arbitrary size and shapes as support windows. In this approach, it is also implicitly assumed that the disparities vary 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 at al. [12] also presented an algorithm that determines the adaptive support-weight by radial computations. They used the certainties of the initial disparity distribution to determine the support-weight. 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 adjusted the support-weights of the pixels in a given window by using the photometric and geometric relationship with the pixel under consideration. Dissimilarity is then computed using the support-weights of the pixels in the support windows, and the correspondence is finally selected by the WTA (Winner-Takes-All) method. Since the adaptive support-weight computation is based on the contextual information within a support window, the proposed method does not depend on the initial disparity estimation at all.

The proposed method is composed of three parts: locally adaptive support-weight computation, dissimilarity computation based on the support-weights, and disparity selection. We give a detailed explanation for each part in Secs. 2–3, and experimental results are shown in Sec. 4. We then discuss the proposed method in Sec. 5 and conclude the paper in Sec. 6.

2. Locally Adaptive Support-Weight Computation

When aggregating support to measure the similarity between image pixels, support from a neighboring pixel is valid only when the neighboring pixel is from the same depth – it has the same disparity – as the pixel under consideration. Therefore, the support-weight should be in proportion to the probability that the pixels have the same disparity. Assuming that images are rectified without loss of generality, this can be expressed as

$$w(p, q) \propto \Pr(d_p = d_q) \tag{1}$$

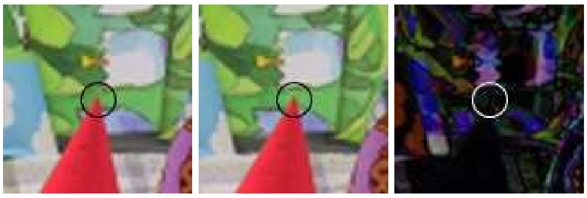
where p is the pixel under consideration and q is the other pixel in the support window of p, N_p. $w(p, q)$ represents the support-weight of the pixel q. d_p and d_q represent the disparities of pixel p and q, respectively, and $\Pr(d_p = d_q)$ is the probability of the event $\{d_p = d_q\}$. It is worthy of notice that the support-weight of a pixel is not dependent on the disparity d and $w(p, q) = w(q, p)$.

The support-weight computation is then specified as the estimation of a probability, $\Pr(d_p = d_q)$ in (1). However, we do not know the disparities of the pixels beforehand because they are what we want to compute. For this reason, some methods [1], [10], [12] iteratively update the 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. Support aggregation in the human visual system

According to the gestalt principles of perception, when perceiving a visual field, some objects (*figures*) seem prominent, and other aspects of field recede into the background (*ground*). This is because similar elements are contrasted with dissimilar elements to give the impression of a whole. This is referred to as figure-ground discrimination and is accomplished by the grouping process [13]. In addition, according to Marr's stages of visual processing [14], stereopsis is performed by using full primal sketches, which are constructed from raw primal sketches by applying various gestalt principles to resolve ambiguities.

From these facts, we can infer that when perceiving depth with binocular disparities, the human visual system makes *figures* by gestalt grouping and uses them as support regions. It is, therefore, reasonable to assume that support



(a) left support window (b) right support window (c) color difference between (a) and (b)

Figure 1. Difference between support windows

aggregation is achieved in the support window formed by gestalt grouping and that the difference in the background does not affect the correspondence search. For example, when trying to find the correspondence of the apex of a red cone in Fig. 1(a), the red cone is used as the support region and the difference in background does not matter at all, although it is very severe when the apex of a red cone is matched as shown in Fig. 1(c).

2.2. Gestalt grouping

As described in Sec. 2.1, 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. Among them, similarity and proximity are the two main grouping concepts in the classic gestalt theory: similarity refers to what items look like and how that affects grouping, and proximity refers to where items are and how that affects grouping. The gestalt rule of organization based on the similarity (or smoothness) and proximity is one of the most important ones and has been widely used in vision research [15], [16], [17].

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 the support-weight of the pixel. In addition, the closer the pixel is, the larger the support-weight. 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.

2.3. Support-weight based on the gestalt grouping

Based on the gestalt principles described above, support-weight can be rewritten as

$$w(p,q) = k \cdot f(\Delta c_{pq}, \Delta g_{pq}) \qquad (2)$$

where k is a proportion constant. Δc_{pq} and Δg_{pq} represent the color difference and the spatial distance between pixel p and q, respectively. The former is the measure of similarity and the latter is the measure of proximity. $f(\Delta c_{pq}, \Delta g_{pq})$ 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

$$f(\Delta c_{pq}, \Delta g_{pq}) = f_s(\Delta c_{pq}) \cdot f_p(\Delta g_{pq}) \qquad (3)$$

where $f_s(\Delta c_{pq})$ and $f_p(\Delta g_{pq})$ represent the strength of grouping by similarity and proximity, respectively. We then compute the adaptive support-weights using the following equation:

$$w(p,q) = k \cdot f_s(\Delta c_{pq}) \cdot f_p(\Delta g_{pq}) \qquad (4)$$

As shown in (4), 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.

2.4. Strength of grouping by similarity

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

$$\Delta c_{pq} = \sqrt{(L_p - L_q)^2 + (a_p - a_q)^2 + (b_p - b_q)^2} \qquad (5)$$

The perceptual difference between two colors is then expressed as

$$D(p,q) = 1 - \exp(-\frac{\Delta c_{pq}}{\gamma}) \qquad (6)$$

where γ is typically 14.

Based on (6), the strength of grouping by color similarity is defined using the Laplacian kernel as

$$f_s(\Delta c_{pq}) = \exp(-\frac{\Delta c_{pq}}{\gamma_c})$$

i.e. as the difference c

Here, we set $\gamma_c = \frac{1}{2}\gamma$ so that $e^{-\frac{\Delta c_{pq}}{\gamma_c}} = (e^{-\frac{\Delta c_{pq}}{\gamma}})^2$.

2.5. 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 pixel under consideration 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

=> exp fa

$$f_p(\Delta g_{pq}) = \exp(-\frac{\Delta g_{pq}}{\gamma_p}) \tag{8}$$

where γ_p is determined empirically.

2.6. Locally adaptive support-weight computation

According to (7) and (8), (4) can be rewritten as

$$w(p,q) = k \cdot \exp\left(-(\frac{\Delta c_{pq}}{\gamma_c} + \frac{\Delta g_{pq}}{\gamma_p})\right) \tag{9}$$

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.

It is also remarkable that the support-weight of q computed by (9) is generally in proportion to the probability that q has the same disparity with p as expressed in (1). In fact, when two pixels have similar colors, it is reasonable to assume that they are from the same smooth surface in a scene as assumed in segmentation-based methods. In addition, the pixel in a support window is generally expected to be from the same smooth surface as the pixels under consideration; this expectation is less certain the farther the pixels in a support window is from the pixel under consideration. Therefore, we can get support-weights proportional to $\Pr(d_p = d_q)$ using (9) approximately.

3. Dissimilarity Computation and Disparity Selection

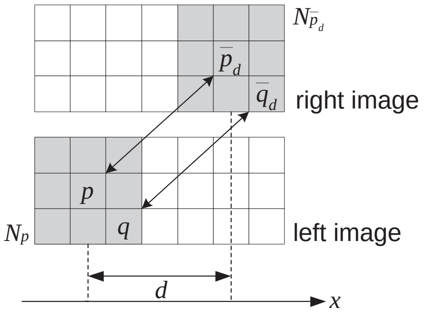


Figure 2. Reference support window (N_p) and target support window ($N_{\bar{p}_d}$) when $d_p = d$

The dissimilarity (i.e. the matching cost) between pixels is then measured by aggregating the raw matching costs with the support-weights in the support windows. In this step, unlike existing methods, we take into account the support-weights of both the reference and target support windows. When only considering the reference support window, the dissimilarity measure may be erroneous because the target support window may have pixels from different depth. To minimize the effect of such pixels, we compute the dissimilarity between pixels by combining the support-weights of both support windows. The combined support-weights encourage the points that are 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

$$E(p,\bar{p}_d) = \frac{\sum_{q \in N_p, \bar{q}_d \in N_{\bar{p}_d}} w(p,q) w(\bar{p}_d, \bar{q}_d) e_0(q, \bar{q}_d)}{\sum_{q \in N_p, \bar{q}_d \in N_{\bar{p}_d}} w(p,q) w(\bar{p}_d, \bar{q}_d)} \tag{10}$$

where $\bar{p}_d$ and $\bar{q}_d$ are the corresponding pixels in the other image when the pixel p and q have disparity d as shown in Fig. 2. $e_0(q, \bar{q}_d)$ represents the pixel-based raw matching cost computed by using the colors of q and $\bar{q}_d$. When using AD (absolute difference), it can be expressed as

$$e_0(q, \bar{q}_d) = \sum_{c \in \{r,g,b\}} |I_c(q) - I_c(\bar{q}_d)| \tag{11}$$

where I_c is the intensity of the color band c.

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

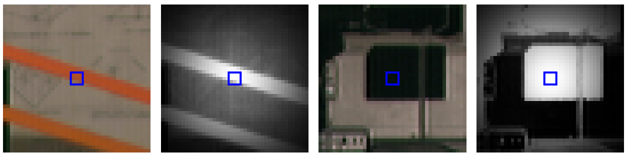
$$d_p = \arg\min_{d \in S_d} E(p, \bar{p}_d) \tag{12}$$

where $S_d = \{d_{min}, \cdots, d_{max}\}$ is a set of all possible disparity values.

4. Experiments

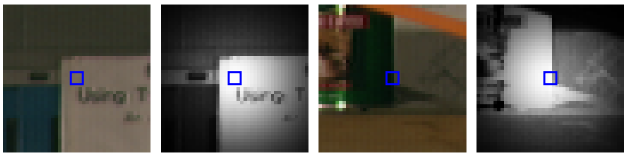
4.1. Support-weight computation

Figures 3–4 show the results of support-weight computation for the reference and target support windows. The small blue rectangle indicates the pixel under consideration. The support-weights for each support window are computed independently and combined as shown in Fig. 5. We can see that the local structures of the support windows are reflected in the combined support-weights. Similarity is then computed by (10) using the combined support-weights.



(a) ref. windows (b) weights of (a) (c) tar. windows (d) weights of (c)

Figure 3. Support-weight computation (1)



(a) ref. window (b) weights of (a) (c) tar. window (d) weights of (c)

Figure 4. Support-weight computation (2)

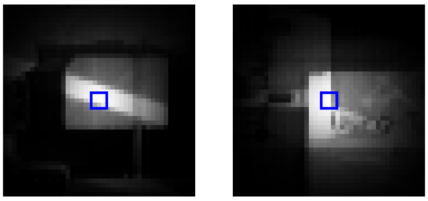
4.2. Correspondence search for the images with ground truth

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 local and area-based methods [3], [4], [7], [19], which perform well among the area-based local methods, as in [18].

Figure 6 shows a result for a synthetic image. The left image and ground truth are shown in Fig. 6(a) and 6(b). The proposed method produces a accurate disparity map as shown in Fig. 6(f). In particular, the depth discontinuities are preserved very well. On the other hand, other methods fail to preserve the depth discontinuities.

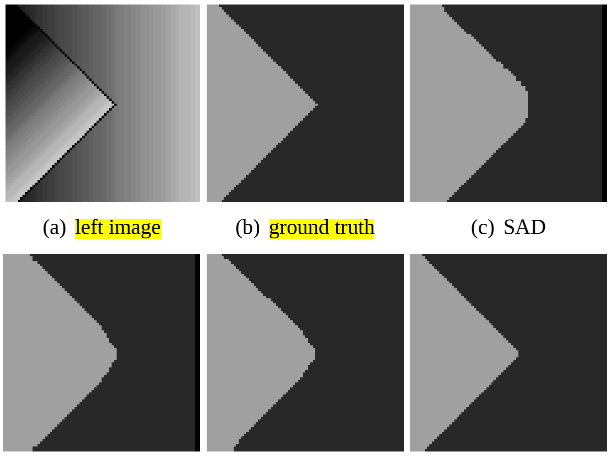
More dense matching results for real images, which are often used for the performance comparison of various methods [18], are shown in Figs. 7–8. The proposed algorithm is run with a constant parameter setting across all four images: the size of a support window = (33 × 33), $\gamma_c = 7$, $\gamma_p = 36$. As shown in Figs. 7–8, 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., 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 non-occluded 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 roughly the best among the state-of-the-art area-based local methods. Particularly, the performance near the 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.



(a) weights for Fig. 3 (b) weights for Fig. 4

Figure 5. Combined support-weights



(a) left image (b) ground truth (c) SAD

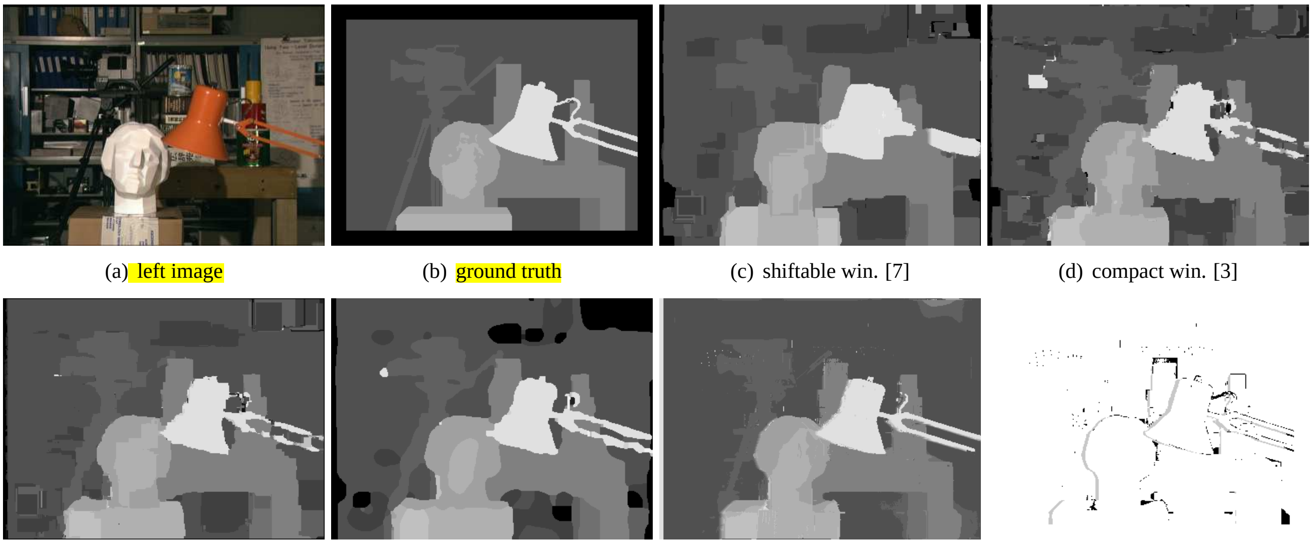
(d) shiftable win. [7] (e) Bay. diff. [19] (f) proposed

Figure 6. Dense disparity map for a synthetic image

5. Discussion

5.1. Sensitivity to the size of a support window

Fig. 9 shows the performance of the proposed method, the SAD (sum of absolute difference) and the SSD (sum of squared difference) based methods, and the shiftable window method [7] for the 'Map' image according to the size of the support window. In this case, we increased the γ_p value in (9) according to the size of the support window for a fair comparison. Fig. 9(a) shows the percentage of error in the non-occluded area and Fig. 9(b) shows the percentage of error at the depth discontinuities. We can see that the proposed method is fairly robust against different sizes of the support window, while others are not. This is because the effect of outliers (i.e., pixels from different depth) does not increase in the proposed method even though the size of the support window increases. We can get similar results for other images.


(a) left image (b) ground truth (c) shiftable win. [7] (d) compact win. [3]

(e) variable win. [4] (f) Bay. diff. [19] (g) our result (h) bad pixels (error > 1)

Figure 7. Dense disparity map for the ‘Tsukuba’ image

Table 1. Performance comparison of the proposed method

image algorithm	Tsukuba			Sawtooth			Venus			Map	
	all	untex.	disc.	all	untex.	disc.	all	untex.	disc.	all	disc.
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
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
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
proposed	**1.51**	**0.65**	**7.25**	**1.15**	**0.29**	**5.47**	**1.19**	**0.72**	**4.49**	**1.42**	**13.40**

5.2. Correspondence search in the presence of specular reflection

Most correspondence search methods assume that the surfaces in a scene are perfect Lambertian, and, therefore, the colors of corresponding pixels are identical across all images. The presence of specular reflection, however, is inevitable in general. While diffuse reflection exhibits little color variation from different viewing positions, specular reflection tends to change significantly in both color and position. Therefore, the colors of the corresponding pixels can be different when one of them has a specular reflection component, and the similarity measure can be erroneous for specular pixels in a support window. Nevertheless, to the best of our knowledge, few local methods can deal with specular reflection in correspondence search properly.

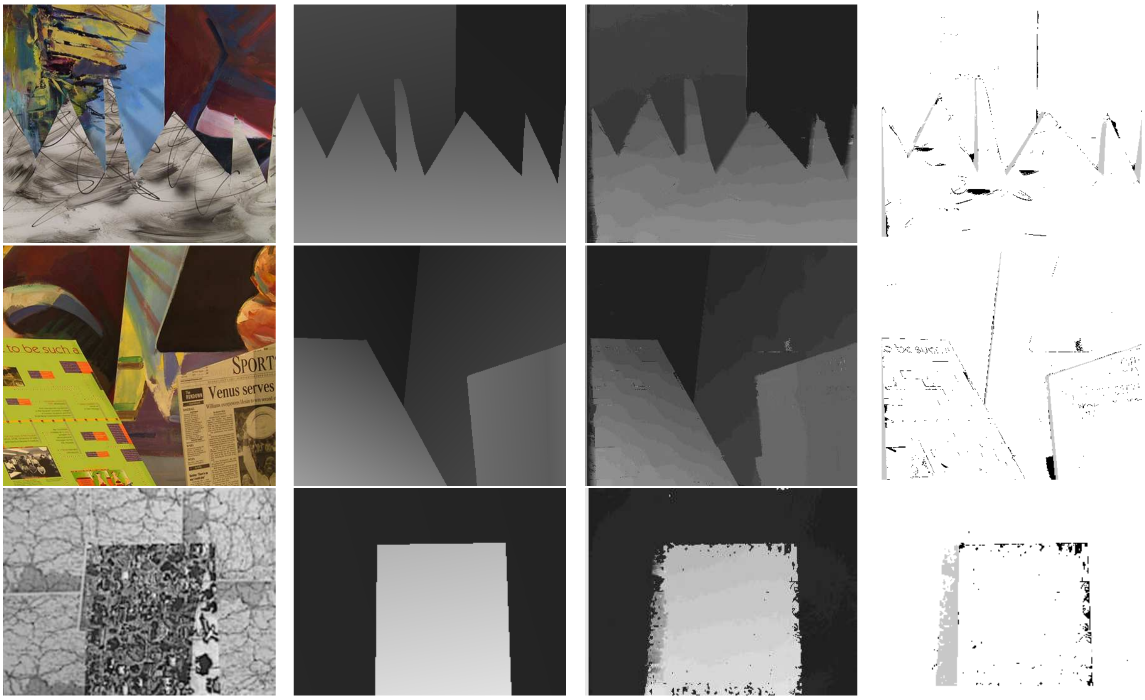
We can effectively handle the specular reflection in correspondence search by using the proposed approach. Specular pixels in a support window should be regarded as outliers in the dissimilarity computation because their raw matching costs are uncertain. Therefore, to minimize the effect of specular pixels in the dissimilarity computation, we assign small support-weights to specular pixels; the larger the specular reflection component of the pixel, the smaller the support-weight. This can be simply expressed as

$$S_{pq} = \exp\left(-\frac{\Sigma s_{pq}}{\gamma_s}\right) \qquad (13)$$

where $\Sigma s_{pq} = (s_p + s_q)$. s_p and s_q are the specularity of pixel p and q, which can be obtained by [20], and S_{pq} represents the certainty of the raw matching costs of p and q based on the specularity. The support-weight is then computed as

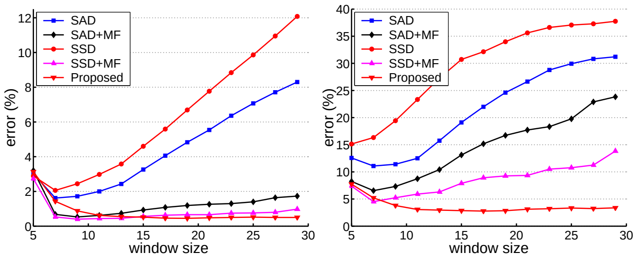
$$\begin{aligned} w(p,q) &= k \cdot S_{pq} \cdot \exp\left(-\left(\frac{\Delta c_{pq}}{\gamma_c} + \frac{\Delta g_{pq}}{\gamma_p}\right)\right) \\ &= k \cdot \exp\left(-\left(\frac{\Delta c_{pq}}{\gamma_c} + \frac{\Delta g_{pq}}{\gamma_p} + \frac{\Sigma s_{pq}}{\gamma_s}\right)\right) \qquad (14) \end{aligned}$$

Figure 10 shows the result of the support-weight computation for a support window with specular reflection. As shown in Fig. 10(a), the pixel of interest has a large specular reflection component. By using the specularities of pixels in Fig. 10(b), specular pixels get assigned smaller support-weights than diffuse pixels as shown in Fig. 10(d).



(a) left image (b) ground truth (c) our result (d) bad pixels (error > 1)

Figure 8. Dense disparity maps for the ‘Sawtooth’, ‘Venus’, and ‘Map’ image

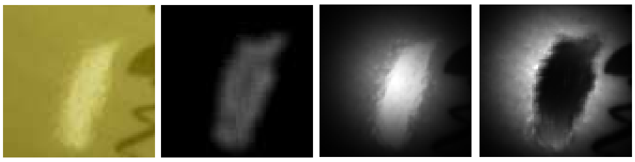


(a) non-occluded (b) depth discontinuities

Figure 9. Performance according to the window size

5.3. Connection with structure-preserving noise filters

It is very interesting that (9) used for support-weight computation in the proposed method is very similar to the functions used for computing adaptive weights in the structure-preserving noise filters such as [21], [22]. This is because the structure-preserving filters also use the adaptive weights based on 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.



(a) support window (b) specularity of (a) (c) weights w/o specular analysis (d) weights with specular analysis

Figure 10. Support-weight for a highlight region

6. Conclusions

In this paper, we proposed a new local area-based method for visual correspondence search that focuses on the dissimilarity computation step. Instead of finding an optimal support window, we adjusted the support-weight of each pixel in a given support window. The adaptive support-weight of the pixel in a support window is computed by measuring the strength of grouping by proximity and similarity. The experimental results show that the proposed method produces accurate piecewise smooth disparity maps. Particularly, the performance near the 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.

The proposed method has some advantages. First, it does not depend on the initial disparity estimation, because the adaptive support-weight is computed noniteratively based on the contextual information within a given support window. Second, the proposed method is fairly robust against different sizes of a support window. In addition, the proposed method can properly deal with the specular reflection in the dissimilarity computation step.

The proposed method, however, is computationally a little more expensive than other area-based methods for the pixel-wise adaptive support-weight computation step. In fact, the typical running time for the window size (33×33) is about one minute on the AMD 2700+ machine. Moreover, although the proposed method works well for real images with little image noise, it may produce an erroneous result when there is severe image noise because the color difference used for the support-weight computation is measured by using an individual pixel color. We are now developing efficient techniques to make the proposed methods faster and more robust against the image noise.

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, 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, 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, 2002.

[4] O. Veksler, “Fast Variable Window for Stereo Correspondence using Integral Images,” in *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,” in *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,” in *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,” in *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,” in *Proc. Asian Conf. Computer Vision*, vol. 1, pp. 366–371, 2004.

[10] K. Prazdny, “Detection of Binocular Disparities,” *Biological Cybern.*, vol. 52, pp. 93–99, 1985.

[11] T. Darrel, “A Radial Cumulative Similarity Transform for Robust Image Correspondence,” in *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,” in *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, 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. of Computer Vision*, vol. 28, no. 2, pp. 155–174, 1998.

[20] K. -J. Yoon and I. -S. Kweon, “Voting-based Separation of Diffuse and Specular Pixels,” *Electronics Letters*, vol. 40, no. 20, pp. 1260–1261, 2004.

[21] C. Tomasi and R. Manduchi, “Bilateral Filtering for Gray and Color Images,” in *Proc. Int’l Conf. Computer Vision*, pp. 839–846, 1998.

[22] 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.