A Novel Trilateral Filter based Adaptive Support Weight Method for Stereo Matching

Dongming Chen
http://liris.cnrs.fr/membres?idn=dchen
Mohsen Ardabilian
http://liris.cnrs.fr/membres?idn=mardabil
Liming Chen
http://perso.ec-lyon.fr/liming.chen/index.htm

LIRIS UMR 5205
Ecole Centrale de Lyon
Lyon, France

Abstract

Adaptive support weight (ASW) approach represents the state-of-the-art local stereo matching method. Recent extensive evaluation studies on ASW approaches show that the bilateral filter weight function enables outstanding performance on a large data set in comparison with various weight functions. However, it does not resolve the ambiguity induced by nearby pixels at different disparities but with similar colors. In this paper, we propose a novel trilateral filter based ASW method which remedies such ambiguities by considering disparity discontinuities through color discontinuity boundaries, *i.e.*, the strength of the boundary between two pixels. The experimental evaluation on the Middlebury benchmark shows that the proposed algorithm ranks 15^{th} out of 150 submissions and is the current most accurate local stereo matching algorithm.

1 Introduction

At the heart of computer vision aiming to endow computers with human like depth vision capabilities, stereo matching remains one of the most active research topics in this flied. As surveyed by Scharstein and Szeliski in [19], stereo matching algorithms can be broadly categorized into two classes: local and global methods. Global methods typically seek a disparity assignment which minimizes a global cost function through a global optimization algorithm, *e.g.*, graph cut [3], belief propagation [13], *etc*. Although these global methods tend to produce more accurate matching results, they are generally computationally expensive due to the iterative nature of the underlying optimization process, and thereby hardly satisfy the requirement of real-time applications. Local methods consider correlations between intensity patterns in support windows. The raw matching costs of all pixels within a support window are aggregated to the center pixel at each disparity; then an optimal disparity that gives a minimum aggregated cost is selected through an efficient local optimization process.

The performance of local methods highly depends on how the support windows is selected. A variety of cost aggregation strategies (proposed before 2008) were comprehensively analyzed in [7, 22], including the square window approach [19], the shiftable window approach [2, 6], the adaptive window approach [23], the multiple window approach [9], the adaptive support weight approach [25] and so forth. These approaches attempt to seek an optimal support window for each pixel (center pixel) by changing the window size [1, 15, 23], shape [9], center offset [2, 6, 19, 23], *etc*. The ideal optimal support window should satisfy the following rule that within this window all pixels lie on the same disparity with the center pixel. For example, in the shiftable window approach [2, 6], an optimal window that gives the smallest average cost is selected from a set of shifted support windows centered at different locations, because such a window is more likely to cover a region with the same disparity.

Because of its outstanding performance, recent years have witnessed a great deal of attention focused on the Adaptive Support Weight (ASW) based local stereo matching methods [10, 16, 21, 25], proposed firstly by Yoon and Kweon in [25]. The ASW methods assign an adaptive weight to each pixel in the support window, depending on how a support pixel is likely to lie on the same disparity with the center pixel. That is, the more likely a support pixel is to lie on the same disparity with the center pixel, the higher the weight is attributed. Essentially, the assignment of an adaptive weight for each support pixel amounts to changing the support window in terms of size, shape and center offset. In this method, the weight computation is very important, because it directly decides the support window. In [25], the adaptive support weight of a pixel is defined as inversely proportional to (1) the color dissimilarity and, (2) the spatial distance between this current pixel and the center pixel. As such, the cost aggregation step is equivalent to filter the cost volume with a joint bilateral filter with edge-preserving property. Following this pioneering work, various methods for the computation of adaptive support weight were proposed in the literature, including in particular the segmentation-based adaptive support weight of Tombari *et al.* [21], the geodesic support weight of Hosni *et al.* [10], guided filter based support weight of Rhemann *et al.* [16] which employs a different edge-preserving filter [8]. These three adaptive support weight algorithms and the original bilateral filter based algorithm [25] mainly differ in the form of support weight function. Thus, which weight function is the most accurate one? Recently, Hosni *et al.* [11] carried out a comprehensive comparative study which fairly evaluates the performance of various weight functions using a large set of 35 ground truth disparity pairs while fixing the preprocessing, matching cost function and post-processing. Their conclusion is that both bilateral filter weight function [25] and guided filter weight function [16] are the best among ten different weight functions, because bilateral filter weight function performs better on the average rank while guided filter weight function produces a lower average error.

In this paper, we revisit the bilateral filter weight function [25] and propose a trilateral filter weight function which resolves the ambiguity induced by nearby pixels at different disparities but with similar colors. Indeed, the bilateral filter weight function makes use of a color similarity term and a spatial proximity term to translate the implicit assumption that, (1) the support pixels whose color are similar to the center pixel's are likely to lie on the same disparity with the center pixel; and (2) the support pixels that are spatially close to the center pixel are also likely to lie on the same disparity. While these two simple rules can handle most depth ambiguities within a support window, they unfortunately fail to sort the disparities in the following situation as illustrated in Fig. 1. Consider two nearby objects of different disparities but with similar color (*e.g.* two planks in (b) of Fig. 1). Now given two pixels of these two objects (pixel p and q in (b)), the weight associated with the pixel q by the bilateral filter with p as center pixel will be high because their colors are similar and their spatial positions are close to each other. But obviously the weight associated with q should be low since pixels p and q are on different disparities. To further highlight this

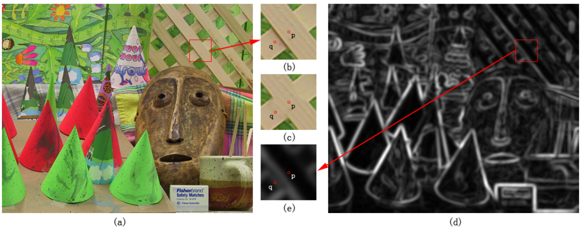

Figure 1: (a) is an image from Middlebury benchmark and we focus on the regions within the red box, zoomed in (b). In (b), there are two nearby planks with similar colors but in different disparities. An imaginary situation is presented in (c): these two planks are substituted by one cross-shaped plank in the same disparity. The bilateral filter weight of pixel p and q in (b) is equal to that in (c), because their color similarity and spatial distance are the same. But obviously these two weights should not be equal because the two pixels are in the same disparity in (c) but not in (b). The boundary cue is helpful to remedy this flaw. As shown in (d), zoomed in (e), there is a boundary between pixel p and q.

erroneous weight attribution by the bilateral filter, we present an imaginary situation in (c) where the two planks in (b) are substituted by a cross-shaped plank in the same disparity plane in (c). (Note that in (b) there are two planks in different disparities but in (c) there is only one cross-shaped plank in the same disparity). The bilateral filter weight of pixel p and q in (b) is equal to that of pixel p and q in (c), because their color similarity and spatial distance are the same. Obviously these two weights should not be equal because the two pixels are in the same disparity in (c) but not in (b). Thus, in this case, the bilateral filter weight function fails to reflect an accurate likelihood that pixel p lies in the same disparity with pixel q. We can observe that the depth discontinuity of the pixels p and q in (b) induces a color discontinuity between these two planks and results in color boundaries nearby, as shown in (e). Therefore, the boundary cue is helpful to remedy this flaw. As a result, we propose in this paper a trilateral filter weight function which extends the current bilateral filter by a third boundary strength term (boundary cue) which measures the strength of the boundary between two pixels. This boundary cue is also used in [4] to improve the non-local cost aggregation method proposed in [24], by constructing a more faithful minimum spanning tree. The experimental evaluation on the Middlebury benchmark demonstrates the effectiveness of the proposed method.

The rest of this paper is organized as follows: in Section 2, we briefly introduce the pipeline of our trilateral filter based ASW method; then, the proposed trilateral filter weight function is described in Section 3 in detail and our disparity refinement strategy is presented in Section 4. The experimental results and analysis are given in Section 5 and Section 6 concludes the paper.

2 Trilateral Filter based ASW Method

The proposed trilateral filter based ASW method encompasses the following five steps: preprocessing, matching cost computation, cost aggregation, disparity optimization and disparity refinement.

(1) Preprocessing we apply a standard median filtering (3×3 window) to preprocess the original image, in order to remove isolated pixels. The experiments in [11] show that such a preprocessing step can enhance the quality of the final disparity map.

(2) Matching Cost Computation we make use of a popular matching cost function, namely Truncated Absolute Difference of Color and Gradient (TAD C+G) [11, 16, 24]. The raw matching cost $C_d(p)$ between a pair of matching points (pixel p in left image I_l and pixel $p-d$ in right image I_r) is defined as [16],

$$\begin{aligned} C_d(p) =& (1-\theta) \times min(|I_l(p) - I_r(p-d)|, \tau_1) \\ & + \theta \times min(|\nabla_x I_l(p) - \nabla_x I_r(p-d)|, \tau_2), \end{aligned} \tag{1}$$

where ∇_x is the derivative in x direction; θ balances the color and derivative terms; τ_1, τ_2 are truncation values in order to reduce the influence of occluded pixels.

(3) Cost Aggregation we employ the state-of-the-art ASW approach. In general, there are two aggregation approaches to compute the weight mask, symmetric approaches (proposed in [25]) and asymmetric approaches (as Eq.2). According to the comparison in [11], the asymmetric approach performs better on the matching cost, calculated by the above TAD C+G cost function. The aggregated cost of pixel p at disparity d, $C_d^A(p)$ is defined using asymmetric approach as [10, 11, 16],

$$C_d^A(p) = \sum_{q \in \omega_p} w(p,q) \cdot C_d(q), \tag{2}$$

where ω_p denotes the support window centered at pixel p; q is one support pixel in ω_p. The ASW approaches [10, 16, 21, 25] differ mainly in the form of weight function $w(p,q)$, which directly decides the support window and indirectly influences the quality of final results. The proposed trilateral filter weight function is presented in Section 3.

(4) Disparity Optimization winner-take-all optimization strategy is commonly used in local methods, which searches for the optimal disparity $D(p)$ that gives the minimum aggregated cost,

$$D(p) = \arg\min_d (C_d^A(p)). \tag{3}$$

(5) Disparity Refinement in this step, the generated disparity map $D(\cdot)$ is post-processed to remove mismatches. Our disparity refinement strategy is presented in Section 4.

3 Trilateral Filter Weight Function

The proposed trilateral filter weight function is based on the bilateral filter weight function [25], but extends the color similarity term and spatial proximity term of bilateral filter by a new boundary strength term that measures the strength of potential boundary between the current support pixel and the center pixel. As pointed out in Section 1, the bilateral filter weight function fails to assign an accurate weight to two nearby pixels at different disparities but with similar colors, because it only considers color similarity and spatial distance. We observe that the boundary cue can remedy this flaw, because the discontinuity boundary indicates the possible disparity discontinuity. Thus, we also consider the boundary cue (boundary strength term) in the proposed weight function. In Section 3.1, we briefly describe the previous bilateral filter weight function, then our trilateral filter weight function is proposed in Section 3.2.

3.1 Previous Bilateral Filter Weight Function

The previous bilateral filter weight function [25] has displayed outstanding performance as shown in [11]. This weight function $w_{bf}(p,q)$ consists of color similarity term and spatial proximity term, defined as [25],

$$w_{bf}(p,q) = e^{-\frac{\Delta c_{pq}}{\gamma_c}} e^{-\frac{\Delta g_{pq}}{\gamma_g}}, \tag{4}$$

where q is a pixel within the support window centered at pixel p. The color similarity Δc_{pq} represents the Euclidean distance between the color of these two pixels, measured in the CIELab color space as

$$\Delta c_{pq} = \sqrt{\sum_{j \in (L,a,b)} (I_j(p) - I_j(q))^2} \tag{5}$$

and the geometric proximity Δg_{pq} is the Euclidean distance between their coordinates (x,y) as

$$\Delta g_{pq} = \sqrt{(p_x - q_x)^2 + (p_y - q_y)^2}. \tag{6}$$

The parameter γ_c and γ_g are set by user to adjust the color similarity term and geometric proximity term respectively.

3.2 Proposed Trilateral Filter Weight Function

The proposed trilateral filter weight function $w_{tf}(p,q)$ extends the bilateral filter weight function by a new boundary strength term, using an empirical formula inspired by the cue combination strategy proposed in [6] as,

$$w_{tf}(p,q) = e^{-\frac{\Delta c_{pq}}{\gamma_c}} e^{-\frac{\Delta g_{pq}}{\gamma_g}} \left(e^{-\frac{\Delta E_{pq}}{\gamma_e}} + e^{-\frac{\Delta c_{pq}}{\gamma_c}} e^{-\frac{\Delta g_{pq}}{\gamma_g}} \right), \tag{7}$$

where $exp(-\Delta c_{pq}/\gamma_c)$ and $exp(-\Delta g_{pq}/\gamma_g)$ are respectively the color similarity term and spatial proximity term defined in the bilateral filter weight function; whereas $exp(-\Delta E_{pq}/\gamma_e)$ is the boundary strength term calculated as follow. The parameter γ_e is set by user to adjust the boundary strength term.

The local energy model [14] is employed to detect the boundary position and measure the boundary strength (local energy). The local energy of an image is formed as a combination of the oriented energy [18] over all orientations and the oriented energy represents the energy at a given orientation calculated via orientation selective filters. Given image I, the local energy at pixel p is defined as [14, 17],

$$E(p) = \sum_{\theta} \sqrt{(I(p) * F_{\theta,odd})^2 + (I(p) * F_{\theta,even})^2}, \tag{8}$$

where $*$ denotes the convolution operator; the odd-phase filter $F_{\theta,odd}$ and the even-phase filter $F_{\theta,even}$ are a pair of quadrature filters in orientation θ, which can be calculated by Gabor function, difference of Gaussian function, difference of offset Gaussian function and so on. We choose the difference of offset Gaussian functions to calculate these filters at various scales and orientations as in [12].

The local energy has a maximum response for boundary of shape, whereas the zero-crossings of even-phase filter $F_{\theta,even}$ locate the positions of the boundaries. The phase of pixel p is defined as

$$\phi(p) = \begin{cases} 1 & ,I(p) * F_{\theta=\theta_{max},even} > 0 \\ -1 & ,I(p) * F_{\theta=\theta_{max},even} < 0 \end{cases} \tag{9}$$

where θ_{max} is the orientation that gives the maximum response as

$$\theta_{max} = \max_{\theta} \sqrt{(I(p) * F_{\theta,odd})^2 + (I(p) * F_{\theta,even})^2}. \tag{10}$$

For two neighboring pixels i and j, the distance of their edge energy, $\Delta E(i,j)$ is expressed as

$$\Delta E(i,j) = \begin{cases} E(i) + E(j) & ,\phi(i) \neq \phi(j) \\ 0 & ,\phi(i) = \phi(j) \end{cases} \tag{11}$$

For arbitrary two pixels p and q, a line l linked them can be found. Suppose that there exist N pixels on line l between p and q. Then the distance of their edge energy is

$$\Delta E(p,q) = \max_{m \in [1,\cdots,N-1]} \Delta E(m, m+1). \tag{12}$$

where m and $m+1$ represent two neighboring pixels.

As discussed before, the difference between the proposed trilateral filter weight function and the bilateral filter weight function is the boundary strength term $exp(-\Delta E_{pq}/\gamma_e)$. If there is no boundary between two pixels p and q, then its boundary strength term equals 1, because $\Delta E(p,q) = 0$; The stronger the boundary is, the smaller the boundary strength term is. For two pixels lying in the different sides of a boundary, their boundary strength term will be small.

4 Disparity Refinement

In this section, we improve the disparity refinement strategy in [24]. Firstly, taking the left and right image as reference image respectively to obtain the left and right disparity map, using the step $1 \sim 4$ described in Section 2. Then, the left-right consistency checking is applied on these two disparity maps to pick out the unstable pixels from the stable pixels. A stable pixel should pass this consistency check and a valid pixel should satisfy $D(\cdot) > 0$. A new matching cost for each pixel p at each disparity level d is computed as,

$$C_d^{new}(p) = \begin{cases} \left(1 - \exp\left(-\frac{(d-D(p))^2}{\sigma^2}\right)\right) \cdot C_d(p) & ,p\ is\ stable\ and\ D(p) > 0 \\ 0 & ,else \end{cases} \tag{13}$$

where the parameter σ is set by user. Different from the matching cost recalculation function in [24], our function contains not only the generated disparity map $D(\cdot)$, but also the raw matching cost $C_d(p)$, because the raw matching cost, calculated by TAD C+G cost function, contains the raw color similarity information between a pair of matching points.

The cost aggregation step is then performed on this new matching cost $C_d^{new}(p)$ to propagate the disparity from stable pixels to unstable pixels, and followed by the disparity optimization step to generate the final disparity map.

Algorithm	Rank	tuskuba			venus			teddy			cones			Avg.
		nocc	all	disc	nocc	all	disc	nocc	all	disc	nocc	all	disc	Error
Our TF	15	1.65	1.96	**5.90**	**0.14**	0.31	**1.51**	6.25	**11.8**	**15.1**	**2.49**	8.32	**7.02**	5.21
GF [16]	30	1.51	1.85	7.61	0.20	0.39	2.42	**6.16**	11.8	16.0	2.71	**8.24**	7.66	5.55
GEO [10]	41	1.45	1.83	7.71	0.14	**0.26**	1.90	6.88	13.2	16.1	2.94	8.89	8.32	5.80
BFSeg [21]	66	**1.25**	**1.62**	6.68	0.25	0.64	2.59	8.43	14.2	18.2	3.77	9.87	9.77	6.44
BF [25]	80	1.38	1.85	6.90	0.71	1.19	6.13	7.88	13.3	18.6	3.97	9.79	8.26	6.67

Table 1: The quantitative comparison of our trilateral filter based ASW algorithm and four popular ASW algorithms on the Middlebury benchmark with error threshold 1. Our algorithm outperforms others in most columns, especially in "disc." column.

Test	Rank	tuskuba			venus			teddy			cones			Avg.
		nocc	all	disc	nocc	all	disc	nocc	all	disc	nocc	all	disc	Error
1	97	2.51	3.23	8.28	0.81	1.57	5.88	8.19	15.1	18.2	4.21	11.9	10.7	7.55
2	105	2.58	3.14	8.87	0.95	1.72	6.77	8.92	15.9	19.3	4.88	12.4	11.7	8.10
3	103	2.81	3.67	8.56	1.44	2.50	13.4	8.53	16.0	18.5	3.43	11.6	9.21	8.30
4	15	1.65	1.96	5.90	0.14	0.31	1.51	6.25	11.8	15.1	2.49	8.32	7.02	5.21
5	29	1.91	2.23	6.31	0.17	0.41	1.95	6.82	12.5	15.6	2.39	8.25	6.86	5.45
6	45	1.89	2.33	8.70	0.13	0.40	1.75	6.99	12.7	17.2	2.77	8.77	7.88	5.96
7	23	1.78	2.09	6.72	0.15	0.32	1.63	6.29	11.9	15.1	2.52	8.37	7.07	5.33
8	39	2.11	2.31	7.58	0.23	0.35	1.60	6.19	11.5	15.1	3.00	8.19	8.45	5.55

Table 2: The detail quantitative results of the eight tests on Middlebury benchmark with error threshold 1.

5 Experimental Results

We implemented the proposed trilateral filter based ASW algorithm in C++ and evaluated it on the common accepted Middlebury benchmark [20], using four pairs of standard datasets, namely Tsukuba, Venus, Teddy, and Cones.

The experimental parameter setup is defined as follows: $\{\gamma_c, \gamma_g, \gamma_e, \sigma\} = \{8, 11, 12, 2\}$; the parameters for matching cost computation ($\{\theta, \tau_1, \tau_2\}$ in Eq.1) are set the same as those in [24] and the support window size is 35×35, which is the same as that in [25]. Firstly, we compare the proposed algorithm with four popular ASW algorithms [10, 16, 21, 25] and the quantitative comparison is presented in Table 1. On the Middlebury benchmark, the errors are evaluated over three different areas in the reference image, classified as non-occlusion (nocc.), discontinuous (disc.) and the entire image (all). From Table 1, we can observe that our proposed method outperforms these four methods, especially in the "disc." column. The term "disc." represents the regions near depth discontinuities (white areas in Fig. 2), which mainly contain boundaries. The proposed algorithm obtains a better performance in "disc." column due to the effective boundary strength term in our weight function, which is further proved by the following evaluations.

In order to better understand the performance of the proposed algorithm, we conduct eight tests to evaluate the proposed weight function and disparity refinement method individually. All tests share the same preprocessing, matching cost function and disparity optimization as described in Section 2. The only difference is the choice of weight function and disparity refinement strategy. The quantitative results of these eight configurations on Middlebury benchmark are presented in Table 2 and the test results are explained and compared in the following two subsections.

Test	Weight	Refinement	Rank	Avg.Error
1	w_{tf}	—	97	7.55
2	w_{bf}	—	105	8.10
3	w_{gf}	—	103	8.30
4	w_{tf}	ours	15	5.21
5	w_{bf}	ours	29	5.45
6	w_{gf}	ours	45	5.96

Table 3: Overall comparison the proposed weight function (w_{tf}) with the bilateral filter weight function (w_{bf}) and the guided filter weight function (w_{gf}). The detail comparison is presented in Table 2. The other steps are set the same and only weight functions are compared.

Evaluation on weight function the conclusion in [11] shows that both the bilateral filter weight function [25] and the guided filter weight function [16] are the top-performing weight functions among ten different functions. Therefore, we only compare the proposed trilateral filter weight function with these two outstanding functions. For fair comparison, the preprocessing, matching cost function and disparity optimization are fixed. The disparity refinement step is not used in test 1 $\sim$ 3, in order to compare the raw weight functions; while we employ our disparity refinement method in test 4 $\sim$ 6.

Firstly, in order to demonstrate the effectiveness of our boundary strength term, we compare our trilateral filter weight function with the bilateral filter weight function, since the only difference of these two functions is the boundary strength term. Comparing test 1 with test 2 in Table 2 and Table 3, we observe that our weight function is more accurate than the bilateral filter weight function in most columns of all four images, especially in the "disc." column in Table 2. As mentioned before, in "disc." column, the error is only evaluated in depth discontinuities region (white areas in Fig.2), mainly containing boundaries. The superior performance of our weight function in "disc." column proves the effectiveness of our boundary strength term. Comparing test 4 with test 5 in Table 2 and Table 3, in which the disparity maps are post-processed by our disparity refinement method, we also observe that our weight function outperforms the bilateral filter weight function in terms of average error and rank.

Moreover, comparing test 1 with test 3 and test 4 with test 6 in Table 2 and Table 3, we observe that our trilateral filter weight function outperforms the guided filter weight function too. All previous experiments show that our weight function is better than these two state-of-the-art weight functions. The visual comparison of the disparity maps generated by these three weight functions in test 4 $\sim$ 6 is shown in Fig.3.

Evaluation on disparity refinement in order to highlight the performance of our disparity refinement strategy, we compare our method with two state-of-the-art methods proposed in [16, 24]. In this comparison, other four steps (step 1 $\sim$ 4 in section 2) remain the same for all related tests so that only disparity refinement methods are compared. Our trilateral filter weight function is applied to calculate the support weight and conduct four tests (test 1, test 4, test 7 and test 8). The overall comparison is presented in Table 4 and the detail comparison is shown in Table 2. Test 1 is the reference result for other three tests, which is obtained without any disparity refinement strategy. As it can be seen, the other three tests are more accurate than test 1, which justifies the necessary of disparity refinement step; Note that our disparity refinement method is better than these two disparity refinement methods.

Test	Weight	Refinement	Rank	Avg.Error
1	w_{tf}	—	97	7.55
4	w_{tf}	ours	15	5.21
7	w_{tf}	[24]	23	5.33
8	w_{tf}	[16]	39	5.55

Table 4: Overall comparison the proposed disparity refinement strategy with two state-of-the-art methods, respectively proposed in [16, 24]. The detail comparison is presented in Table 2. The other steps are set the same and only disparity refinement methods are compared.

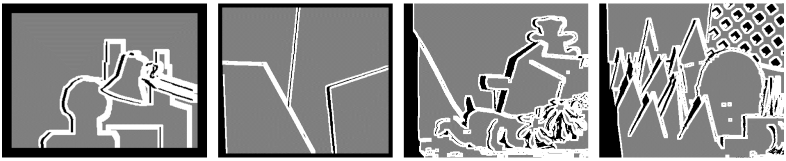

Figure 2: The term 'disc.' in Table 1 and Table 2 means the regions near depth discontinuities (white areas), occluded and border regions (black), and other regions (gray). In 'disc.' column, errors are only evaluated in the white areas, which mainly contain boundaries.

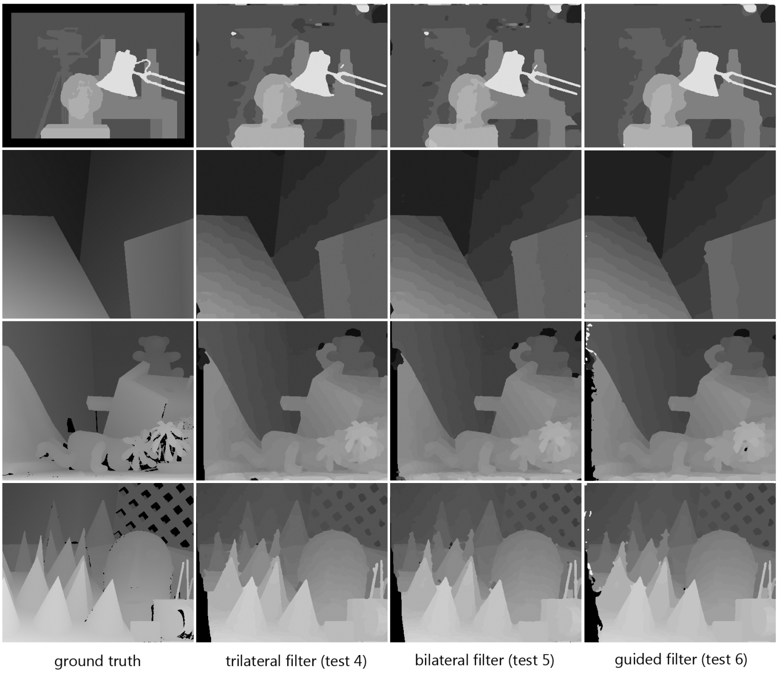

Figure 3: The visual comparison of the disparity maps generated by the proposed trilateral filter weight function (test 4), the bilateral filter weight function (test 5) and the guided filter weight function (test 6).

6 Conclusion

In this paper, we proposed a novel trilateral filter based ASW method which extends the color similarity and spatial proximity terms of bilateral filter by a new boundary strength term that measures the strength of the boundary between two pixels. Experimental results demonstrate that (1) the newly introduced boundary strength term is effective as the proposed trilateral filter weight function outperforms the bilateral filter weight function, especially in the depth discontinuity areas; (2) the proposed weight function proves to be the current best adaptive support weight function as it also outperforms the guided filter weight function. Besides, our disparity refinement strategy compares favorably with two state-of-the-art methods.

Acknowledgements This work was funded in part by the French research agency, Agence Nationale de Recherche (ANR) through the 3D Face Analyzer project (grant ANR 2010 INTB 0301 01), and the 3D Face Interpreter project supported by the LIA 2MCSI lab between the group of Ecoles Centrales and Beihang University.

References

[1] S. Adhyapak, N. Kehtarnavaz, and M. Nadin. Stereo matching via selective multiple windows. *Journal of Electronic Imaging*, 16(1):013–012, 2007.

[2] A. F. Bobick and S. S. Intille. Large occlusion stereo. *IJCV*, 33:181–200, 1999.

[3] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *PAMI*, 23(11):1222–1239, 2001.

[4] D. Chen, M. Ardabilian, X. Wang, and L. Chen. An improved non-local cost aggregation method for stereo matching based on color and boundary cue. In *ICME*, 2013.

[5] T. Cour, F. Benezit, and J. Shi. Spectral segmentation with multiscale graph decomposition. In *CVPR*, pages 1124–1131, 2005.

[6] A. Fusiello and V. Roberto. Efficient stereo with multiple windows. In *CVPR*, pages 858–863, 1997.

[7] M. Gong, R. Yang, L. Wang, and M. Gong. A performance study on different cost aggregation approaches used in real-time stereo matching. *IJCV*, 2007.

[8] K. He, J. Sun, and X. Tang. Guided image filtering. In *ECCV*, pages 1–14, 2010.

[9] H. Hirschmuller, P. R. Innocent, and J. M. Garibaldi. Real-time correlation-based stereo vision with reduced border errors. *IJCV*, 47:229–246, 2002.

[10] A. Hosni, M. Bleyer, M. Gelautz, and C. Rhemann. Local stereo matching using geodesic support weights. In *ICIP*, pages 2069–2072, 2009.

[11] A. Hosni, M. Bleyer, and M. Gelautz. Secrets of adaptive support weight techniques for local stereo matching. *CVIU*, 117:620–632, 2013.

[12] J. Malik and P. Perona. Preattentive texture discrimination with early vision mechanisms. *Journal of the Optical Society of America A*, 7:923–932, 1990.

[13] T. Meltzer, C. Yanover, and Y. Weiss. Globally optimal solutions for energy minimization in stereo vision using reweighted belief propagation. In *ICCV*, pages 428–435, 2005.

[14] M.C. Morronea and R.A. Owens. Feature detection from local energy. *PRL*, 6:303–313, 1987.

[15] M. Okutomi and T. Kanade. A locally adaptive window for signal matching. *IJCV*, 7 (2):143–162, 1992.

[16] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz. Fast cost-volume filtering for visual correspondence and beyond. In *CVPR*, pages 3017–3024, 2011.

[17] B. Robbins and R. A. Owens. 2d feature detection via local energy. *Image and Vision Computing*, 15(5):353–368, 1997.

[18] L. Rosenthaler, F. Heitger, O. Kubler, and R. von der Heydt. Detection of general edges and keypoints. In *ECCV*, 1992.

[19] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1-3):7–42, 2002.

[20] D. Scharstein and R. Szeliski. Middlebury stereo evaluation - version 2. http://vision.middlebury.edu/stereo/.

[21] F. Tombari, S. Mattoccia, and L. Di Stefano. Segmentation-based adaptive support for accurate stereo correspondence. In *PSIVT*, volume 1, pages 427 – 438, 2007.

[22] F. Tombari, S. Mattoccia, L. Di Stefano, and E. Addimanda. Classification and evaluation of cost aggregation methods for stereo correspondence. In *CVPR*, 2008.

[23] O. Veksler. Fast variable window for stereo correspondence using integral images. In *CVPR*, pages 556–561, 2003.

[24] Q. Yang. A non-local cost aggregation method for stereo matching. In *CVPR*, pages 1402–1409, 2012.

[25] K. Yoon and I. Kweon. Adaptive support-weight approach for correspondence search. *PAMI*, 28(4):650–656, 2006.