Full-Image Guided Filtering for Fast Stereo Matching

Qingqing Yang, Dongxiao Li, *Member, IEEE*, Lianghao Wang, and Ming Zhang

***Abstract*—A novel full-image guided filtering method is proposed. Different with many existing neighborhood filters, all input elements are employed during the proposed filtering approach. In addition, a novel scheme called *weight propagation* is proposed to compute support weights. It fulfills the requirements of edge preserving and low complexity. It is applied to the cost-volume filtering in the local stereo matching framework. The algorithm utilizing the proposed filtering method is currently one of the best local algorithms on the Middlebury stereo testbed in terms of both speed and accuracy.**

***Index Terms*—Cost-volume filtering, full-image guided filtering, local stereo matching, weight propagation.**

I. Introduction

ACCURACY and speed are two main performance metrics of stereo matching algorithms. Although global algorithms usually generate accurate disparity maps, they are relatively slow due to the inevitable iterations, and do not scale well to high-resolution images. On the other hand, many existing fast implementations are local algorithms.

One of the most important steps in local stereo matching algorithms is cost aggregation [1]. It is usually performed by summing up the (weighted) matching cost within a support window to remove the influence of possible noise during the similarity measuring. If we divide the sum by the normalizing constant, it is exactly a neighborhood filtering approach. In this letter, we address the cost aggregation problem from the point of view of cost-volume filtering.

Filters with fixed-size convolution kernel, e.g., uniform (box filters) or Gaussian, can be used. However, these methods fail in preserving object borders and result in poor disparity maps with fattened edges. A better choice is to utiliz"e edge preserving filters. The original adaptive weight approach [2] used the bilateral filter [3] to preserve disparity borders. One disadvantage is that, the complexity of the brute force implementation of bilateral filter is dependent on the kernel window size. Although some fast implementations [4]–[8], have been proposed in recent years, they are some kind of approximations and reduce the complexity at the price of quality degradation. In [9], the authors proposed to compute weights using geodesic distance. Though it performs better at object borders, the complexity is still relatively high. An intermediate solution [10] was proposed using the multiscale approach. Smaller support windows were used in each scale. In [11], the authors used iterative geodesic diffusion to get better results. The recently proposed guided image filter [12] fulfills both two requirements of edge preserving and linear complexity. It uses a local linear model. Local stereo matching algorithms utilizing this filter currently report the best results [13]. Although it declares the complexity of $O(1)$[1], if color guide image is used, it requires about 22 box filtering processes to finish the whole guided filtering approach. Moreover, one common shortage of existing neighborhood filters is, the support windows are of fixed size. This makes a great impact on their performance in the large low-texture regions. Hirschmüller [14] proposed a fast approximation of global cost function by path-wise optimization from multiple directions, but not all pixels are utilized. In [15], the authors used a two-pass paradigm to perform efficient aggregation, but the performance is limited by the smoothness kernel design. Recently, Yang [16] proposed a non-local solution, the cost aggregation was still constrained on a minimum spanning tree.

In this letter, a novel filtering algorithm is proposed. Every pixel contributes to the filtering approach. This ensures the filtering approach employs as many pixels as possible in the same object. The weights are calculated using the proposed *weight propagation* scheme. Since we use the weights calculated from the whole reference image to guide the filtering of the cost-volume, we refer to the proposed filtering algorithm as *full-image guided filtering*. It has an $O(1)$ exact implementation. The average operations required per pixel are four multiplications and eight additions, which is extremely fast. By applying it to the local stereo matching algorithm, competitive results can be achieved. The proposed full-image guided filter is detailed in Section II. It is applied to our local stereo matching framework in Section III. We report experimental results and give some discussions in Section IV, and conclude this letter in Section V.

II. The Proposed Filter

A. Filter Modeling

We first denote the signal/image to be filtered as C, the guide image as I, and the filter output as C'. Then the output value C'_i for pixel i is defined by

$$C'_i = \frac{1}{N_i} \sum_{j \in \Omega} W_{i,j}(I) C_j. \tag{1}$$

[1]The computational complexity is estimated by counting the operations required for each pixel.

Manuscript received October 31, 2012; revised January 05, 2013; accepted January 15, 2013. Date of publication January 22, 2013; date of current version February 01, 2013. This work was supported in part by the National Natural Science Foundation of China (Grants 60802013, 61072081, and 61271338), the National High Technology Research and Development Program (863) of China (Grant 2012AA011505), the National Science and Technology Major Project of the Ministry of Science and Technology of China (Grant 2009ZX01033-001-007), and the Key Science and Technology Innovation Team of Zhejiang Province, China (Grant 2009R50003). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Weisi Lin.

The authors are with the Institute of Information and Communication Engineering, Zhejiang University, and Zhejiang Provincial Key Laboratory of Information Network Technology, Hangzhou 310027, China (e-mail: qqyang@zju.edu.cn; lidx@zju.edu.cn; wanglianghao@zju.edu.cn; zhangm@zju.edu.cn).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LSP.2013.2241759

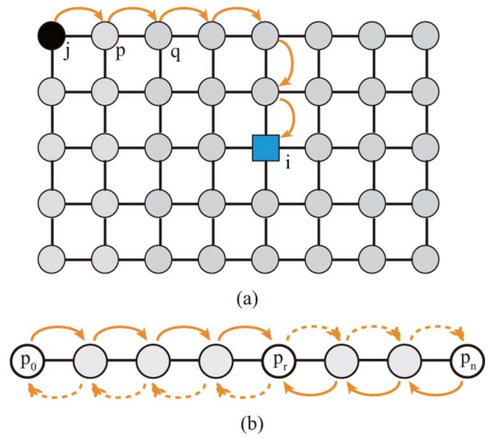


Fig. 1. (a) Weight propagation on the four-connected grid of the guide image. Orange arrows represent the propagation path $P(i,j)$. (b) Example of horizontal filtering pass. Weighted sum of pixel p_r can be computed by adding the intermediate results stored in two scanned arrays.

Here, $N_i = \sum_j W_{i,j}(I)$ is a normalizing constant, $W_{i,j}(I)$ is the weight of pixel pair (i,j) depending on the guide image I, which can be either gray or color, and Ω represents the set that includes *all* the element indices of the input signal/image.

To compute the weight for each pixel pair (i,j), we propose a novel scheme called *weight propagation*. Before describing how it works, we first build a four-connected grid based on the guide image. Every pixel is presented as a node on this grid, as shown in Fig. 1(a). The weight of pixel pair (i,j) is then computed by

$$W_{i,j}(I) = \prod_{(p,q)\in P_{i,j}} T_{p,q}(I), \tag{2}$$

where $(p,q) \in P_{i,j}$ are adjacent nodes on the path $P_{i,j}$ that connects node pair (i,j), and $T_{p,q}(I)$ is a propagation function that defines the capacity of weight propagation between two adjacent nodes. To be precise, the propagation function is defined by

$$T_{p,q}(I) = \exp\left(-\frac{\|I_p - I_q\|}{\sigma}\right), \tag{3}$$

where the pixel similarity between the two adjacent pixels (p,q) are measured by the Euclidean distance in the RGB color space, and σ is a constant parameter that adjusts the smoothness degree of the filtering.

There are numerous paths $P_{i,j}$ that can connect node pair (i,j). The best path would be the one that has the minimum propagated weight. But the computational complexity will be too high to have an efficient implementation. We choose the path using a horizontal-first policy. For each end-node j, the weight is initialized to 1. The weight value is first propagated in the horizontal direction toward the target pixel i. After the path reaches the column where i lies in, it is then propagated in the vertical direction until it reaches the target node. We refer to this process as *weight propagation*, which is illustrated by Fig. 1(a).

B. $O(1)$ *Time Exact Implementation*

It is work-inefficient to compute weights for each node pair, we implement the proposed filtering method using a two-pass model. The filtering approach is first carried out in the horizontal direction in separate rows. And the intermediate values are processed using the same way in separate columns. Take the horizontal pass as an example, for an element r in a row, the intermediate sum of weighted value is composed of three terms: the weighted sum from the left and the right plus its own value C_r, which can be expressed by

$$\begin{aligned} C_r^{imed} &= \sum_{i=0}^{r-1} W_{i,r}(I)C_i + \sum_{i=r+1}^{n} W_{i,r}(I)C_i + C_r \\ &= \sum_{i=0}^{r-1}\left(\prod_{u=i+1}^{r} T_{u^-,u}(I)\right)C_i \\ &\quad + \sum_{i=r+1}^{n}\left(\prod_{u=r}^{i-1} T_{u,u^+}(I)\right)C_i + C_r. \end{aligned} \tag{4}$$

Here, u^- is the left neighbor of u, and u^+ represents the right neighbor. The calculation of (4) can be further accelerated by using the two-pass scan paradigm, which is commonly used [15], [17] when symmetrical calculation is possible. One scan process is carried out from the left to right and the other is performed in the reverse direction. The intermediate results are stored in two temporary arrays. The scan process is a sequential computation of weighted cumulative sum on the input array, which can be expressed by

$$\begin{aligned} A_r^L &= C_r + \sum_{i=0}^{r-1}\left(\prod_{u=i+1}^{r} T_{u^-,u}(I)\right)C_i \\ &= C_r + T_{r-1,r}(C_{r-1} + T_{r-2,r-1}(C_{r-2} + \cdots + T_{0,1}(C_0)\cdots)) \\ &= C_r + T_{r-1,r}A_{r-1}^L, \end{aligned} \tag{5}$$

where A^L represents the temporary array which stores the weighted cumulative sums calculated from the left to right. Let A^R be the array that stores the weighted sums calculated in the reversed direction, (4) can be simply computed by the following equation:

$$C_r^{imed} = A_r^L + A_r^R - C_r. \tag{6}$$

This reduces the complexity to a great extend. The average operations required for each element are only two multiplications and four additions in the horizontal pass.

After the horizontal pass, the vertical pass is performed on the intermediate results in separate columns. The same operations used in horizontal pass can be utilized. Then, after the vertical pass, the average operations required for each element are four multiplications and eight additions.

Considering the normalizing term N_i in (1), the operations required for each pixel will double. Nevertheless, the computational complexity of the full-image guided filtering is exactly $O(1)$. In some applications such as the local stereo matching problem, the normalizing term N_i can be omitted to further speed up the algorithm.

C. *Comparison With Neighborhood Filters*

The main advantage of the proposed full-image guided filter over existing neighborhood filters is that, the filtering approach is supported by all the available pixels. It ensures that the filter

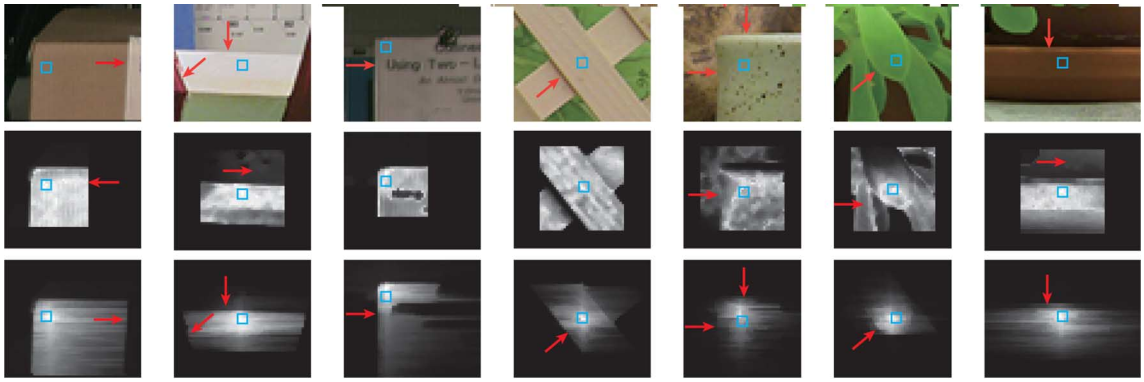


Fig. 2. Support weights for selected regions of the Middlebury benchmark images. The first row contains image crops. The second row shows support weights computed by the bilateral filtering algorithm with a support window of 35 × 35. Support pixels are constrained in the fixed-size window, and many background pixels are given relatively high weights. Support weights computed by our method are in the third row. Due to the weight propagation, support pixels are not constrained and wrong large weights in the background are avoided.

will employ as many related pixels as possible. This property is important for cost filtering in large textureless regions. Reasonable support weights are computed through *weight propagation*, which preserves sharp object edges. These advantages can be illustrated by the support weight examples of Fig. 2. We have compared the support weights computed by the proposed method and the bilateral filter, which is a typical neighborhood edge-preserving filter.

Fig. 2(a) shows the weights of support pixels corresponding to the center pixel within a textureless region. Constrained by the fixed kernel size, the number of support pixels of bilateral filtering are limited. While in the proposed method, all the pixels in the same region as the center pixel contribute to the filtering approach. Moreover, sharp object edges are well preserved. Fig. 2(b)–(c) provide more such examples.

When computing support weights, many neighborhood filters rely heavily on the similarity term, which measures the color difference to the center pixel. Thus, background pixels in the support window will easily be assigned high weights. It is not desirable for preserving sharp edges. As shown in the second row in Fig. 2, many pixels in different objects are assigned relatively large weights. To overcome this shortage, in [2], a weight term measuring the pixel similarity between stereo images was added. We compute support weights depending on the guide image only. Weight values are propagated from the periphery to the center pixel. Pixels in different regions will be assigned extremely low weights, and the influence of these pixels can be neglected during the filtering. Fig. 2(d)–(g) show the support weights in some selected regions.

Despite the advantages described above, the support weight computing of the proposed filter heavily depends on the quality of guide images. Noisy guide image will degrade the filtering results to some extent.

III. Application to Local Stereo Matching

In a general local stereo matching algorithm, four steps are carried out to generate the final dense disparity map. They are: cost computation, cost-volume filtering, winner-take-all (WTA) disparity selection and post-processing. During the cost computation stage, a cost volume C is calculated. It is a three dimensional array which stores the matching costs of every pixel for all possible disparity candidates. Let $C_{i,d}$ represents the cost value when pixel at $i = (x, y)$ is assigned disparity d, we compute the correspondence cost by

$$C_{i,d} = (1-\alpha)\min\left(C^{BT}_{i,d}, \tau_1\right) + \alpha\min\left(C^{GD}_{i,d}, \tau_2\right), \quad (7)$$

where α is a parameter that balances two sub cost terms, and τ_1 and τ_2 are truncation values. C^{BT} presents the cost of BT measure [18], and C^{GD} is the absolute difference of the gradient of stereo pairs.

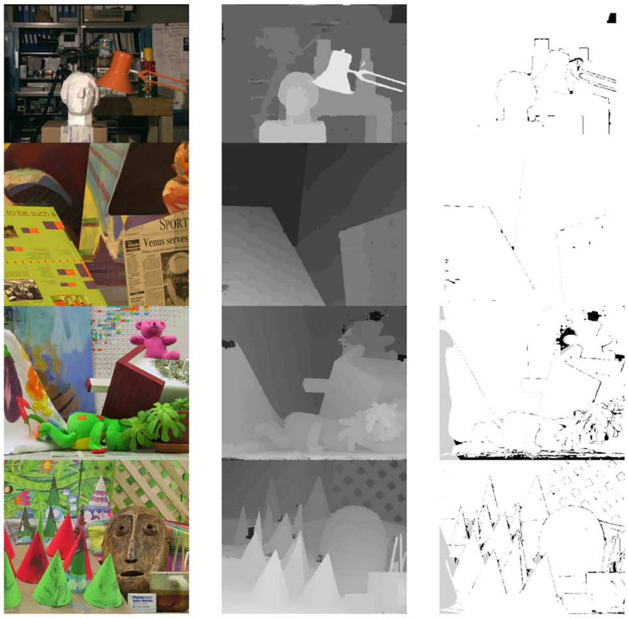
Then, for each possible disparity candidate d, the filtered cost volume C'_d of the dth cost slice C_d is computed using the proposed full-image guided filtering (1). After cost volume filtering, disparity D_i is selected by the WTA strategy:

$$D_i = \arg\min_{d\in R} C'_{i,d}, \quad (8)$$

where R is the range of candidate disparity values. Occlusions are handled in the post-processing step. In our algorithm, occlusions and mismatched pixels are first detected by cross checking. Then they are filled by the lowest disparity value of the nearest non-occluded pixel on the same scanline, and a weighted median filtering is applied to remove possible streak-line effects.

IV. Experimental Results

The performance is tested on the Middlebury stereo benchmark [19]. The smoothness adjusting parameter is optional and depends on the guide image only. It is set to the constant value $\sigma = 0.08$ for all four benchmark stereo pairs. The generated disparity results along with corresponding error maps are plotted in Fig. 3. We can observe that edges are well preserved in the disparity maps without the aid of image segmentation. More importantly, disparity in large low-texture regions are also correctly recovered, which is difficult for most local stereo matching algorithms. However, there are still some errors near the pink teddy



(a) (b) (c)

Fig. 3. Results of benchmark stereo images on the Middlebury website. From the top to bottom are: Tsukuba, Venus, Teddy and Cones. (a) Left image; (b) results of the proposed method; (c) error maps (bad estimates with absolute disparity error >1.0).

TABLE I
OBJECTIVE EVALUATION RESULTS ACCORDING TO THE MIDDLEBURY STEREO EVALUATION PLATFORM [19]

Algorithm	Rank	Avg. Error(%)	Error pixels in non-occluded(%)			
			Tsukuba	Venus	Teddy	Cones
Proposed	**12**	**4.86**	**1.51**	**0.23**	**5.43**	**2.16**
CostFilter [13]	21	5.55	1.51	0.20	6.16	2.71
NLFilter [16]	26	5.48	1.47	0.25	6.01	2.87
GeoSup [9]	29	5.80	1.45	0.14	6.88	2.94
AdaptWeight [2]	68	7.26	1.38	0.71	7.88	3.97

bear in data set Teddy, which is caused by the small repetitive textures.

Our stereo matching algorithm takes the 12th place in all 136 submissions on the stereo evaluation platform up to the time of submission. Almost all better performing algorithms use global optimization and/or segmentation. The average errors can be reduced up to 4.06%, but it usually takes several minutes to get such results. Our algorithm achieves competitive results with extremely low complexity. Furthermore, our stereo matching algorithm performs the best for data set Cones in both non-occluded and discontinuity regions among all the submitted methods. According to the evaluation platform, the proposed method is currently one of the best local stereo matching algorithms when considering the average performance. Quantitative results are summarized in Table I. The comparative results show that the proposed method clearly outperforms other local methods.

We have tested the proposed method on a desktop with Core Duo 3.16 GHz CPU and 2 GB 800 MHz RAM. No parallelism technique is utilized. The average runtime for cost-volume filtering on the Middlebury benchmark data sets is about 68 ms. It performs 2.76 $\times$ faster than the non-local cost aggregation solution [16] (188 ms), and 27 $\times$ faster than the approach [13] using guided image filtering (1850 ms).

V. CONCLUSION

A new guided filtering algorithm is proposed. All elements in the input signal contribute to the filtering approach. The novel *weight propagation* method ensures support elements are assigned reasonable weights while fulfilling the requirements of edge preserving and low complexity. We have applied it to the cost-volume filtering. The experimental results demonstrate that the proposed algorithm outperforms all local methods on the Middlebury benchmark in terms of both speed and accuracy.

REFERENCES

[1] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. J. Comput. Vision*, vol. 47, pp. 7–42, 2002.
[2] K.-J. Yoon and I. S. Kweon, "Adaptive support-weight approach for correspondence search," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 28, no. 4, pp. 650–656, 2006.
[3] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Proc. IEEE Int. Conf. Comput. Vis.*, 1998, pp. 839–846.
[4] F. Porikli, "Constant time $O(1)$ bilateral filtering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2008, pp. 1–8.
[5] S. Paris and F. Durand, "A fast approximation of the bilateral filter using a signal processing approach," *Int. J. Comput. Vision*, vol. 81, no. 1, pp. 24–52, 2009.
[6] K. N. Chaudhury, D. Sage, and M. Unser, "Fast $O(1)$ bilateral filtering using trigonometric range kernels," *IEEE Trans. Image Process.*, vol. 20, no. 12, pp. 3376–3382, 2011.
[7] C. Richardt, D. Orr, I. Davies, A. Criminisi, and N. A. Dodgson, "Real-time spatiotemporal stereo matching using the dual-cross-bilateral grid," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 510–523.
[8] S. Mattoccia, S. Giardino, and A. Gambini, "Accurate and efficient cost aggregation strategy for stereo correspondence based on approximated joint bilateral filtering," in *Proc. Asian Conf. Comput. Vis*, 2010, pp. 371–380.
[9] A. Hosni, M. Bleyer, M. Gelautz, and C. Rhemann, "Local stereo matching using geodesic support weights," in *Proc. IEEE Conf. Image Process.*, 2009, pp. 2093–2096.
[10] D. Min and K. Sohn, "Cost aggregation and occlusion handling with wls in stereo matching," *IEEE Trans. Image Process.*, vol. 17, no. 8, pp. 1431–1442, 2008.
[11] L. De-Maeztu, A. Villanueva, and R. Cabeza, "Near real-time stereo matching using geodesic diffusion," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 34, no. 2, pp. 410–416, 2012.
[12] K. He, J. Sun, and X. Tang, "Guided image filtering," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 1–14.
[13] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz, "Fast cost-volume filtering for visual correspondence and beyond," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2011, pp. 3017–3024.
[14] H. Hirschmüller, "Stereo processing by semiglobal matching and mutual information," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 30, no. 2, pp. 328–341, 2008.
[15] C. Cigla and A. Alatan, "Efficient edge-preserving stereo matching," in *IEEE Int. Conf. Comput. Vis. Workshops*, 2011, pp. 696–699.
[16] Q. Yang, "A non-local cost aggregation method for stereo matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, 2012, pp. 1402–1409.
[17] M. Bleyer and M. Gelautz, "Simple but effective tree structures for dynamic programming-based stereo matching," in *Proc. Int. Conf. Comput. Vis. Theory Appl.*, 2008, pp. 415–422.
[18] S. Birchfield and C. Tomasi, "A pixel dissimilarity measure that is insensitive to image sampling," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 20, no. 4, pp. 401–406, 1998.
[19] D. Scharstein and R. Szeliski, Middlebury Stereo Evaluation—Version 2 2002 [Online]. Available: http://vision.middlebury.edu/stereo/eval