Pattern Recognition Letters 49 (2014) 201–206

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

Real-time local stereo via edge-aware disparity propagation ☆

Xun Sun [a,b,*], Xing Mei [c], Shaohui Jiao [b], Mingcai Zhou [b], Zhihua Liu [b], Haitao Wang [b]

[a] Baidu Institute of Deep Learning, Beijing, China
[b] China Lab, Samsung Advanced Institute of Technology, Beijing, China
[c] NLPR, Institute of Automation, Chinese Academy of Sciences, Beijing, China

ARTICLE INFO

Article history:
Received 24 December 2013
Available online 1 August 2014

Keywords:
Stereo matching
Disparity propagation
Edge-preserving smoothing

ABSTRACT

This letter presents a novel method for real-time local stereo matching. Different from previous methods which have spent many efforts on cost aggregation, the proposed method re-solves the stereo problem by propagating disparities in the cost domain. It is started by pre-detecting the disparity priors, on which a new cost volume is built for disparity assignment. Then the reliable disparities are propagated via filtering on this cost volume. Specially, a new $O(1)$ geodesic filter is proposed and demonstrated effective for the task of edge-aware disparity propagation. As can be expected, the proposed framework is highly efficient, due to leaving double aggregation on left–right views, as well as costly post-processing steps, out of account. Moreover, by properly designing a quadric cost function, our method could be extended to good sub-pixel accuracy with a simple quadratic polynomial interpolation. Quantitative evaluation shows that it outperforms all the other local methods both in terms of accuracy and speed on Middlebury benchmark. It ranks 8th out of over 150 submissions if sub-pixel precision is considered, and the average runtime is only 9 ms on a NVIDIA GeForce GTX 580 GPU.

1. Introduction

Dense two-frame stereo is one of the most extensively studied topics in computer vision [14]. It takes in a rectified image pair captured from two viewpoints, and aims at inferring the depth information in the form of disparity. Generally, different stereo algorithms could be broadly separated into two major classes: *local* and *global* methods.

Local methods compute each pixel's disparity independently. They aggregate the cost (support) from a local region (usually a window) with an implicit smoothness assumption, and choose the disparity hypotheses with the minimal cost (Winner-Take-All). On the contrary, global methods optimize the disparities of all pixels at the same time. They formulate the stereo problem with an explicit smoothness assumption, and infer the disparity map by minimizing a global energy function. To approximate the energy minimum, powerful global optimizers such as belief propagation [3,23] and graph cuts [2] are used. The adoption of these time-consuming optimizers make these global methods far away from being real-time.

Global methods were generally expected to be more accurate than local methods until the use of modern edge-preserving filters [16,11,4]. Yoon and Kweon [25] firstly demonstrated that, by aggregating costs with a joint bilateral filter, the local methods could produce good results on par with those generated by global methods. Although a full-kerneled implementation of bilateral filtering is slow, since that time adaptive support-weight approach has become popular in the stereo community, and various great improvements on weighting scheme are continuously being made [13,12,19,20,8]. Current leading local methods can produce high-quality disparity map with sharp edges, and usually are with a kernel independent of any window size thus very efficient [6]. All of the above works have greatly advanced the state-of-arts. However, their efforts mainly focus on the cost aggregation step, and the computational redundancy still remains. Recalling the recent state-of-art local methods, nearly all of them share such a same work flow: firstly the matching costs on the left and right views are aggregated separately. Then with the WTA results on both views, a left–right consistency check is employed to remove the occluded pixels. Finally these "missing" pixels are filled with a background interpolation followed by a weighted-median filter. Even with acceleration, the adaptively-weighted median filter (e.g., with bilateral weights [13]) is time-consuming (with a complexity comparable to the cost aggregation). As a result, to calculate a final disparity map on the reference view (left view), this

☆ This paper has been recommended for acceptance by L. Yin.

* Corresponding author at: Baidu Institute of Deep Learning, Beijing, China. Tel.: +86 10 18612940083.
E-mail address: xun.s@qq.com (X. Sun).

http://dx.doi.org/10.1016/j.patrec.2014.07.010
0167-8655/

popular pipe-line would take *three* expensive operations in aggregation-level.

Recently, Ma et al. proposed to directly apply a weighted median filter on a noisy disparity map [7]. The authors claimed that, if the inliers are dominant in local regions, only one aggregation-level operation by weighted median filtering could suffice to remove outliers while preserving edges. Nevertheless, there are non-trivial errors that could not be recovered in such a mechanism: to generate the initial disparity map, they use simple box-aggregation and background interpolation to fill in the holes after left–right check. This is questionable since not only occluded pixels fail in this test, a large portion of other pixels, e.g., pixels in textureless regions could fail as well. The main reason of this problem is the poor matching quality of box-aggregation (comparing to the sophisticated methods). Therefore, their method could lead to "blur" results when mis-matched pixels dominate the local regions. Even median filter cannot eliminate this kind of errors.

In this letter, the local stereo problem is re-examined, and a novel framework with only one aggregation-level operation is proposed. At first, inspired from recent insights [9], with the box-filtered initial matching costs, a set of stable pixels are determined by a left–right consistency check. Besides, for each stable pixel, a compact sub-set for its disparity varying range is detected from the cost profile. A new cost volume is then built with the above two key ingredients. More concretely, the cost for a pixel is re-computed according to its membership (stable or unstable). For a stable pixel, a penalty is put on disparity variation departing from its pre-detected disparity priors. In contrast, for an unstable pixel, the cost will keep zero for all disparity hypotheses. Thus an edge-aware filter applied on this new cost volume leads to reliable disparity propagation in cost domain: for the unstable pixels, the cost for their disparity selection will purely depend on the stable pixels. Comparing to [7], our solution delivers sharp disparity maps in a unified framework with no early hard decision. Both the occluded and mis-matched pixels are considered as unstable pixels, and their disparities are assigned by softly aggregating support from stable pixels. Comparing to [17], a previous global method which models the regularization from ground control points into a energy term, our method achieves real-time performance and do not need a sophisticated mechanism for GCPs' generation. Another great advantage of our method over discrete method such as [7] is that, with a specially designed quadratic cost term, our method could be extended to good sub-pixel accuracy using a simple quadratic polynomial interpolation.

To avoid prorogating disparities across depth edges, a new $O(1)$ geodesic filter inspired from [19] is proposed in this letter. Comparing to the original MST (minimal spanning tree) based filter, this new filter has two important advantages: firstly it is more GPU-friendly since it treats each scanline as a separate tree. Secondly, it makes use of every edge on the 4-connected image grid hence no information is lost as in a MST structure. These merits combined make our filtering scheme not only more efficient, but also more accurate than [19]. There are other image grid based filtering techniques such as [12,20,18], however, recursive filter based methods [12,20] suffer from a directional bias (see Section 2.1 [18]) needs an additional high-dimensional buffer for saving the temporal results in the case of cost-volume filtering (in contrast, our filtered values are updated in place).

Experimental results demonstrate the effectiveness of our approach. It is current top performer both in terms of accuracy and speed on Middlebury benchmark. In summary, the contributions in this work are:

- A novel and simplified framework for local stereo matching with leading accuracy and real-time performance.
- A $O(1)$ geodesic filter proposed for disparity propagation. This filter is very effective for stereo matching.

2. Algorithm

This section presents the proposed stereo matching algorithm. In Section 2.1, the stereo framework by propagating reliable disparities is presented. Then in Section 2.2, a new $O(1)$ geodesic filter is presented for the purpose of preserving structures in disparity propagation (Section 2.1).

2.1. Local stereo via reliable disparity propagation

The framework of the proposed reliable disparity propagation scheme is shown in Fig. 1. First, a set of stable pixels, as well as their disparity sub-sets (candidates with high likelihood to be correct) are determined in a pre-processing step. Second, a new cost volume is built purely with the detected disparity priors. Then an edge-preserving filter is applied on this cost volume for disparity propagation. Finally, the disparity map is calculated by WTA optimization. In the following, the detailed descriptions of the pre-processing step and disparity propagation procedure are presented.

Pre-processing. In this step, all the image pixels are roughly divided into stable or unstable pixels. Then for each stable pixel, a compact disparity sub-set is extracted.

Given a pixel $\mathbf{p} = (x, y)$ in I^{Left}, a matching cost $C(\mathbf{p}, d)$ at disparity d is initialized with the exact measure used in [13,19,12]. To suppress the noises, a 5×5 box filter is applied on the initial cost volume. This is done for all disparity levels, but its complexity is trivial since such a small-kerneled filter is very efficient in GPU implementation (the average runtime is only 1 ms on Middlebury data sets, which is negligible comparing to other operations). With the initial cost, a raw disparity map D^{Left}_{Raw} on left view is calculated by WTA. Symmetrically, a corresponding disparity map D^{Right}_{Raw} is also computed. Then a left–right consistency check is employed to roughly divide all image pixels into stable or unstable pixels. For a stable pixel $\mathbf{p}$, its disparity value d on D^{Left}_{Raw} should strictly equal to $D^{Right}_{Raw}(\mathbf{p} - (d, 0))$.

Recently, Min et al. [9] proposed a compact representation for local stereo matching. In their approach, a per-pixel sub-set of disparity searching range is extracted by box-filtering and cost profile analysis. In this letter, a simplified compact representation is used and it is proven to be still very effective. Instead of sorting the local

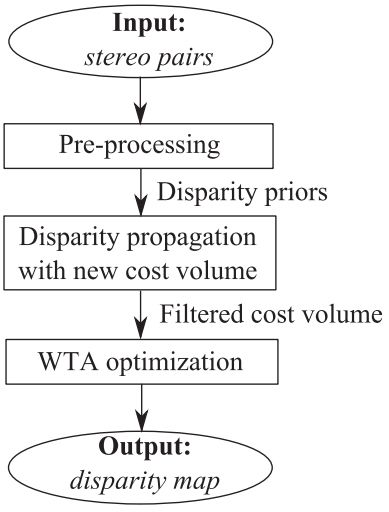


Fig. 1. Block diagram of our stereo matching algorithm.

minimums on cost profile, D_c disparity candidates are selected by directly choosing the disparities corresponding to the minimal D_c costs ($D_c \ll D_f$, where D_f is the full disparity range, and D_c is a pre-defined number which is set to 3 throughout). Disparities which reside in the sub-set are supposed to be correct with high likelihood. A difference between our disparity compact representation and [9] is that we only extract the compact representation for stable pixels, i.e., pixels which pass the left–right check. The densities, outlier percentages and recall rate of compact sub-set for stable pixels (measured in non-occluded regions) are reported in Table 1. The reported numbers are calculated from Middlebury standard data sets. As can be seen, although our pre-processing step is quite simple, the quality of extracted disparity priors is fairly high.

Disparity propagation. With the extracted informative disparity priors, now we propagate the disparities from stable pixels to the unstable pixels. Given D_{Raw}^{Left} and a per-pixel membership assignment (stable or unstable), a quadric new cost is computed for each pixel $\mathbf{p}$ at each disparity level d as

$$C(\mathbf{p},d)_{new} = \begin{cases} \left|d - D_{Raw}^{Left}(\mathbf{p})\right|^2 + R(\mathbf{p},d) & \mathbf{p} \text{ is stable,} \\ 0 & \text{else.} \end{cases} \tag{1}$$

where $R(\mathbf{p},d)$ is a penalty function for disparity varying away from per-pixel candidate sub-set. Supposing each hypothesis in the compact sub-set represents a vote, $R(\mathbf{p},d)$ is computed by

$$R(\mathbf{p},d) = \sum_{i=1}^{D_c} \begin{cases} \lambda_c \cdot |\, d - d_i|^2 & |\, d - d_i| \leq 1, \\ \lambda_t & \text{else.} \end{cases} \tag{2}$$

where λ_c is a parameter which controls the increasing rate of $R(\mathbf{p},d)$, and λ_t is a threshold which is set to be $2\lambda_c$. Note that for all unstable pixels (may be caused by occlusion, texture-less regions, specularity or poor matching power of box aggregation etc.), the new matching cost keeps zero for all disparity candidates. Hence a cost volume filtering on $C(\mathbf{p},d)_{new}$ makes the unstable pixels' matching cost completely depend on stable pixels. The result turns out to be a disparity propagation from the stable pixels to unstable pixels. The reason why we employ $R(\mathbf{p},d)$ into the new cost computation for disparity propagation (Eq. (1)) is twofold: first, even for stable pixels there are mis-matched errors which could not be neglected (see Table 1), to this end $R(\mathbf{p},d)$ adjusts the penalty to disparity variation to be more *soft*, due to the adoption of multi-label sub-sets; second, we experimentally observe that the incorporation of $R(\mathbf{p},d)$ significantly increases the matching accuracy.

To preserve the edges in disparity propagation in real-time performance, the filtering operation can be readily realized with any efficient edge-aware filters such as $O(1)$ bilateral filter [22], guided image filter [5] or recursive filter [4,20] etc. The later two kinds of filter are faster and require no quantization. It has been noticed that recursive filter such as domain transform filter is more suitable for image filtering but guided image filter excels at preserving image structures [5,7]. In contrast, guided image filter is 3x slower than domain transform filter on GPU [12]). In this letter, a new $O(1)$ geodesic filter is proposed for the task of edge-aware disparity propagation. Our proposed filter can better preserve image structures than recursive filer, and runs much faster than the guided image filter.

Table 1
Density, outlier and recall rate (measured in non-occ regions) for stable pixels on Middlebury standard data sets. Outlier (%) is the percentages of stable pixels whose absolute disparity error is larger than 1 pixel. Recall rate (%) is the percentages of stable pixels whose ground truth disparity value reside in the compact sub-set.

	Tsukuba	Venus	Teddy	Cones	Avg.
Outlier (%)	2.8	2.2	4.9	1.6	2.9
Density (%)	84.0	82.8	82.1	86.6	83.9
Recall (%)	98.9	98.8	97.6	98.5	98.4

2.2. O(1) geodesic filter for stereo matching

In this section, a novel $O(1)$ geodesic filter is proposed. The proposed filter is applied on the new cost volume which is computed from Eq. (1). This filter is greatly inspired by Yang's non-local aggregation method [19], which aggregates the matching cost of all the pixels with the weights determined by the exact geodesic distance defined on a tree. In contrast to their solution, our filter treats each scanline (either horizontal or vertical) as a separate tree. The tree nodes are all the image pixels along the scanline. Without the loss of generality, we define the left-most/right-most pixel as leaf/root node respectively. Then the 1D horizontal geodesic filter is performed via a two-pass aggregation as illustrated in Fig. 2.

For the first scan pass from left to right (from leaf to root), each pixel's cost value is sequentially updated as

$$C'(\mathbf{p},d)_{new} = C(\mathbf{p},d)_{new} + \alpha(\mathbf{p},\mathbf{pl}) \cdot C'(\mathbf{pl},d)_{new} \tag{3}$$

where $\alpha(\mathbf{p},\mathbf{pl})$ is determined by an edge cost $\Delta(\mathbf{p},\mathbf{pl})$ (the maximum of absolute difference computed separately from RGB channels between pixel $\mathbf{p}$ and its left-neighbor $\mathbf{pl}$) computed as

$$\alpha(\mathbf{p},\mathbf{pl}) = \exp\left(-\frac{1}{\sigma_s} - \frac{\Delta(\mathbf{p},\mathbf{pl})}{\sigma_r}\right) \tag{4}$$

where σ_s and σ_r are two parameters for balancing the spatial and range (color) components respectively. While this left-to-right scan appears to be similar to the 1st order recursive bilateral filter [20], a major difference for our scheme lies in the second scan from right to left (from root to leaf): each pixel collects supports from its right side and its value is updated by

$$C''(\mathbf{p},d)_{new} = (1 - \alpha^2(\mathbf{p},\mathbf{pr})) \cdot C'(\mathbf{p},d)_{new} + \alpha(\mathbf{p},\mathbf{pr}) \cdot C''(\mathbf{pr},d)_{new} \tag{5}$$

After the two-pass scan, the supporting weight between arbitrary two pixels $\mathbf{p}$ and $\mathbf{q}$ along the scanline is

$$W(\mathbf{p},\mathbf{q}) = \exp\left(-\frac{|\mathbf{p}-\mathbf{q}|}{\sigma_s} - \frac{\sum\Delta(\mathbf{pi},\mathbf{pj})}{\sigma_r}\right) \tag{6}$$

where pixels $\mathbf{pi}$ and $\mathbf{pj}$ are any two neighboring pixels along the path from $\mathbf{p}$ to $\mathbf{q}$. The correctness of Eq. (6) could be easily proved by considering the 1D scanline data as a special tree structure (see [19] for a detailed proof). The only difference is that we include a spatial component. For 2D filtering, the proposed 1D filter is then performed vertically on the result produced by horizontal passes.

Similar as [19], the proposed filter is also a $O(1)$ filter whose computational complexity is linear in both image size and

(a) 1st pass: from left to right.

(b) 2nd pass: from right to left.

Fig. 2. Two-pass filtering on a horizontal scanline.

Fig. 3. The toy example for filtering scheme comparison.

dimensionality. In contrast, it is GPU-friendly since it takes each scanline as a seperate processing unit. Moreover, it do not need a extra-step for tree construction. Last but not the least, it utilizes every edge on a 4-connected grid graph, thus better effect could be reasonably expected. For a clear comparison between our proposed filter and the previous representative filter implemented as a recursive filter (domain transform filter [4]), the recursive filtering scheme is shortly reviewed at first, and then the advantages of our filter are illustrated with a toy example.

Given a 1D signal x, the output 1D signal y of a 1st-order recursive filter like in [4] is

$$y_i = (1-\alpha)\cdot x_i + \alpha \cdot y_{i-1} \tag{7}$$

where $\alpha \in [0,1]$ is the *feedback coefficient* for controlling the smoothing effect. To implement an edge-preserving filter, α could be defined by Eq. (4) between two neighboring pixels. In recursive filter the update process defined in Eq. (7) is performed twice: 1st time from left to right (casual pass), and the 2nd time from right to left on the filtered values (anti-casual pass). For 2D image filtering, the 1D recursive filter is also performed in horizontal-vertical manner. When applied to cost aggregation, the difference between the proposed filter with recursive filter is illustrated with a toy example shown in Fig. 3. In this example, there are three pixels (**p1**, **p2** and **p3**) along a horizontal scanline. They are with initial costs of $C1, C2$ and $C3$. And the feedback coefficients on the two edges are α and α' respectively. After two-pass scan (from left to right and vice versa), the $C2$ for the center pixel is updated in recursive filter as

$$C2' = \alpha\cdot(\alpha'^2 + 1 - \alpha')\cdot C1 + (1-\alpha)\cdot(\alpha'^2 + 1 - \alpha')\cdot C2 + (1-\alpha')\cdot\alpha'\cdot C3 \tag{8}$$

whereas if our proposed filter is adopted, the $C2$ is updated as

$$C2' = \alpha\cdot C1 + C2 + \alpha'\cdot C3 \tag{9}$$

From this comparison we could clearly observe the following two key differences of above two filtering schemes:

Table 2
The numerical evaluation on Middlebury benchmark (error threshold is set to be 1 pixel). The numbers support the authors' claim: the proposed method is currently the state-of-art on both accuracy and speed. It is the fastest local stereo method on GPU. And at the time of submission (October, 2013), it ranks 21st out of over 150 methods (outperforms other state-of-art local methods).

Methods	Avg. Err (%)	GPU runtime
Edge-aware disparity prorogation (exact our method)	5.23	9 ms
Our method but excluding the term $R(\mathbf{p}, d)$ in Eq. (1)	5.74	9 ms
Proposed framework with domain transform filter	6.33	10 ms
Proposed framework with recursive bilateral filter [20]	5.57	11 ms
Hardware-efficient bilateral filter agg. + post-refinement [21]	5.41	15 ms
Box filter agg. + weighted median filter [7]	6.19	22 ms
Guided image filter agg. + weighted median filter [13]	5.55	65 ms
Non-local filter agg. + non-local refinement [19]	5.48	NA

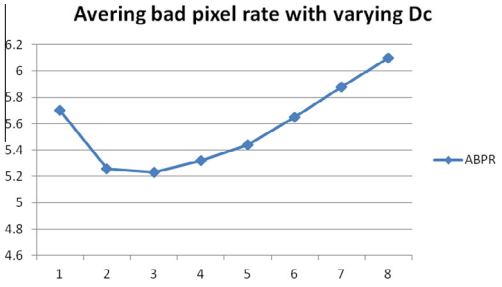


Fig. 4. Average bad pixel rates for our method with varying D_c.

Table 3
The sub-pixel accuracy evaluation on Middlebury benchmark (error threshold is set to be 0.5 pixel). At the time of submission (October, 2013), it ranks 8th out of over 150 methods. The ranking result is very impressive especially when considering the runtime of our approach is less than 10 ms.

Methods	Avg.Err (%)
Our approach (sub-pixel enhancement)	9.80
PatchMatch [1]	9.91
SubPixDoubleBP [24]	10.7

1. The recursive filter is *NOT balanced on the supports from opposite directions*. That is to say, as it can be seen from Eq. (8), even if $\alpha = \alpha', C1$ and $C3$ have different weights for $C2'$. On the contrary, our proposed filter is free from this directional bias.
2. In recursive filter, the support from $C1$ to the $C2'$ is determined by not only α but also α'. In stereo matching, this property is not desirable since the supporting weight is not accurate (**p3** is not on the path from **p1** to **p2**).

3. Experimental results

In this section, our algorithm is evaluated and compared with other state-of-art methods. The testing platform is a PC equipped with a 3.0 GHz Intel i5 CPU, 8 GB of memory and a Geforce GTX 580 graphics card. The GPU implementation is written in CUDA. In all experiments, we use the *same* parameter settings: $\{D_c, \lambda_c, \sigma_s, \sigma_r\} = \{3, 0.2, 42.5, 22.5\}$.

First of all, our approach is qualitatively evaluated using the Middlebury standard data sets [15] with accurate ground truth. The numerical comparisons (error threshold: 1 pixel) are provided in Table 2. The numbers for other methods are reported by original papers or produced by publicly available source code. Non-local filter [19] is not GPU-friendly, and the speed of its CPU implementation is not comparable to those GPU implementations of state-of-arts. On the standard data sets, the averaging running time of cost volume initialization, pre-detecting disparity priors, and disparity propagation is 1.1 ms, 1.9 ms and 5.9 ms, respectively. The qualitative evaluation shows that the proposed method outperforms other methods both in terms of accuracy and speed. It runs 6x faster than previous leading method guided image filter [13]. And even when comparing to the fastest method in previous literatures [21], our method reduces 50% time consumption and produces lower average error rates. Furthermore, if the same framework as proposed in this letter is adopted, except that the proposed geodesic filter is replaced by recursive filter [4,20], the results are much less competitive (parameters already tuned for lowest bad pixel rate). These results support the authors' claim, that the proposed geodesic filter outperforms recursive filters on stereo matching. Besides, we also studied the effect of varying D_c in our designed cost function, as shown in Fig. 4. Interestingly, it

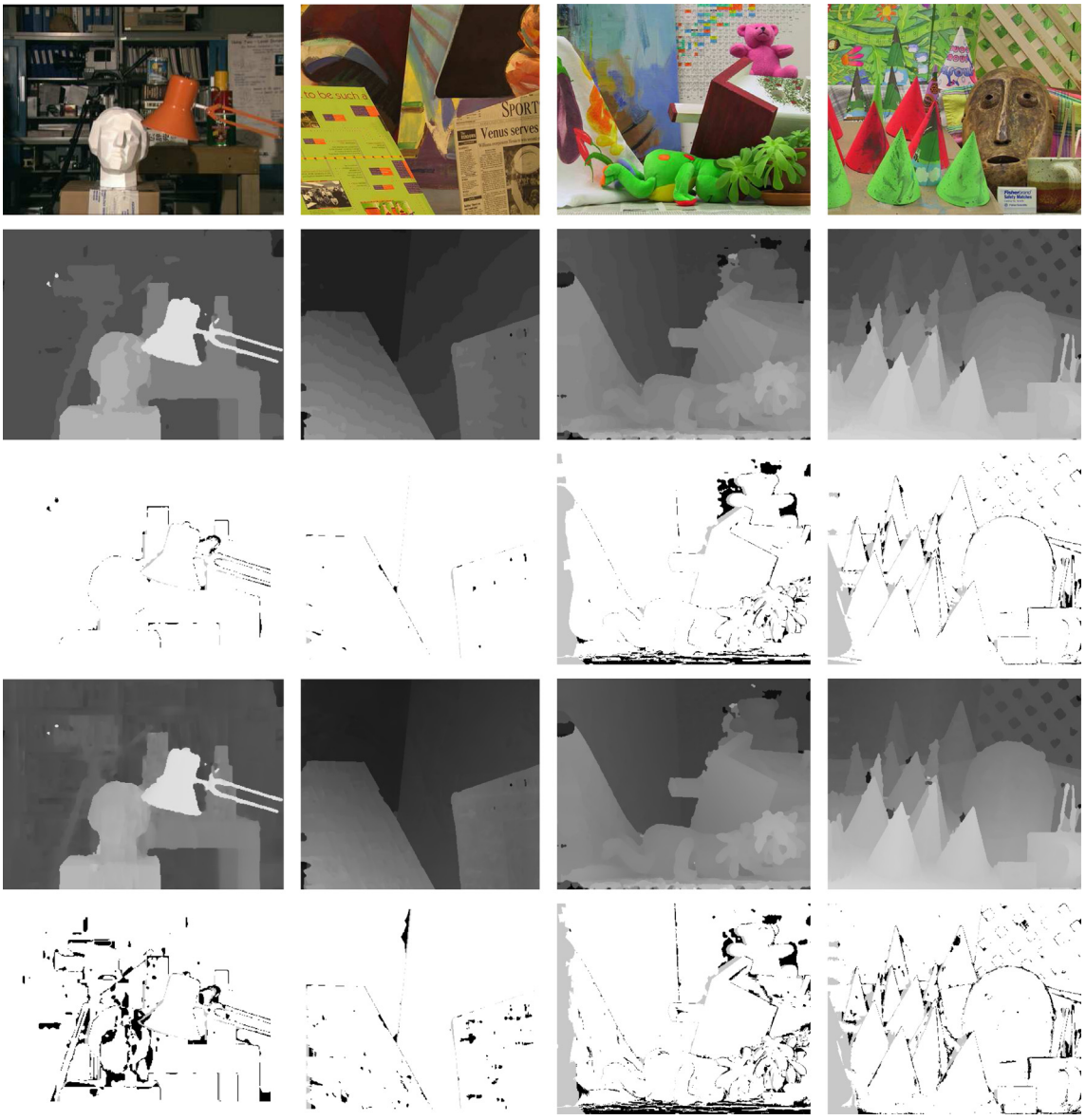
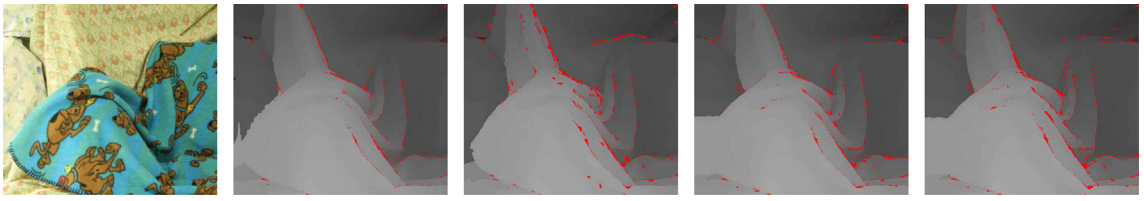


Fig. 5. Results on the Middlebury standard data set. The first row: reference views. From second row to third row: disparity results by our integer-valued algorithm and error maps with 1 pixel as threshold. From fourth row to fifth row: our disparity results by sub-pixel enhancement and corresponding error maps with 0.5 pixel as threshold. Errors in un-occluded and occluded areas are marked in black and gray, respectively.



(a) Reference view (b) Our method, 1.04% bad (c) Non-local filter, 2.16% bad (d) Box+WMF, 1.60% bad (e) GF+WMF, 2.04% bad

Fig. 6. Result demonstrating the effectiveness of our method on Middlebury *Cloth*3 data set which are with highly curved surface. The bad pixels in non-occluded areas are marked in red. This figure is best viewed in color. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

is observed that only a compact sub-set (with small D_c) would benefit the accuracy. In addition to the integer-valued stereo, with an exact quadric cost function on d designed in Eq. (1), our algorithm intrinsically well-suits the quadratic polynomial interpolation [23]. By fitting a parabola in the discrete cost function, the *exact* sub-pixel values corresponding to the minimal costs are determined. Surprisingly, our sub-pixel method ranks 8th on Middlebury benchmark when 0.5 pixel error threshold is considered (Table 3). Note that this result is better than some recent specifically-designed sub-pixel methods such as [1], while our method runs over 100× faster. The disparity results are presented in Fig. 5.

Next, our method is evaluated on a scene with highly curved surfaces. As shown in Fig. 6, it well-propagates the disparities even on curved surfaces. Here the non-local filter [19] shows its limit on

Fig. 7. Visual comparison using the Book Arrival real-world stereo video [10]. The first column presents the reference view of 20th, 30th and 70th frames. And the next three columns show the results produced by [13,7] and our approach respectively. No temporal information is used for each method. Details are best viewed in electronic version.

complex and detailed structures (due to the information lost in its tree construction step).

Finally, our method is visually compared with the [13,7] (another recent state-of-art method mostly related to our method). The test is performed on Book Arrival sequence [10], a real-world stereo video clip which is captured in an un-controlled environment. The disparity results of 20th, 30th and 70th frames obtained from our approach and comparing methods are presented in Fig. 7. For results from [13], the exact weighted median filter from [7] is used for post-processing. As can be seen, the proposed approach clearly outperforms other two representative state-of-arts. It performs reasonably well on texture-less regions even without any parameter tuning. At the same time, the drawbacks of [7] are shown in this real-world example: as we have stated before, when the simple box aggregation produces a extremely noisy disparity map, weighted median filter applied on the low-quality raw results would not resolve the matching ambiguities (see the blur disparity map results computed by [7]).

4. Conclusion

In this letter, the authors have tried to shed some light on developing a real-time local stereo method. The proposed method is based on a novel framework which re-solves the stereo problem by propagating disparities in the cost domain. It is the current top performer in local methods with leading accuracy. On a NVIDIA GeForce GTX 580 GPU, it can process Middlebury standard data sets 110 frames per second. As a future work, we would like to port our algorithm to an embedded system and build a solution for practical applications.

Acknowledgement

Xing Mei is funded by National Natural Science Foundation of China (Grant Nos. 61271430, 61172104, and 61332017).

Appendix A. Supplementary data

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.patrec.2014.07.010.

References

[1] M. Bleyer, C. Rhemann, C. Rother, Patchmatch stereo – stereo matching with slanted support windows, in: Proc. BMVC, 2011.
[2] Y. Boykov, O. Veksler, R. Zabih, Fast approximate energy minimization via graph cuts, PAMI 23 (2001) 1222–1239.
[3] P.F. Felzenszwalb, D.P. Huttenlocher, Efficient belief propagation for early vision, IJCV 70 (2006) 41–54.
[4] E.S.L. Gastal, M.M. Oliverira, Domain transform for edge-aware image and video processing, ACM TOG, 2011, pp. 1–12.
[5] K. He, J. Sun, X. Tang, Guided image filtering, PAMI, 2013, pp. 1397–1409.
[6] A. Hosni, M. Bleyer, M. Gelautz, Secrets of adaptive support weight techniques for local stereo matching, CVIU, 2013, pp. 620–632.
[7] Z. Ma, K. He, Y. Chen, J. Sun, E. Wu, Constant time weighted median filtering for stereo matching and beyond, in: Proc. ICCV, 2013.
[8] X. Mei, X. Sun, W. Dong, H. Wang, X. Zhang, Segment-tree based cost aggregation for stereo matching, in: Proc. CVPR, 2013, pp. 313–320.
[9] D. Min, J. Lu, M. Do, A revisit to cost aggregation in stereo matching: how far can we reduce its computational redundancy? in: Proc. ICCV, 2011.
[10] MOBILE3DTV, The book arrival data sets, <http://sp.cs.tut.fi/mobile3dtv/stereo-video/>.
[11] S. Paris, P. Kornprobst, J. Tumblin, F. Durand, Bilateral filtering: theory and applications, Found. Trends Comput. Graphics Vision 4 (2008) 1–73.
[12] C. Pham, J.W. Jeon, Domain transformation-based efficient cost aggregation for local stereo matching, CSVT (2012).
[13] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, M. Gelautz, Fast cost-volume filtering for visual correspondence and beyond, in: Proc. CVPR, 2011, pp. 3017–3024.
[14] D. Scharstein, R. Szeliski, A taxonomy and evaluation of dense two-frame stereo correspondence algorithms, IJCV 47 (2002) 7–42.
[15] D. Scharstein, R. Szeliski, Middlebury stereo evaluation – version 2, 2010, <http://vision.middlebury.edu/stereo/eval/>.
[16] C. Tomasi, R. Manduchi, Bilateral filtering for gray and color images, in: Proc. ICCV, 1998.
[17] L. Wang, R. Yang, Global stereo matching leveraged by sparse ground control points, in: Proc. CVPR, 2011.
[18] Q. Yang, D. Li, L. Wang, M. Zhang, Full-image guided filtering for fast stereo matching, SPL 20 (2013) 237–240.
[19] Q. Yang, A non-local cost aggregation method for stereo matching, in: Proc. CVPR, 2012, pp. 1402–1409.
[20] Q. Yang, Recursive bilateral filtering, Proc. ECCV (2012) 399–413.
[21] Q. Yang, Hardware-efficient bilateral filtering for stereo matching, PAMI (2013) 1026–1032.
[22] Q. Yang, K.H. Tan, N. Ahuja, Real-time $O(1)$ bilateral filtering, in: Proc. CVPR, 2009.
[23] Q. Yang, L. Wang, R. Yang, H. Stewénius, D. Nistér, Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling, PAMI 31 (2009) 492–504.
[24] Q. Yang, R. Yang, J. Davis, D. Nister, Spatial-depth super resolution for range images, in: Proc. CVPR, 2007.
[25] K.J. Yoon, I.S. Kweon, Adaptive support-weight approach for correspondence search, PAMI 28 (2006) 650–656.