Stereo Matching with Reliable Disparity Propagation

Xun Sun, Xing Mei, Shaohui Jiao, Mingcai Zhou, Haitao Wang
Samsung Advanced Institute of Technology, China Lab
Beijing, China
xunshine.sun,xing.mei,sh.jiao,mingcai.zhou,ht.wang@samsung.com

***Abstract*—In this paper, we propose a novel propagation-based stereo matching algorithm. Starting from an initial disparity map, our algorithm selects highly reliable pixels and propagates their disparities along the scanline to produce dense disparity results. The key idea is to construct a line segment region for each pixel with local color and connectivity constraints. The pixelwise line segments are efficiently used to compute initial disparities, select reliable pixels and determine proper propagation regions. Streaking artifacts are effectively removed in a refinement process. Experimental results demonstrate the performance of the proposed method: it ranks 5th in the Middlebury benchmark, and the results can be computed within a few seconds.**

***Keywords*-stereo matching; disparity propagation; line segments;**

I. Introduction

Dense two-frame stereo matching is one of the most actively studied topics in computer vision. Most stereo algorithms can be roughly classified into two categories: *local* and *global* methods [1].

Local methods compute each pixel's disparity independently over a support region. The matching costs are aggregated over the region, and the disparity level with the minimal cost is selected as the output of the pixel. The performance of the local methods mainly depends on the support windows they choose: square windows with simple averaging lead to blurred low-quality results in real-time [2], while some recent techniques such as adaptive weight [3], [4] and segment support [5] can produce accurate disparity maps with considerable running time.

Global methods formulate the stereo problem as an energy function with data and smoothness constraints, and minimize this function with dynamic programming [6], graph cuts [7] or belief propagation [8]. Global optimizers can greatly suppress the matching ambiguities caused by various factors such as illumination variation and textureless regions, and therefore produce more accurate results than common local methods. And they are usually computationally expensive due to the slow-converging optimization process. Most state-of-the-art algorithms in the Middlebury benchmark [9] fall into this category.

Our algorithm is mainly inspired by a class of methods which produce quasi-dense or dense disparity results from a set of seed pixels [10], [11], [12], [13]. These methods first find reliable seed pixels with simple local or sparse matching techniques, and then propagate to neighboring pixels in a growing-like way. If no global optimization is involved in the computation process, the propagation-based methods usually show low computational complexity. However, an inherent problem with these methods is that early wrong matches might lead to large disparity errors during the propagation process. Wei and Quan [14] proposed to use segmented regions as the propagation units, which greatly alleviates the problem. The propagation ranges of the reliable pixels can be well controlled with the segmented regions, but the running time of the algorithm is dominated by the expensive segmentation process.

In this paper, we propose a novel stereo matching algorithm based on pixelwise line segments and 1D propagation. Different from previous methods [12], [14], our algorithm only propagates the disparities of the seed pixels along the scanline directions. This 1D restriction allows us to control the propagation process with simple lines instead of complex segmented regions. In practice, the line segments are efficiently constructed for each pixel with color similarity and connectivity constraints. These pixelwise line segments serve not only as the supported regions for initial disparity computation, but also as the reliable regions for accurate disparity propagation. Finally, by employing vertical voting and disparity filtering, our algorithm solves the typical inter-scanline inconsistency problem due to the 1D propagation. Despite its simplicity, our algorithm shows good performance in both speed and accuracy, which is attractive for real-world stereo applications.

The rest of the paper is organized as follows: Section II provides a full description of the algorithm. Section III reports the experimental results. And conclusion is given in Section IV.

II. Algorithm

The framework of our algorithm is shown in Figure 1. First, the pixelwise line segments are constructed (Section II-A). Second, these line segments are used to compute the initial disparity map (Section II-B). Third, the seed pixels with highly reliable disparities are detected with consistency check and confidence evaluation (Section II-C). Fourth, the dense disparity map is generated with scanline propagation and seed pixels (Section II-D). Finally, the artifacts in the

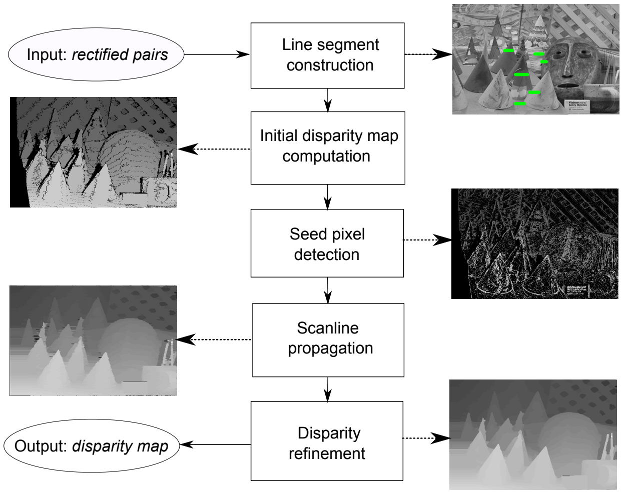


Figure 1. Block diagram of our algorithm with input data, intermediate and final output results.

disparity map are removed in a refinement process (Section II-E). In the following, we present a detailed description of these individual steps.

A. Line Segment Construction

As stated in the introduction, a line segment along the scanline is dynamically built on each pixel of the stereo image pair, which is used as the support region as well as the reliable propagation region. For accurate matching and propagation, the disparities should vary smoothly in each line segment. A simple but effective assumption on support regions is that neighboring pixels with similar colors should have similar disparities (color-disparity consistency), which has been extensively adapted by recent local methods [4], [5]. We also follow this assumption when building the line segments.

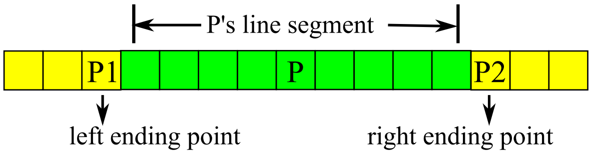


Figure 2. The constructed line segment for pixel $\mathbf{p}$.

For a given pixel $\mathbf{p}$, its line segment construction process is illustrated in Figure 2. Starting from $\mathbf{p}$, the left arm of the segment stops at an ending pixel $\mathbf{p_1}$ that breaks one of the two following rules:

1) $D_c(\mathbf{p_1}, \mathbf{p}) < \tau$, where $D_c(\mathbf{p_1}, \mathbf{p})$ is the color difference between $\mathbf{p_1}$ and $\mathbf{p}$, and τ is a preset threshold value. The color difference is defined as $D_c(\mathbf{p_1}, \mathbf{p}) = \max_{i=R,G,B} |I_i(\mathbf{p_1}) - I_i(\mathbf{p})|$.
2) $D_s(\mathbf{p_1}, \mathbf{p}) < L$, where $D_s(\mathbf{p_1}, \mathbf{p})$ is the spatial distance between $\mathbf{p_1}$ and $\mathbf{p}$, and L is a preset maximum length (measured in pixels). The spatial distance is defined as $D_s(\mathbf{p_1}, \mathbf{p}) = |\mathbf{p_1} - \mathbf{p}|$.

The first rule guarantees the color similarity in the line region, and the second rule poses a limitation on the line length to avoid over-smoothed disparity results and incorrect propagation. The right ending point $\mathbf{p_2}$ can also be determined by the two rules. And $\mathbf{p}$'s line segment is recorded by a pair of length values a_l, a_r:

$$a_l = D_s(\mathbf{p_1}, \mathbf{p}) - 1, a_r = D_s(\mathbf{p_2}, \mathbf{p}) - 1 \qquad (1)$$

Note that we have constructed a *horizontal* line segment for pixel $\mathbf{p}$. If a *vertical* line segment is built in a similar way, we get a cross-shaped skeleton. This skeleton can be used to define a dynamic 2D support region over pixel $\mathbf{p}$, which was first proposed by Zhang *et al* [15]. Compared to 2D support windows [3], [4], [5], [15], our line segments consider only the neighboring pixels in the scanline direction, but this simple structure still provides enough information for disparity computation and propagation with relatively low computation cost, as shown in later sections.

B. Initial Disparity Computation

This step computes the initial disparity results with a simple local method. Given a pixel $\mathbf{p} = (x, y)$ in the left image and a disparity level d, the matching cost $C_1(\mathbf{p}, d)$ is first computed by linearly combining two simple measures:

$$\begin{aligned} C_1(\mathbf{p}, d) = & \min(C_{AD}(\mathbf{p}, d), \lambda_{AD}) \\ & + \min(C_{census}(\mathbf{p}, d), \lambda_{census}) \end{aligned} \qquad (2)$$

where C_{AD} is the RGB color difference of pixel $\mathbf{p}$ and its correspondence $\mathbf{pd} = (x - d, y)$ in the right image:

$$C_{AD}(\mathbf{p}, d) = \sum_{i=R,G,B} |I_i^{Left}(\mathbf{p}) - I_i^{Right}(\mathbf{pd})| \qquad (3)$$

C_{census} is a cost measure that encodes local image structures with relative orderings of the pixel intensities [16]. It is defined as the Hamming distance of the two 64-bit strings that stand for $\mathbf{p}$ and $\mathbf{pd}$ respectively. The strings are encoded with a 9×7 census transform. λ_{AD} and λ_{census} are threshold parameters to reject outliers. The reason why we employ Equation (2) for matching cost computation is twofold: first, we experimentally find that the combined AD-Census measure shows improved matching accuracy than individual AD and census measures; second, C_{census} introduces the structure information of a 2D window around each pixel into the cost, which helps to alleviate the computation errors due to the 1D line structures.

Next, we employ an aggregation step to reduce the matching ambiguities in the original matching cost volume. The aggregated cost value $C_2(\mathbf{p}, d)$ is computed over $\mathbf{p}$'s line

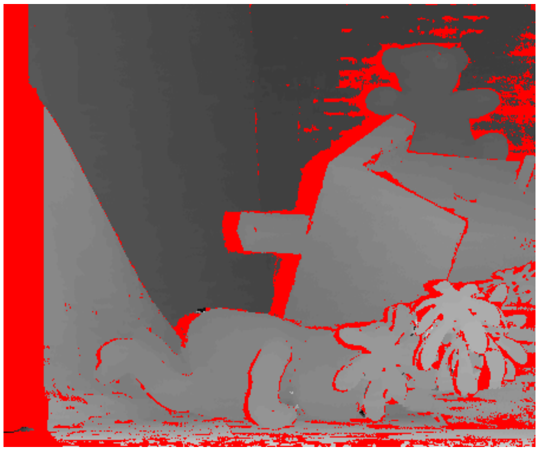


Figure 3. The initial disparity map for the Teddy image pair. Erroneous disparities are marked in red.

segment:

$$C_2(\mathbf{p}, d) = \frac{\sum_{\mathbf{q} \in Line(\mathbf{p})} C_1(\mathbf{q}, d)}{|Line(\mathbf{p})|} \tag{4}$$

For robust aggregation, this step is iterated for 2 times. Finally, the output disparity for pixel $\mathbf{p}$ is simply computed with the aggregated cost values C_2 and a winner-take-all searching strategy:

$$D_L(\mathbf{p}) = \arg\min_d C_2(\mathbf{p}, d) \tag{5}$$

In a similar way, the disparity map D_R for the right image is computed. The initial disparity map D_L for the Teddy image pair is presented in Figure 3, which shows significant errors (marked in red) in occlusion regions, textureless regions and around discontinuities. We tackle these errors with propagation and refinement steps.

C. Seed Pixel Detection

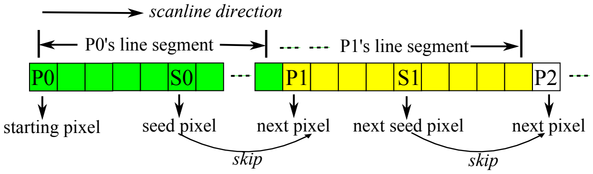


Figure 4. The seed pixel detection process along the scanline.

This step detects the highly reliable seed pixels (also called as *ground control points*) from the initial disparity map, which serve as a good starting point for the propagation process. We propose to perform a sequential search along each scanline of the map, as shown in Figure 4. Starting from the leftmost pixel $\mathbf{p_0}$ of the scanline, we search for a seed pixel $\mathbf{s_0}$. $\mathbf{s_0}$ is detected as a 'seed' pixel if the following conditions are satisfied:

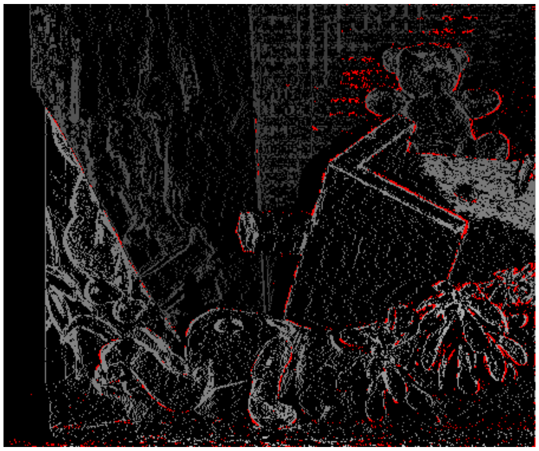


Figure 5. The detected seed pixels for the Teddy disparity map. Erroneous disparities are marked in red.

1) $D_L(\mathbf{s_0}) = D_R(\mathbf{s_0} - (D_L(\mathbf{s_0}), 0))$
2) $\dfrac{C_2(\mathbf{s_0}, d)}{C_2(\mathbf{s_0}, D_L(\mathbf{s_0}))} > \lambda_c, \forall d \neq D_L(\mathbf{s_0})$

Rule 1 requires $\mathbf{s_0}$ to pass the left-right consistency check, while rule 2 states that pixel $\mathbf{s_0}$'s initial disparity is reliable if its matching cost is significantly smaller than the other competitors. And λ_c is a threshold value for evaluating the disparity reliability. When $\mathbf{s_0}$ is found, it is not necessary to detect more seed pixels in $\mathbf{p_0}$'s line segment due to the smooth disparity assumption. The searching process skips some pixels, moves to pixel $\mathbf{p_1}$ (the end of $\mathbf{p_0}$'s line segment) and looks for the next seed pixel $\mathbf{s_1}$, until it reaches the end of the scanline.

The seed pixels detected from Teddy's initial disparity map are presented in Figure 5. Compared to the initial disparity map (Figure 3), the erroneous disparities (marked in red) in the seed pixels are greatly reduced, which demonstrates the effectiveness of the evaluation conditions. Another advantage of our detection method is that a large part of the seed pixels are well distributed around depth discontinuities, and some potential seed pixels inside the homogeneous regions are skipped with the line segment information. These pixels can help preserve the depth discontinuities during the propagation process.

D. Scanline Propagation

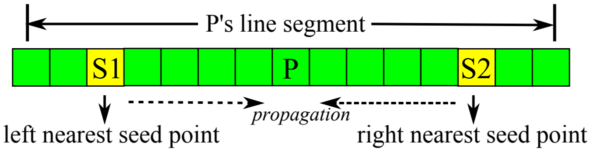


Figure 6. The scanline propagation process.

This step propagates the disparities of the seed pixels along the scanline direction, as shown in Figure 6. For an 'unseeded' pixel $\mathbf{p}$, we first search the left and right nearest seed pixels ($\mathbf{s_1}$, $\mathbf{s_2}$) within its line segment. If only one seed pixel is found in the segment, $D_L(\mathbf{p})$ is simply replaced by $D_L(\mathbf{s_1})$ or $D_L(\mathbf{s_2})$ since no more information can be used for reliable disparity estimation. If both seed pixels are available, the propagation strategy proceeds as follows:

1) $D_L(\mathbf{p}) = \min(D_L(\mathbf{s_1}), D_L(\mathbf{s_2}))$, if $\mathbf{p}$ is in occluded regions. $\mathbf{p}$ is detected as 'occluded' if it fails in the left-right consistency check.
2) $D_L(\mathbf{p}) = \min(D_L(\mathbf{s_1}), D_L(\mathbf{s_2}))$, if $\mathbf{p}$ is near depth discontinuities. A depth discontinuity is detected if $|D_L(\mathbf{s_1}) - D_L(\mathbf{s_2})| > \alpha \cdot d_{max}$. d_{max} is the maximum disparity level.
3) $D_L(\mathbf{p})$ is replaced by the linearly interpolated results of $D_L(\mathbf{s_1})$ and $D_L(\mathbf{s_2})$ in all other conditions.

Once pixel $\mathbf{p}$ is updated, it is immediately labelled as a 'seed' pixel for the subsequent pixels. If no seed pixels are found in $\mathbf{p}$'s line segment, $\mathbf{p}$ is left unchanged. After the scanline propagation process, the remaining 'unseeded' pixels are updated with the minimal disparity values of the nearest seed pixels.

This propagation process share some similarities with the traditional dynamic-programming based algorithms [10], [11], [17], since they all work along scanline directions. However, there are some fundamental differences between the two methods: first, most DP-based methods operate on the full matching cost volumes, while our propagation process works only in the image space which leads to much lower computation cost; second, for DP algorithms, the color-disparity consistency assumption is usually added to the energy function as a soft constraint, while for propagation, it is represented with line segments, and then used as a hard constraint to assign disparity values directly to the 'unseeded' pixels. For image regions where this assumption doesn't hold due to image noise or large occlusion, the propagation process might not be as stable as DP algorithms.

The updated Teddy disparity map is presented in Figure 7, which shows great improvements over the initial disparity results in occlusion regions and around depth discontinuities. A problem with our scanline propagation is that the line segments can lead to visible streaking artifacts, which need to be solved with a refinement process.

E. Disparity Refinement

This step refines the disparity map with a two-step process.

Vertical Voting: Since the disparity computation relies on horizontal line segments, the streaking artifacts can be reduced by including more disparity information from the vertical direction. For each pixel $\mathbf{p}$, a histogram $H_\mathbf{p}$ with $d_{max}+1$ bins (d_{max} is the maximum disparity level) is built for a voting scheme. We collect N disparity votes in a vertical line segment that starts at pixel $\mathbf{p} - (0, N/2)$ and ends at pixel $\mathbf{p} + (0, N/2)$: for any pixel $\mathbf{q}$ in the line, if the color difference $D_c(\mathbf{q}, \mathbf{p})$ is smaller than the threshold value τ, the votes on disparity $D_L(\mathbf{q})$ increase by 1. Color similarity is considered in the voting process. And $D_L(\mathbf{p})$ is updated with the final voting results:

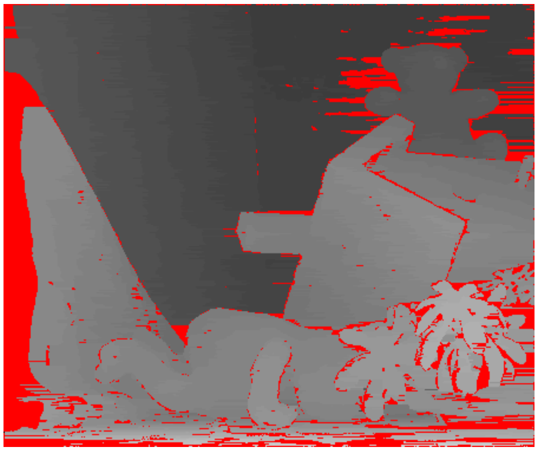


Figure 7. The Teddy disparity map after the propagation process.

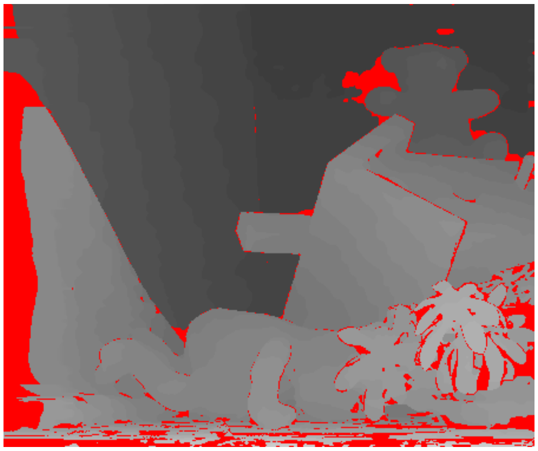


Figure 8. The Teddy disparity map after the refinement process.

$$D_L(\mathbf{p}) = \arg\max_d H_\mathbf{p}(d) \tag{6}$$

After the 1D voting process, most horizontal streaking artifacts in large areas can be effectively alleviated. We further employ a 4-neighbor update method from [18] to remove small scale artifacts and outliers.

4-Neighbor Update with Bilateral Filtering: For pixel $\mathbf{p}$, the disparities from its four neighbors are collected $D_\mathbf{p} = \{D_L(\mathbf{p} \pm (1,0)), D_L(\mathbf{p} \pm (0,1))\}$. Then $D_L(\mathbf{p})$ is updated as follows:

$$D_L(\mathbf{p}) = \arg\min_{d \in D_\mathbf{p}} \frac{\sum_{\mathbf{q} \in W_\mathbf{p}} f(\mathbf{q}, \mathbf{p}) C_3(d, D_L(\mathbf{q}))}{\sum_{\mathbf{q} \in W_\mathbf{p}} f(\mathbf{q}, \mathbf{p})} \tag{7}$$

where $\mathbf{q}$ is a pixel in a square window $W_{\mathbf{p}}$ over $\mathbf{p}$, $f(\mathbf{q},\mathbf{p})$ is a bilateral weight:

$$f(\mathbf{q},\mathbf{p}) = exp(-\frac{D_c(\mathbf{q},\mathbf{p})}{\sigma_c}) \cdot exp(-\frac{D_s(\mathbf{q},\mathbf{p})}{\sigma_s}) \qquad (8)$$

and $C_3(d, D_L(\mathbf{q})) = \min(\beta \cdot d_{max}, |d - D_L(\mathbf{q})|)$. In practice, the size of $W_{\mathbf{p}}$ is set to 11×11 for all the data sets. The updated $D_L(\mathbf{p})$ value is directly used for the computation of the subsequential pixels. This method further enhances the inter-scanline consistency by propagating good matches in a 4-connected neighborhood. The refined Teddy disparity map is presented in Figure 8, which shows that most streaking artifacts and outliers are effectively removed with the two refinement steps.

III. Experimental Results

We test our algorithm with the Middlebury benchmark [9]. The test platform is a PC with Core2Duo 3.0GHz CPU and 2GB memory. The parameters are presented in Table I, which are kept constant for all the data sets.

Table I
Parameter settings for the Middlebury data sets

L	τ	λ_{AD}	λ_{census}	λ_c
17	20	60	20	1.1
N	σ_s	σ_c	α	β
16	4	2.5	0.2	0.2

The intermediate and final disparity results are presented in Figure 9, which demonstrates the effectiveness of the algorithm: various errors are gradually removed from the initial results with seed pixel detection, scanline propagation and proper refinement. With pixelwise line segments, our algorithm performs well in textureless regions and around depth discontinuities. The quantitative results are presented in Table II. Satisfactory disparity results are achieved for all the data sets. At the time of submission (December 2011), our method ranks 5th in the Middlebury data sets, which is impressive when considering that no global optimization is involved in the computation process.

Another advantage of our algorithm is that we do not perform any segmentation operations. Most computations are performed in each pixel's line segment, which brings relatively low computation costs. The running time for the four data sets (Tsukuba, Venus, Teddy and Cones) are 2.5 seconds, 3.8 seconds, 8.7 seconds and 8.6 seconds respectively, which is competitive among the top 10 Middlebury algorithms. The most time-consuming part of the algorithm is the initial disparity computation step, which usually takes about 50% of the running time.

We further test our algorithm with more examples in the Middlebury 2005, 2006 data sets [19], [20], and some results are presented in Figure 10. To verify the robustness of the algorithm, we still employ the parameter settings in Table I. Without any parameter tuning, our algorithm shows quite stable performance for these image pairs, and produces satisfactory disparity results.

Finally, based on the erroneous regions shown in Figure 9 and 10, we briefly discuss some limitations and possible improvements of the algorithm:

First, our algorithm relies on the line segments, which are constructed with local color and connectivity constraints. For most Middlebury data sets with ideal image data, these regions can be constructed with high accuracy, but for practical indoor and outdoor scenes, the image quality might be severely brought down by noise and illumination variation, which leads to incorrect line construction and large computation errors (Figure 10 (e)). Bilateral filtering can suppress image noise without destroying local structures and edges [21], which can used as a pre-process before the line construction step;

Second, our algorithm performs the computation along scanline directions. For some regions with large occlusion, there might no reliable pixels in the 1D neighborhood such that wrong disparities are propagated into these regions, as shown in Figure 10 (a)(c)(d). A possible solution, inspired by Hirschmüller's semi-global matching method [22], is to propagate in multiple 1D directions, and select the most reliable disparity candidate for the occluded pixel.

Finally, the propagation process operated in the image space, and no refined cost volumes are produced. Therefore, effective sub-pixel interpolation techniques [1], [23] are not available to our algorithm.

IV. Conclusions

This paper has presented a simple and effective stereo matching algorithm. The algorithm detects seed pixels with reliable disparities and propagates them to produce the disparity results. The accuracy of the propagation process can be controlled with pixelwise line segments, which shows good performance in both speed and accuracy. As a future work, we would like to apply our algorithm to real world problems. And we plan to port the algorithm to graphics hardware for performance acceleration.

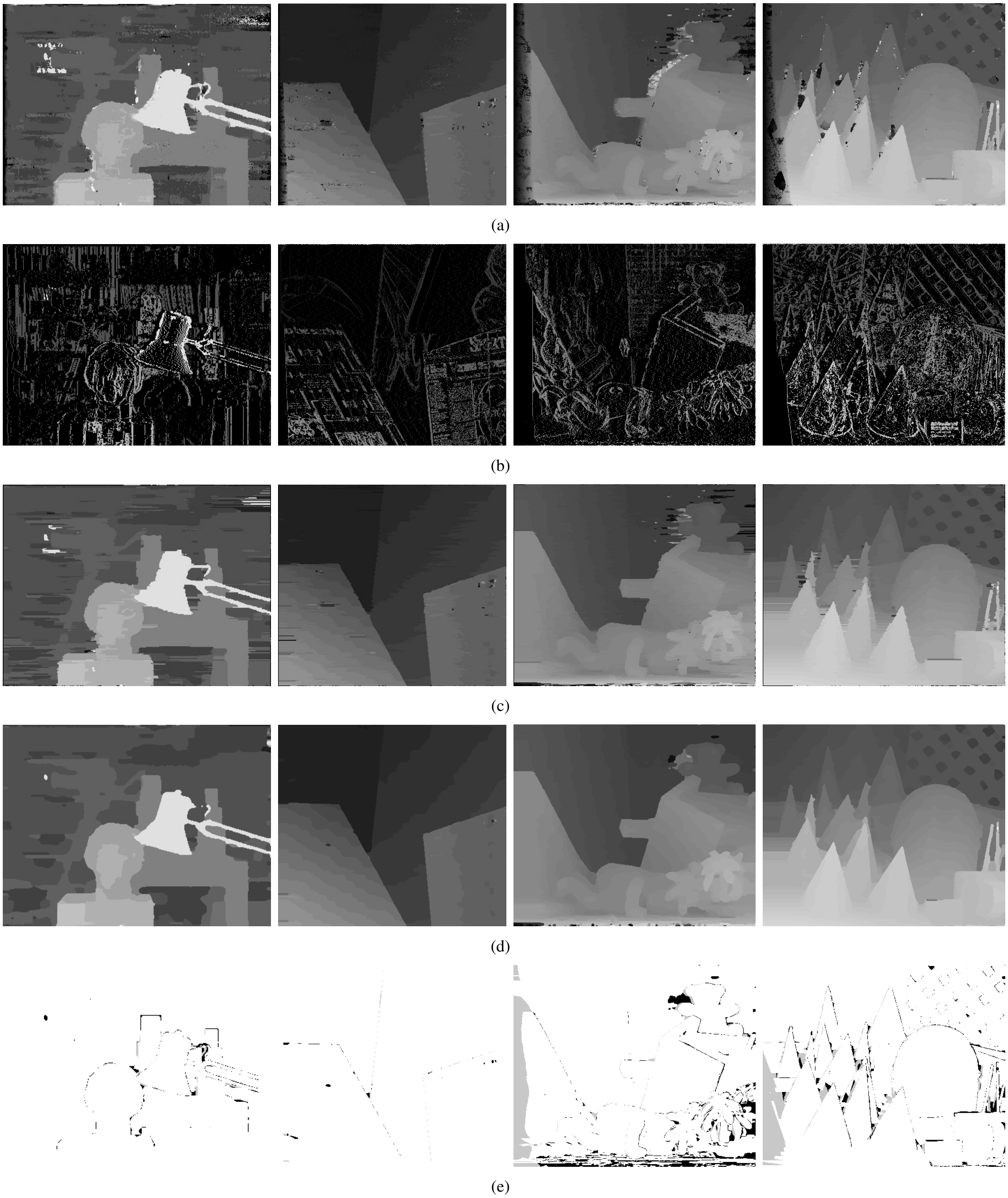
Acknowledgment

We would like to thank Daniel Scharstein and Richard Szeliski for providing the stereo image data and the test bed.

References

[1] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *IJCV*, vol. 47, no. 1-3, pp. 7–42, 2002.

[2] T. Kanade, "Development of a video-rate stereo machine," in *Image Understanding Workshop*, 1994, pp. 549–557.



(a)

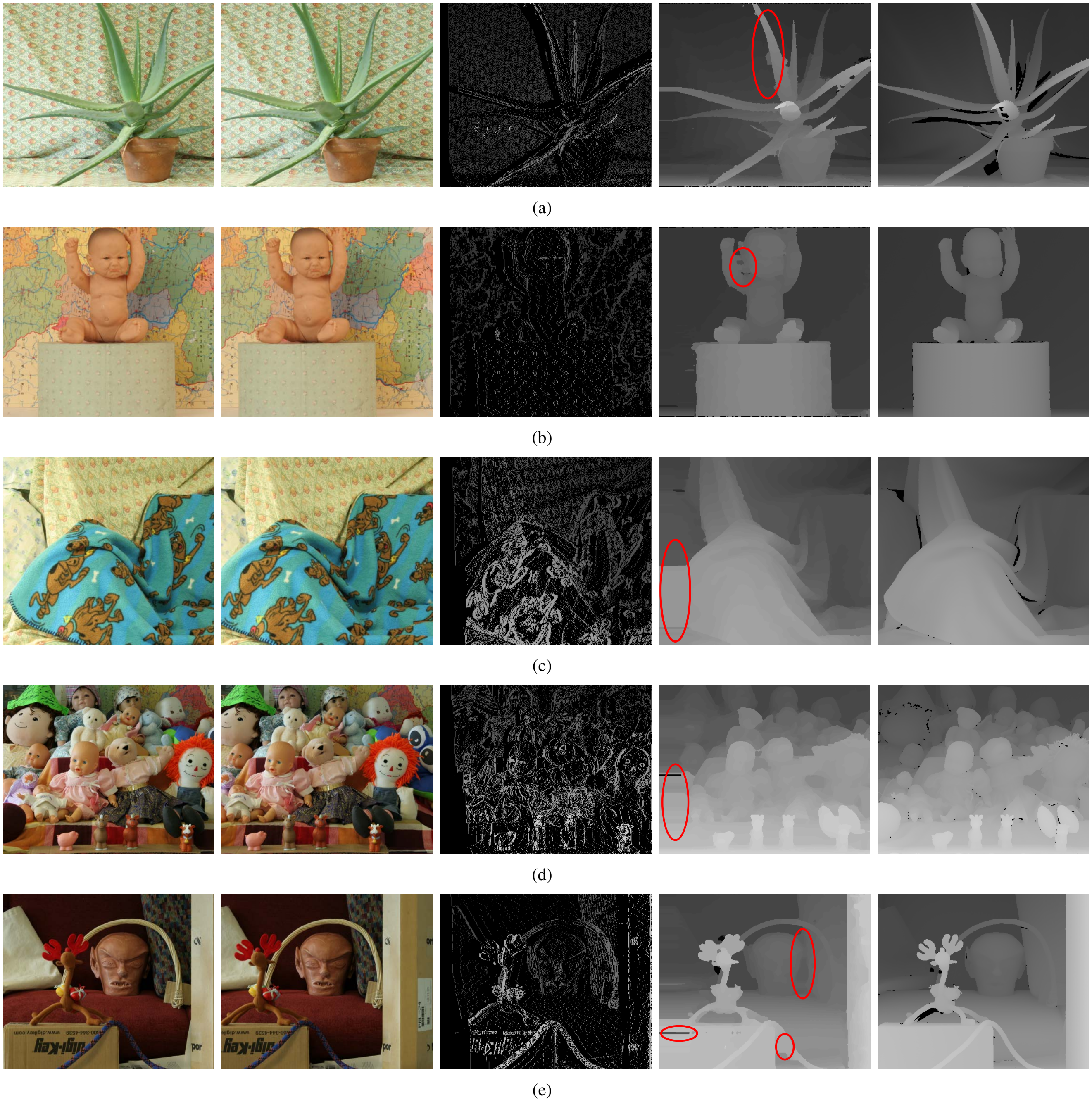
(b)

(c)

(d)

(e)

Figure 9. Results on the Middlebury data set. (a) Initial disparity maps. (b) Detected seed points. (c) Scanline propagation results. (d) Refinement disparity maps. (e) Error maps under pixel error threshold 1. Errors in unoccluded and occluded regions are marked in black and gray respectively.



(a)

(b)

(c)

(d)

(e)

Figure 10. More results from the Middlebury 2005 and 2006 data sets [19], [20]: (a) Aloe. (b) Baby. (c) Cloth. (d) Dolls. (e) Reindeer. For each row (from left to right): left image, right image, detected seed points, final disparity map and groundtruth data. We still use the parameter setting in Table I for these images. Some erroneous regions in the disparity maps are marked with red ellipses.

Table II

THE RANKINGS IN THE MIDDLEBURY BENCHMARK. THE ERROR PERCENTAGES IN DIFFERENT REGIONS (*nonocc, all ,disc*) ARE PRESENTED.

Algorithm	Avg. Rank	Tsukuba			Venus			Teddy			Cones		
		nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc
CoopRegion [24]	6.6	0.87_1	1.16_1	4.61_1	0.11_3	0.21_2	1.54_5	5.16_{14}	8.31_9	13.0_{11}	2.79_{12}	7.18_4	8.01_{16}
DoubleBP [23]	8.8	0.88_3	1.29_2	4.76_3	0.13_6	0.45_{16}	1.87_{10}	3.53_3	8.30_8	9.63_2	2.90_{16}	8.78_{23}	7.79_{13}
Our method	9.4	0.97_6	1.39_6	5.00_5	0.21_{19}	0.38_{14}	1.89_{11}	4.84_8	9.94_{15}	12.6_9	2.53_5	7.69_7	7.38_8
OutlierConfi [25]	9.7	0.88_2	1.43_8	4.74_2	0.18_{13}	0.26_8	2.40_{17}	5.01_{10}	9.12_{12}	12.8_{10}	2.78_{11}	8.57_{19}	6.99_4
SurfaceStereo [26]	13.7	1.28_{24}	1.65_{15}	6.78_{28}	0.19_{15}	0.28_9	2.61_{24}	3.12_1	5.10_1	8.65_1	2.89_{15}	7.95_{11}	8.26_{20}

[3] A. Hosni, M. Bleyer, M. Gelautz, and C. Rheman, “Local stereo matching using geodesic support weights,” in *Proc. ICIP*, 2009, pp. 2093–2096.

[4] K.-J. Yoon and I.-S. Kweon, “Adaptive support-weight approach for correspondence search,” *IEEE TPAMI*, vol. 28, no. 4, pp. 650–656, 2006.

[5] F. Tombari, S. Mattoccia, and L. D. Stefano, “Segmentation-based adaptive support for accurate stereo correspondence,” in *Proc. PSIVT*, 2007, pp. 427–438.

[6] A. F. Bobick and S. S. Intille, “Large occlusion stereo,” *IJCV*, vol. 33, no. 3, pp. 181–200, 1999.

[7] Y. Boykov, O. Veksler, and R. Zabih, “Fast approximate energy minimization via graph cuts,” *IEEE TPAMI*, vol. 23, no. 11, pp. 1222–1239, 2001.

[8] P. F. Felzenszwalb and D. P. Huttenlocher, “Efficient belief propagation for early vision,” *IJCV*, vol. 70, no. 1, pp. 41–54, 2006.

[9] D. Scharstein and R. Szeliski, “Middlebury stereo evaluation - version 2,” 2010, http://vision.middlebury.edu/stereo/eval/.

[10] M. Gong and Y. Yang, “Fast stereo matching using reliability-based dynamic programming and consistency constraints,” in *Proc. ICCV*, 2003, pp. 610–617.

[11] C. Kim, K. M. Lee, B. T. Choi, and S. U. Lee, “A dense stereo matching using two-pass dynamic programming with generilized ground truth points,” in *Proc. CVPR*, 2005, pp. 1075–1082.

[12] M. Lhuillier and L. Quan, “Match propagation for image-based modeling and rendering,” *IEEE TPAMI*, vol. 24, no. 8, pp. 1140–1146, 2002.

[13] J. Čech and R. Šára, “Efficient sampling of disparity space for fast and accurate matching,” in *Proc. CVPR*, 2007, pp. 1–8.

[14] Y. Wei and L. Quan, “Region-based progressive stereo matching,” in *Proc. CVPR*, 2004, pp. 106–113.

[15] K. Zhang, J. Lu, and G. Lafruit, “Cross-based local stereo matching using orthogonal integral images,” *IEEE TCSVT*, vol. 19, no. 7, pp. 1073–1079, 2009.

[16] R. Zabih and J. Woodfill, “Non-parametric local transforms for computing visual correspondence,” in *Proc. ECCV*, 1994, pp. 151–158.

[17] L. Wang, M. Liao, M. Gong, R. Yang, and D. Nister, “High-quality real-time stereo using adaptive cost aggregation and dynamic programming,” in *Proc. 3DPVT*, 2006, pp. 798–805.

[18] Q. Yang, L. Wang, and N. Ahuja, “A constant-space belief propagation algorithm for stereo matching,” in *Proc. CVPR*, 2010, pp. 1458–1465.

[19] H. Hirschmüller and D. Scharstein, “Evaluation of cost functions for stereo matching,” in *Proc. CVPR*, 2007, pp. 1–8.

[20] D. Scharstein and C. Pal, “Learning conditional random fields for stereo,” in *Proc. CVPR*, 2007, pp. 1–8.

[21] C. Tomasi and R. Manduchi, “Bilateral filtering for gray and color images,” in *Proc. ICCV*, 1998, pp. 839–846.

[22] H. Hirschmüller, “Stereo processing by semiglobal matching and mutual information,” *IEEE TPAMI*, vol. 30, no. 2, pp. 328–341, 2008.

[23] Q. Yang, L. Wang, R. Yang, H. Stewénius, and D. Nistér, “Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling,” *IEEE TPAMI*, vol. 31, no. 3, pp. 492–504, 2009.

[24] Z. Wang and Z. Zheng, “A region based stereo matching algorithm using cooperative optimization,” in *Proc. CVPR*, 2008, pp. 1–8.

[25] X. Li and J. Jia, “Stereo matching: An outlier confidence approach,” in *Proc. ECCV*, 2008, pp. 775–787.

[26] M. Bleyer, C. Rother, and P. Kohli, “Surface stereo with soft segmentation,” in *Proc. CVPR*, 2010, pp. 1570–1577.