ADAPTIVE SMOOTHNESS CONSTRAINTS FOR EFFICIENT STEREO MATCHING USING TEXTURE AND EDGE INFORMATION

Kyung-Rae Kim and Chang-Su Kim

School of Electrical Engineering, Korea University, Seoul, Korea
E-mails: krkim@mcl.korea.ac.kr, changsukim@korea.ac.kr

ABSTRACT

An efficient stereo matching algorithm, which applies adaptive smoothness constraints using texture and edge information, is proposed in this work. First, we determine non-textured regions, on which an input image yields flat pixel values. In the non-textured regions, we penalize depth discontinuity and complement the primary CNN-based matching cost with a color-based cost. Second, by combining two edge maps from the input image and a pre-estimated disparity map, we extract denoised edges that correspond to depth discontinuity with high probabilities. Thus, near the denoised edges, we penalize small differences of neighboring disparities. Based on these adaptive smoothness constraints, the proposed algorithm outperforms the conventional methods significantly and achieves the state-of-the-art performance on the Middlebury stereo benchmark.

Index Terms— Stereo matching, texture analysis, edge analysis, adaptive smoothness constraint.

1. INTRODUCTION

Dense stereo matching aims to estimate a pixel-wise disparity map from a pair of stereo images by matching the correspondences between the two images. Even though stereo matching is a well studied topic in low-level vision, it is still challenging to provide accurate pixel-wise disparities in case of occlusions, near object boundaries, and on repetitive texture regions or textureless regions. To overcome this problem and provide more precise and reliable disparity maps, numerous stereo matching methods have been developed.

Scharstein and Szeliski [1] observe that a general stereo matching method has four steps: matching cost computation, cost aggregation, disparity optimization, and disparity refinement. According to the techniques of cost aggregation and disparity optimization, stereo methods can be classified into two categories: local methods and global methods. Local stereo methods aggregate matching costs, depending on support regions, and simply choose the disparity that minimize the cost for each pixel [2–5]. Various windows or filters for support regions have been proposed, including the adaptive support-weight [2], the guided filter [3, 4] and the trilateral filter [5]. These local methods, however, are ineffective on large textureless regions or repetitive textures.

Global stereo methods [6–19] make explicit smoothness assumptions, whereas local methods make implicit ones. More specifically, a global stereo method optimizes a global energy function that consists of data and smoothness terms over an entire image. In order to optimize the energy function, efficient techniques such as dynamic programming [6], graph cuts [7] and belief propagation [8] can be utilized. Moreover, Hirschmüller [9] proposes the semiglobal matching (SGM), which optimizes a 2D global energy function efficiently by minimizing 1D aggregated matching costs in several directions. Hong and Chen [10] formulate stereo matching as the minimization of an energy function in the segment domain, instead of the typical pixel domain. Klaus *et al*. [11] use a color segmentation result of an image and adopt self-adapting matching scores for the disparity plane fitting. Humenberger *et al*. [12] compute the census-based correlation as a matching cost and obtain a disparity map using the texture-based plane fitting. Mei *et al*. [13] adopt the census transform to compute the matching cost and use the cross-based cost aggregation [20]. Yamaguchi *et al*. [14] formulate stereo matching with a hybrid Markov random field (MRF), composed of continuous random variables for slanted 3D planes and discrete random variables for occlusion boundaries. Jung *et al*. [15] exploit the consistency criterion across real and virtual intermediate views and minimize an energy function, including the consistency term. Sinha *et al*. [16] model locally slanted planes by matching and clustering sparse local features, and solve the local plane sweep problem using SGM. Yamaguchi *et al*. [17] construct a slanted plane model over superpixels and optimize an energy function, composed of segment label and boundary energy terms. Žbontar and LeCun [18] compute matching costs by learning a convolutional neural network (CNN). Zhang *et al*. [19] construct a mesh structure for stereo matching and optimize it by employing a two-layer MRF.

In this paper, we present an efficient global stereo matching algorithm, which applies adaptive smoothness constraints using texture and edge information. First, we determine non-textured regions by detecting windows that do not include strong gradient magnitudes in an input image. Then, in the non-textured regions, we penalize large differences between adjacent disparities and augment the baseline CNN-based matching cost [18] with a color-based one. Second, we extract edge maps from the input image and a pre-estimated disparity map, and combine them to yield denoised edges. Near the denoised edges, we encourage large disparity differences and penalize small differences. By imposing these adaptive smoothness constraints, the proposed algorithm provides excellent stereo matching performance. Extensive experimental results on the Middlebury stereo data set confirm that the proposed algorithm significantly outperforms the conventional methods.

2. PRELIMINARIES

This section presents a baseline stereo matching method, on which the proposed algorithm is based. This baseline method has been also adopted in recent stereo matching algorithms, including [13, 18].

Let I^{L} and I^{R} be a rectified pair of left and right images. When

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIP) (No. NRF-2015R1A2A1A10055037).

a pixel $\mathbf{p} = (x, y)$ in I^{L} matches another pixel $\mathbf{p}_d = (x - d, y)$ in I^{R} via a disparity d, the pixel-wise matching cost at $\mathbf{p}$ is denoted by $C(\mathbf{p}, d)$. In this work, we compute the matching cost by employing the CNN in [18], which provides excellent matching performance by introducing machine learning techniques into the stereo matching. Then, by using the cross-based cost aggregation [20], we combine matching costs of neighboring pixels with similar values.

To obtain a dense disparity map D, an energy function $E(D)$ is defined, which consists of data and smoothness terms in general. In particular, in [9], the energy function is defined as

$$\begin{aligned} E(D) \;&=\; \sum_{\mathbf{p}} \Bigg(C(\mathbf{p}, D_{\mathbf{p}}) + \sum_{\mathbf{q} \in \mathcal{N}_{\mathbf{p}}} P_1 \cdot \mathrm{T}[|D_{\mathbf{p}} - D_{\mathbf{q}}| = 1] \\ &\qquad + \sum_{\mathbf{q} \in \mathcal{N}_{\mathbf{p}}} P_2 \cdot \mathrm{T}[|D_{\mathbf{p}} - D_{\mathbf{q}}| > 1] \Bigg) \end{aligned} \tag{1}$$

where P_1 and P_2 are smoothness penalties, and $P_1 < P_2$. Also, $\mathrm{T}[\cdot]$ is the indicator function, $D_{\mathbf{p}}$ is the disparity at $\mathbf{p}$, and $\mathcal{N}_{\mathbf{p}}$ is the set of adjacent pixels to $\mathbf{p}$. The second term imposes a small penalty P_1 on the pixels $\mathbf{q}$ whose disparities differ by one from the disparity of $\mathbf{p}$. Similarly, the third term imposes a large penalty P_2 on the pixels $\mathbf{q}$ whose disparities differ by more than one.

To minimize the energy function $E(D)$ efficiently, we apply the SGM method [9], which approximates the global energy function based on the path-wise optimization in two horizontal and two vertical directions. More specifically, to minimize $E(D)$ in a direction $\mathbf{r}$, the path-wise matching cost is formulated recursively by

$$\begin{aligned} L_{\mathbf{r}}(\mathbf{p}, d) = C(\mathbf{p}, d) + \min\{&L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d), \\ &L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d - 1) + P_1, L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, d + 1) + P_1, \\ &\min_i L_{\mathbf{r}}(\mathbf{p} - \mathbf{r}, i) + P_2\}. \end{aligned} \tag{2}$$

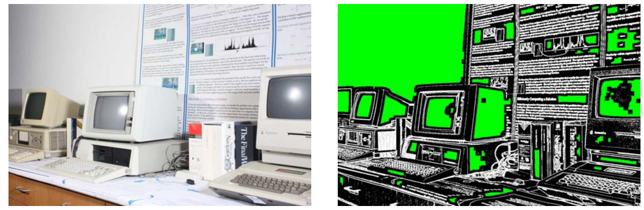
Then, the dense disparity map D is determined by summing up the path-wise matching costs in all directions and minimizing the sum,

$$D^*_{\mathbf{p}} = \arg\min_d \sum_{\mathbf{r}} L_{\mathbf{r}}(\mathbf{p}, d). \tag{3}$$

This result is refined in a post-processing step. Specifically, for the left-right consistency check, the left disparity map D^{L} is computed by considering the left image as a reference image, and the right disparity map D^{R} is computed similarly in a symmetrical manner. If $\left|D^{\mathrm{L}}_{\mathbf{p}} - D^{\mathrm{R}}_{\mathbf{p}_d}\right| > 1$, the disparity of $\mathbf{p}$ is regarded as erroneous and interpolated using those of neighboring pixels. Then, the subpixel estimation is performed through a quadratic curve fitting of the neighboring costs. Finally, a 5×5 median filter and a bilateral filter are applied subsequently.

3. PROPOSED ALGORITHM

We propose adaptive smoothness constraints using texture and edge information, to improve the performance of stereo matching. First, we determine a non-textured region using a gradient map. Then, we obtain a denoised edge map using both color and pre-estimated disparity information. Lastly, based on the non-textured region and the denoised edge map, we compte a final disparity map by optimizing a modified energy function with the adaptive smoothness terms.



(a) (b)

Fig. 1. (a) An input image and (b) its gradient map. In (b), green parts depict the detected non-textured region.

3.1. Adaptive Smoothness Constraint in Non-Textured Regions

Generally, non-textured regions, which are flat in pixel intensities, also have homogeneous disparities. However, the matching cost computation, including the CNN-based one [18], in non-textured regions is unreliable, and the disparity estimation in such regions is one of the major challenges in dense stereo matching. Hence, to improve the matching performance, we find non-textured regions and apply an adaptive smoothness constraint in those regions. Also, in non-textured regions, we compute the differences of color intensities and combine them with the CNN-based matching cost.

To find non-textured regions, we extract the gradient magnitude maps $I_{\mathrm{g},x}$ and $I_{\mathrm{g},y}$ in the x and y directions from an input image I, after normalizing it by subtracting the mean intensity and dividing by the standard deviation. Then, by employing a threshold $\tau_{\mathrm{g}} = 0.1$, we obtain a set $\mathcal{P}_{\mathrm{g}} = \{\mathbf{p} \mid I_{\mathrm{g},x}(\mathbf{p}) \geq \tau_{\mathrm{g}} \text{ or } I_{\mathrm{g},y}(\mathbf{p}) \geq \tau_{\mathrm{g}}\}$. We find a window, which does not include any pixel in $\mathcal{P}_{\mathrm{g}}$, and declare that all pixels in the window belong to the non-textured region. In other words, the non-textured region $\mathcal{P}_{\mathrm{NT}}$ is defined as

$$\mathcal{P}_{\mathrm{NT}} = \left\{ \mathbf{p} \mid {}^{\exists}W_{\mathbf{p}} \text{ such that } \sum_{\mathbf{q} \in W_{\mathbf{p}}} \mathrm{T}[\mathbf{q} \in \mathcal{P}_{\mathrm{g}}] = 0 \right\} \tag{4}$$

where $W_{\mathbf{p}}$ is a 31×31 window including pixel $\mathbf{p}$. Also, to reduce detection errors, the pixels within ten pixel distances from $\mathbf{q} \in \mathcal{P}_{\mathrm{g}}$ are excluded from the non-textured region, as shown in Fig. 1.

As mentioned above, in the non-textured region $\mathcal{P}_{\mathrm{NT}}$, disparities tend to be homogeneous or continuous. Hence, in $\mathcal{P}_{\mathrm{NT}}$, we set a large penalty $\hat{P}_2$, instead of P_2 in (1), for a large disparity difference between adjacent pixels by

$$\hat{P}_2(\mathbf{p}) = \begin{cases} P_2 & \text{if } \mathbf{p} \notin \mathcal{P}_{\mathrm{NT}}, \\ \sigma_{\mathrm{NT}} P_2 & \text{if } \mathbf{p} \in \mathcal{P}_{\mathrm{NT}}, \end{cases} \tag{5}$$

where σ_{NT} is an amplification parameter greater than 1.

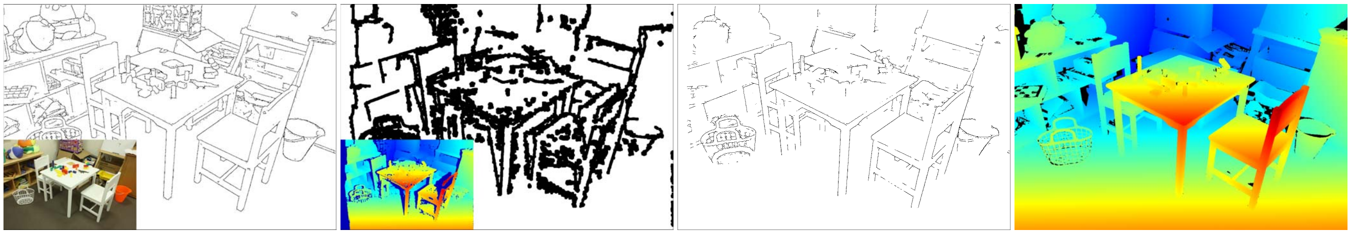
Furthermore, in $\mathcal{P}_{\mathrm{NT}}$, we compute an additional matching cost $C_{\mathrm{a}}(\mathbf{p}, d)$ at pixel $\mathbf{p}$ by computing the sum of the l_2-distances of colors between $V_{\mathbf{p}}$ and $V_{\mathbf{p}_d}$ in the RGB space. Here, $V_{\mathbf{p}}$ denotes a 5×5 window centered at $\mathbf{p}$. The difference of color intensities is simple and can compensate for the weakness of the CNN-based matching cost $C(\mathbf{p}, d)$ in (1) in the non-textured region. To summarize, we compute the matching cost by

$$\hat{C}(\mathbf{p}, d) = \begin{cases} C(\mathbf{p}, d) & \text{if } \mathbf{p} \notin \mathcal{P}_{\mathrm{NT}}, \\ C(\mathbf{p}, d) + \omega C_{\mathrm{a}}(\mathbf{p}, d) & \text{if } \mathbf{p} \in \mathcal{P}_{\mathrm{NT}}, \end{cases} \tag{6}$$

where $\omega = 0.2$ is a weight parameter. In (6), the additional matching cost $C_{\mathrm{a}}(\mathbf{p}, d)$ is normalized into the range $[0, 1]$.

3.2. Adaptive Smoothness Constraint on Denoised Edges

We develop another adaptive smoothness constraint, which employs a denoised edge map. We exploit the tendency that depth discontinuity occurs at edges in an input image. However, edges do not



(a) (b) (c) (d)

Fig. 2. Denoising of an edge map: (a) an edge map for an input image, (b) an edge map for the pre-estimated disparity image, (c) the denoised edge map, and (d) the ground truth disparity map.

always imply depth discontinuity. Therefore, we define edges in an image, which are in homogeneous regions in the corresponding disparity map, as noisy edges. We eliminate those noisy edges similarly to [21], but using a simpler scheme. Note that a pre-estimated disparity map using the constraint in Section 3.1 can inform of depth discontinuity roughly. Thus, we use the pre-estimated disparity map to eliminate noisy edges.

We obtain an edge map E_{I} for the input image. We also extract an edge map E_{P} for the pre-estimated disparity map, which is dilated to broaden the edges. For computing both edge maps, we use the sketch tokens in [22]. By employing a threshold $\epsilon = 0.5$, we obtain two sets of edge pixels $\mathcal{P}_{\mathrm{I}} = \{\mathbf{p} \mid E_{\mathrm{I}}(\mathbf{p}) \geq \epsilon\}$ and $\mathcal{P}_{\mathrm{P}} = \{\mathbf{p} \mid E_{\mathrm{P}}(\mathbf{p}) \geq \epsilon\}$. A set of denoised edge pixels $\mathcal{P}_{\mathrm{DE}}$ is then defined as the intersection of $\mathcal{P}_{\mathrm{I}}$ and $\mathcal{P}_{\mathrm{P}}$,

$$\mathcal{P}_{\mathrm{DE}} = \mathcal{P}_{\mathrm{I}} \cap \mathcal{P}_{\mathrm{P}}. \tag{7}$$

Fig. 2 shows step-by-step results of denoising an edge map. We see in Fig. 2(c) that edges in the denoised edge map faithfully indicate large differences of neighboring disparities.

In the set of denoised edge pixels $\mathcal{P}_{\mathrm{DE}}$, the probability of a large difference between adjacent disparities is very high. Therefore, contrary to non-edge pixels, in $\mathcal{P}_{\mathrm{DE}}$, we set a small penalty for a large difference of adjacent disparities and a large penalty to a small difference. To this end, by adding a new smoothness term, we modify the energy function in (1) to

$$\begin{aligned}\hat{E}(D) \quad = \quad & \sum_{\mathbf{p}} \Bigg(\hat{C}(\mathbf{p}, D_{\mathbf{p}}) + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_0^{\mathrm{E}}(\mathbf{p}) \cdot \mathrm{T}[\mathrm{D}_{\mathbf{p}} = \mathrm{D}_{\mathbf{q}}] \\ & + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_1^{\mathrm{E}}(\mathbf{p}) \cdot \mathrm{T}[|\mathrm{D}_{\mathbf{p}} - \mathrm{D}_{\mathbf{q}}| = 1] \\ & + \sum_{\mathbf{q} \in N_{\mathbf{p}}} P_2^{\mathrm{E}}(\mathbf{p}) \cdot \mathrm{T}[|\mathrm{D}_{\mathbf{p}} - \mathrm{D}_{\mathbf{q}}| > 1] \Bigg)\end{aligned} \tag{8}$$

and determine the three adaptive penalties by

$$P_0^{\mathrm{E}}(\mathbf{p}) = \begin{cases} 0 & \text{if } \mathbf{p} \notin \mathcal{P}_{\mathrm{DE}}, \\ \hat{P}_2(\mathbf{p}) & \text{if } \mathbf{p} \in \mathcal{P}_{\mathrm{DE}}. \end{cases} \tag{9}$$

$$P_1^{\mathrm{E}}(\mathbf{p}) = \begin{cases} P_1 & \text{if } \mathbf{p} \notin \mathcal{P}_{\mathrm{DE}}, \\ \hat{P}_2(\mathbf{p}) & \text{if } \mathbf{p} \in \mathcal{P}_{\mathrm{DE}}. \end{cases} \tag{10}$$

$$P_2^{\mathrm{E}}(\mathbf{p}) = \begin{cases} \hat{P}_2(\mathbf{p}) & \text{if } \mathbf{p} \notin \mathcal{P}_{\mathrm{DE}}, \\ P_1 & \text{if } \mathbf{p} \in \mathcal{P}_{\mathrm{DE}}. \end{cases} \tag{11}$$

Note that $\hat{P}_2(\mathbf{p})$, which is defined in (5), is larger than P_1. Therefore, neighboring disparity differences of 0 or 1 are strongly penalized in the set of denoised edge pixels $\mathcal{P}_{\mathrm{DE}}$, whereas differences larger than 1 are strongly penalized in the set of ordinary pixels $\mathcal{P}_{\mathrm{DE}}^{c}$. We obtain the dense disparity map D by minimizing $\hat{E}(D)$ in (8) and post-processing the result as mentioned in Section 2. By employing the adaptive penalties in (9)$\sim$(11), we can improve the matching performance significantly.

4. EXPERIMENTAL RESULTS

We evaluate the performance of the proposed stereo matching algorithm on the Middlebury stereo benchmark [23]. The percentage of disparity errors that are greater than one or two pixels (bad 1.0, bad 2.0) and the average absolute disparity error (avgerr) are used to measure stereo results. For implementing the proposed algorithm, we mostly follow the approach of the MC-CNN-acrt method [18], except that we skip the cross-based cost aggregation after the semiglobal matching and the reduction of P_1 in the vertical direction. The proposed algorithm has two steps: First, it obtains a pre-estimated disparity map using only the non-texture constraint in Section 3.1. Second, it further applies the denoised edge constraint in Section 3.2. In the first step, we set the parameters as $P_1 = 1.3$ in (1), $P_2 = 18.1$ and $\sigma_{\mathrm{NT}} = 4.4$ in (5). In the second step, $P_1 = 1.0$, $P_2 = 45$, and $\sigma_{\mathrm{NT}} = 2.0$.

4.1. Component Analysis

Let us analyze the performance gain that is achieved by each component in the proposed algorithm, in comparison with the baseline method in Section 2. We evaluate three versions of the proposed algorithm:

- NT - only the non-texture constraint is used
- DE - only the denoised edge constraint is used
- NTDE - both constraints are used.

In other words, NT yields a pre-estimated disparity map as a final result. When evaluating DE, $\hat{C}(\mathbf{p}, D_{\mathbf{p}})$ and $\hat{P}_2(\mathbf{p})$ in (8)$\sim$(11) are replaced by $C(\mathbf{p}, D_{\mathbf{p}})$ and P_2, respectively.

Table 1 compares the results on the Middlebury stereo *training* data sets [23]. Since the provided MC-CNN-acrt network [18] for the matching cost computation was trained using the Middlebury training data sets, we train a new network without using them. As compared with the baseline method, both NT and DE reduce the error rates, but we observe that DE is more effective than NT in general. Moreover, by combining both constraints, the proposed NTDE algorithm provides the best performance.

4.2. Comparison with Conventional Stereo Methods

Next, in Table 2, we compare the performances of the proposed algorithm with those of the 11 state-of-the-art stereo methods whose results are available in [23]. The Middlebury stereo *test* data sets are used in this test. In Table 2, the methods without references are

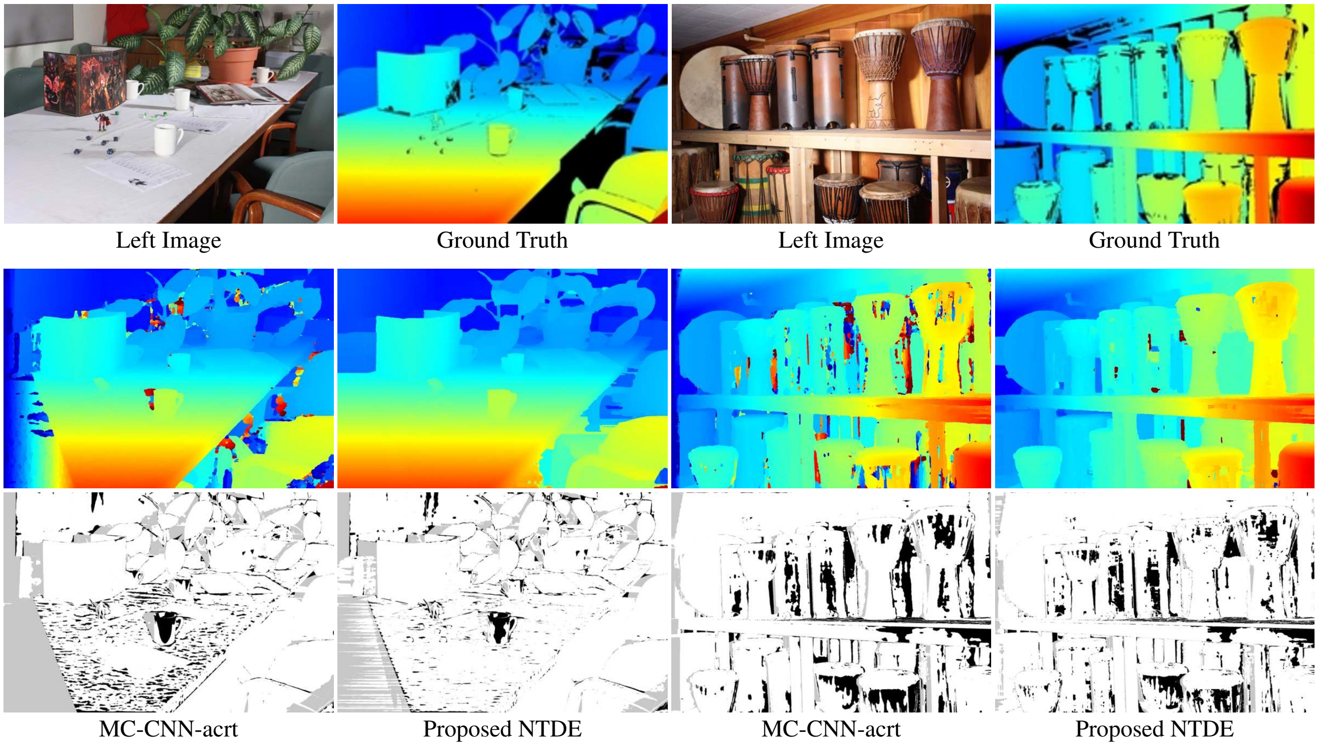


Fig. 3. Comparison of disparity maps of the proposed NTDE algorithm with those of the MC-CNN-acrt method [18]. The first and second columns are the results on "CrusadeP," while the others are on "DjembeL." The second row provides the resultant disparity maps. The third row marks disparity errors greater than two, in which black and gray depict the erroneous pixels in non-occluded areas and occluded areas, respectively.

Table 1. Component analysis using the Middlebury stereo training data sets [23]. The performance on non-occluded pixels (Nonocc) and all pixels (All) are evaluated.

	bad 1.0		bad 2.0		avgerr	
	Nonocc	All	Nonocc	All	Nonocc	All
Baseline	19.8	27.3	10.7	17.5	2.95	5.97
NT	19.7	27.2	10.4	17.3	2.47	5.64
DE	19.5	27.1	10.3	17.2	2.48	5.38
Proposed NTDE	**19.4**	**27.0**	**10.1**	**17.1**	**2.20**	**5.23**

not published yet. We observe that the proposed algorithm significantly outperforms the conventional methods in all categories. For example, in terms of the bad 2.0 metric on non-occluded pixels, the proposed NTDE algorithm yields 7.62, while the second best performing method provides 8.29. As of January 31, 2016, the proposed algorithm ranks 1st in the Middlebury stereo benchmark.

Fig. 3 compares disparity maps of the proposed algorithm with those of the state-of-the-art MC-CNN-acrt method [18]. We see that the proposed algorithm outperforms MC-CNN-acrt and provides more accurate and more faithful disparities. For example, MC-CNN-acrt yields noisy disparities on the non-textured desk in "CrusadeP." In contrast, the proposed NTDE algorithm suppresses the noise effectively and provides much cleaner disparities. Also, we see that the proposed algorithm provides more precise disparities near object boundaries as well.

Table 2. Comparison of the proposed NTDE algorithm with the state-of-the-art stereo methods on the Middlebury stereo test datasets [23]. The best performance is boldfaced, while the second best performance is underlined.

	bad 1.0		bad 2.0		avgerr	
	Nonocc	All	Nonocc	All	Nonocc	All
TMAP [24]	32.6	39.4	17.1	24.8	4.75	10.1
LCU	39.6	45.3	17.0	24.0	3.86	7.15
GCSVR	25.5	32.6	14.8	21.7	7.97	11.5
INTS	29.0	35.8	14.8	21.9	3.90	8.10
SOU4P-net	25.3	34.4	13.5	23.6	4.18	17.7
MeshStereo [19]	26.8	33.7	13.4	20.1	4.63	8.12
MDP	31.1	37.4	12.6	20.2	5.28	13.6
MC-CNN-fst [18]	18.9	29.1	9.69	20.8	4.37	19.3
MCCNN_Layout	18.7	27.0	9.16	17.4	4.27	12.0
MC-CNN+FBS	18.9	28.2	8.62	18.4	2.67	8.19
MC-CNN-acrt [18]	18.0	28.1	8.29	19.2	3.82	17.9
Proposed NTDE	**17.1**	**25.7**	**7.62**	**15.5**	**2.58**	**6.21**

5. CONCLUSIONS

In this work, we proposed an efficient stereo matching algorithm, called NTDE, which applies the adaptive smoothness constraints using texture and edge information. The proposed algorithm first detects non-textured regions, which are flat in pixel intensities. In the non-textured regions, the proposed algorithm amplifies the penalty for depth discontinuity and complements the primary CNN-based matching cost with a color-based cost. Also, it extracts denoised edges, by intersecting two edge maps from an input image and a pre-estimated disparity map. Then, it applies a large penalty for small differences of adjacent disparities on the denoised edges. Experimental results demonstrated that the proposed NTDE algorithm outperforms the conventional stereo methods significantly and ranks 1st in the Middlebury stereo benchmark.

6. REFERENCES

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

[2] K.-J. Yoon and I. S. Kweon, "Adaptive support-weight approach for correspondence search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 650–656, 2006.

[3] A. Hosni, C. Rhemann, M. Bleyer, C. Rother, and M. Gelautz, "Fast cost-volume filtering for visual correspondence and beyond," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 2, pp. 504–511, 2013.

[4] K. He, J. Sun, and X. Tang, "Guided image filtering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 6, pp. 1397–1409, 2013.

[5] D. Chen, M. Ardabilian, and L. Chen, "A novel trilateral filter based adaptive support weight method for stereo matching," in *Proc. Brit. Mach. Vis. Conf.*, Oct. 2013, pp. 1–11.

[6] S. Birchfield and C. Tomasi, "Depth discontinuities by pixel-to-pixel stereo," *Int. J. Comput. Vis.*, vol. 35, no. 3, pp. 269–293, 1999.

[7] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 11, pp. 1222–1239, 2001.

[8] J. Sun, N.-N. Zheng, and H.-Y. Shum, "Stereo matching using belief propagation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 7, pp. 787–800, 2003.

[9] H. Hirschmüller, "Stereo processing by semiglobal matching and mutual information," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 2, pp. 328–341, 2008.

[10] L. Hong and G. Chen, "Segment-based stereo matching using graph cuts," in *Proc. IEEE CVPR*, June 2004, pp. 74–81.

[11] A. Klaus, M. Sormann, and K. Karner, "Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure," in *Proc. Int. Conf. Pattern Recognition*, Aug. 2006, pp. 15–18.

[12] M. Humenberger, T. Engelke, and W. Kubinger, "A census-based stereo vision algorithm using modified semi-global matching and plane fitting to improve matching quality," in *Proc. IEEE CVPR Workshops*, June 2010, pp. 77–84.

[13] X. Mei, X. Sun, M. Zhou, S. Jiao, H.Wang, and X. Zhang, "On building an accurate stereo matching system on graphics hardware," in *Proc. IEEE ICCV Workshops*, Nov. 2011, pp. 467–474.

[14] K. Yamaguchi, T. Hazan, D. McAllester, and R. Urtasun, "Continuous markov random fields for robust stereo estimation," in *Proc. IEEE ECCV*, Oct. 2012, pp. 45–58.

[15] I.-L. Jung, T.-Y. Chung, J.-Y. Sim, and C.-S. Kim, "Consistent stereo matching under varying radiometric conditions," *IEEE Trans. Multimedia*, vol. 15, no. 1, pp. 56–69, 2013.

[16] S. N. Sinha, D. Scharstein, and R. Szeliski, "Efficient high-resolution stereo matching using local plane sweeps," in *Proc. IEEE CVPR*, June 2014, pp. 1582–1589.

[17] K. Yamaguchi, D. McAllester, and R. Urtasun, "Efficient joint segmentation, occlusion labeling, stereo and flow estimation," in *Proc. IEEE ECCV*, Sept. 2014, pp. 756–771.

[18] J. Žbontar and Y. LeCun, "Stereo matching by training a convolutional neural network to compare image patches," arXiv:1510.05970, 2015.

[19] C. Zhang, Z. Li, Y. Cheng, R. Cai, H. Chao, and Y. Rui, "MeshStereo: A global stereo model with mesh alignment regularization for view interpolation," in *Proc. IEEE ICCV*, Dec. 2015, pp. 2057–2065.

[20] K. Zhang, J. Lu, and G. Lafruit, "Cross-based local stereo matching using orthogonal integral images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 7, pp. 1073–1079, 2009.

[21] D. Chen, M. Ardabilian, and L. Chen, "Depth edge based trilateral filter method for stereo matching," in *Proc. IEEE ICIP*, Sept. 2015, pp. 2280–2284.

[22] J. J. Lim, C. L. Zitnick, and P. Dollár, "Sketch tokens: A learned mid-level representation for contour and object detection," in *Proc. IEEE CVPR*, June 2013, pp. 3158–3165.

[23] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nešić, X. Wang, and P. Westling, "High-resolution stereo datasets with subpixel-accurate ground truth," in *Proc. German Conf. Pattern Recognit. (GCPR)*, Jan. 2014, pp. 31–42.

[24] E. T. Psota, J. Kowalczuk, M. Mittek, and L. C. Pérez, "MAP disparity estimation using hidden markov trees," in *Proc. IEEE ICCV*, Dec. 2015, pp. 2219–2227.