

Global Depth Refinement Based on Patches

Xu Huang¹(✉), Yanfeng Zhang², Gang Zhou¹, Lu Liu¹,
and Gangshan Cai¹

¹ Wuhan Engineering Science and Technology Institute,
Jiangda Road 30, Wuhan, China
huangxu.chess@163.com

² School of Remote Sensing and Information Engineering,
Wuhan University, Wuhan, China

Abstract. Current stereo matching methods can be divided into 1D label algorithms and 3D label algorithms. 1D label algorithms are simple and fast, but they can't avoid fronto-parallel bias. 3D label algorithms can solve fronto-parallel bias. However, they are very time-consuming. In order to avoid fronto-parallel bias efficiently, this paper introduces a new global depth refinement based on patches. The method transforms the depth optimization problem into a quadratic function computation, which has a low time complexity. Experiments on Motorcycle imagery and Wuhan university imagery verify the correctness and the effectiveness of the proposed method.

Keywords: 1D label · 3D label · Fronto-parallel bias · Patch
Global optimization

1 Introduction

Stereo dense matching has been attracting increased attention in the photogrammetry and computer vision communities for decades [1]. According to the assignments of every pixels, stereo matching methods can be divided into 1D label methods and 3D label methods. 1D label algorithms assume fronto-parallel planes and assign one label for every pixel. 3D label algorithms assign three labels (disparity and normal direction) for every pixel [2]. The newest rank in Middlebury Benchmark show that there are no significant advantages on matching accuracies for both kinds of matching methods, as shown in Table 1. PMSC [3] and MeshStereoExt [4] belong to 3D label methods. LW-CNN [5], NTDE [6] and MC-CNN-arct [7] belong to 1D label methods.

Table 1. Rank in middlebury stereo version 3 (11/01/2017).

Matching algorithm	Rank	Running time	Weight avg.	Running environment
PMSC	1	599 s	14.8	GPU + 1 CPU @4 GHz
LW-CNN	2	314 s	14.9	GPU + 1 CPU @4 GHz
MeshStereoExt	3	161 s	15.6	GPU + 8 CPU
NTDE	4	152 s	16.2	GPU + 1 CPU @2.2 GHz
MC-CNN-arct	5	150 s	17.1	GPU + 1 CPU

1D label algorithms are usually simple and fast, and they can acquire disparity image directly. According to the cost aggregation, 1D label algorithms can be divided into semi-global matching (SGM) [8], image-guided matching [9, 10] and global matching [11]. However, 1D label methods assume fronto-parallel planes and produces fronto-parallel bias in slanted planes, as shown in Fig. 1. Figure 1(a) shows the original reference image. The surface of the lamp is a typical slanted plane. Figure 1(b) shows the corresponding ground truth. Figure 1(c)–(g) represent the matching results of image-guided matching (IG) [10], SGM [8], Graph Cut (GC) [11], INTS [12] and NTDE [6], respectively. All of above algorithms are 1D label algorithms. The fronto-parallel bias in Fig. 1(c)–(g) influences the visualization of 3D reconstruction.

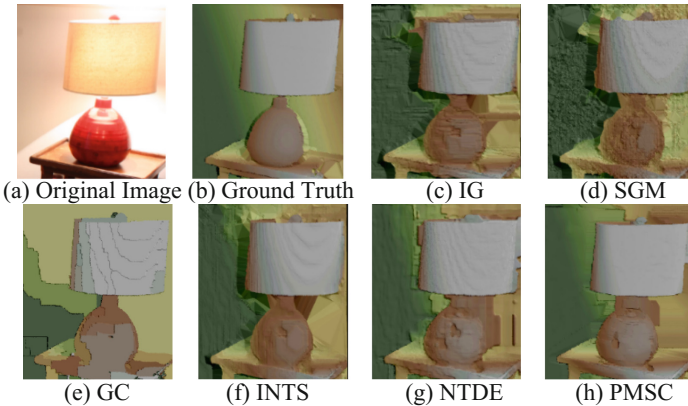


Fig. 1. Results of different matching methods in slanted planes.

PMSC is a 3D label algorithm. 3D label algorithms penalize the angular difference between neighboring tangent plane normals, thus they can avoid fronto-parallel bias in slanted planes, as shown in Fig. 1(h). However, 3D label algorithms are time consuming, which is not suitable for large scale reconstruction.

This paper proposes a new global depth refinement based on patches (GDRP). It can remove fronto-parallel bias efficiently. The contributions of this paper are as follows:

- (1) Traditional 3D label algorithms transform matching into a NP-hard problem, resulting in a high time complexity. The proposed GDRP transforms the depth optimization problem into a quadratic function computation, which is simple and fast.
- (2) The proposed GDRP can refine not only disparity image, but also DSM/DEM products. Disparity and elevation are also called depth in this paper.
- (3) The proposed GDRP can remove fronto-parallel bias and obtain continuous, smooth depths without changing the original matching accuracy.

2 Related Work

The current stereo matching algorithms consist of four steps: (1) cost computation, (2) cost aggregation, (3) disparity computation, and (4) disparity refinement [1]. 3D label algorithms mainly refine cost computation and cost aggregation.

Traditional cost computations assume regular support window with a constant disparity. In practice, the assumption is unlikely to hold in slanted planes. So far, slanted support window based cost computation can be divided into initial matching based cost computation, CNN training based cost computation and Patch Match based cost computation. Initial matching based cost computation adopts 1D label methods to achieve initial matching results quickly, and then changes the support window adaptively, according to the initial matching [13]. CNN training based cost computation [14] uses numerous examples to train a convolutional neural network (CNN). During training, affine windows are used for matching in slanted planes. Patch Match based cost computation [15, 16] adopts PatchMatch [17] method which can directly assign an approximate best 3D label by random sampling for each slanted support window.

The challenge of the cost aggregation is how to perform global optimization in the infinite three dimensional label space of each pixel. The cost aggregation of 3D label methods can be divided into initial matching based cost aggregation and direct cost aggregation. Initial matching based cost aggregation [18–23] uses window matching or 1D label algorithms to achieve initial matching results quickly. The initial matching results are approximate to the ground truth. Then, higher order smoothness constraints are used to optimize the initial matching results iteratively. The direct cost aggregations can achieve accurate matching results without initial matching [3, 4, 24–27]. They define a NP-hard global energy function and use PatchMatch [17] or fusion move [28] to reduce the huge search space in continuous infinite 3D label space. Both initial matching based cost aggregation and direct cost aggregations are iterative optimization processes which are very time consuming.

3 Proposed Method

The work flow of GDRP is shown in Fig. 2. ① The input of GDRP is a depth image. ② SLIC [29] is adopted to segment the depth image. ③, ④ A global energy function including data term and smooth term is constructed to optimize the depth image. ⑤ Feather algorithm is designed to eliminate seam lines between patches.

3.1 SLIC Segmentation

This paper assumes piecewise continuous scene and adopts SLIC [29] to segment the input depth image into a series of patches. S_i represents the i th patch. Every Patch can be described by a depth plane function:

$$d(\mathbf{t}_i) = a_i \cdot \overline{t_{ix}} + b_i \cdot \overline{t_{iy}} + c_i; \mathbf{t}_i \in S_i. \quad (1)$$

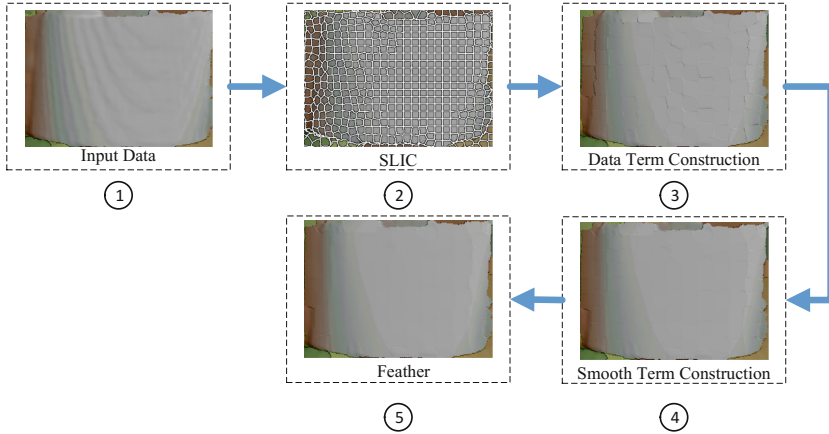


Fig. 2. Work flow of GDRP.

where, a_i, b_i, c_i represents the plane parameters of patch S_i ; $t_i = (t_{ix}, t_{iy})^T$ represents a pixel in S_i ; $(\overline{t_{ix}}, \overline{t_{iy}})^T$ represents centralized coordinates. The purpose of coordinate centralization is to improve the robustness of adjustment models.

Due to radiometric distortions or textureless regions, invalid depths exist in depth images inevitably. Patches in invalid regions are not considered in later optimization for two reasons: (1) Patches in invalid regions are lack of valid depths, resulting in unreliable refinement results; (2) The proposed GDRP aims at smoothing depths instead of interpolation. Number of valid depths in patches are used to judge if patches are valid or not:

$$S_i = \begin{cases} \text{Valid} & |S_i| \geq \delta \\ \text{Invalid} & |S_i| < \delta \end{cases} \quad (2)$$

where, $|S_i|$ represents number of valid depths in S_i ; δ represents threshold.

3.2 Global Energy Function Construction

D represents a depth image. $E(D)$ represents a global Energy function as follows:

$$E(D) = E_{data} + E_{smooth} \quad (3)$$

where, E_{data} represents a data term which controls the approximation between the original depth and the refined depth; E_{smooth} represents a smooth term which controls the smoothness of depths.

3.2.1 Data Term

Data term is defined as the sum of cost of all the valid patches, as follows:

$$E_{data} = \sum_{i=1}^q C(\mathcal{S}_i, \vec{n}_i). \quad (4)$$

where, q represents the number of valid patches; C represents cost of patches; $\vec{n}_i = (a_i, b_i, c_i)^T$ represents depth plane parameters of \mathcal{S}_i . Cost is defined as the distance between original depths and refined depths, as follows:

$$Cost(\mathcal{S}_i, \vec{n}_i) = \sum_{t_i \in \mathcal{S}_i} (a_i \cdot \bar{t}_{ix} + b_i \cdot \bar{t}_{iy} + c_i - d_0(t_i))^2. \quad (5)$$

where, d_0 represents original depths. t_i represents a pixel in \mathcal{S}_i ; $(\bar{t}_{ix}, \bar{t}_{iy})$ represents the centralized coordinates.

Equation (4) can be described in matrix form by defining $\tilde{\mathbf{x}} = (\vec{n}_0 \ \vec{n}_1 \ \dots \ \vec{n}_q)^T$:

$$E_{data} = \tilde{\mathbf{x}}^T \mathbf{G}_{data} \tilde{\mathbf{x}} - 2\mathbf{H}_{data}^T \tilde{\mathbf{x}} + l_{data}. \quad (6)$$

where, \mathbf{G}_{data} represents the coefficient matrix of the quadratic term; \mathbf{H}_{data} represents the coefficient matrix of linear term; l_{data} represents the constant term. All the terms are expressed as follows:

$$\mathbf{G}_{data} = \text{Diag}(\mathbf{g}_i); \mathbf{H}_{data} = \begin{pmatrix} \mathbf{h}_0^T & \mathbf{h}_1^T & \dots & \mathbf{h}_q^T \end{pmatrix}^T; l_{data} = \sum_{i=1}^q l_i;$$

$$\mathbf{g}_i = \begin{pmatrix} \sum_{t_i \in \mathcal{S}_i} \bar{t}_{ix}^2 & \sum_{t_i \in \mathcal{S}_i} \bar{t}_{ix} \cdot \bar{t}_{iy} & \sum_{t_i \in \mathcal{S}_i} \bar{t}_{ix} \\ \sum_{t_i \in \mathcal{S}_i} \bar{t}_{ix} \cdot \bar{t}_{iy} & \sum_{t_i \in \mathcal{S}_i} \bar{t}_{iy}^2 & \sum_{t_i \in \mathcal{S}_i} \bar{t}_{iy} \\ \sum_{t_i \in \mathcal{S}_i} \bar{t}_{ix} & \sum_{t_i \in \mathcal{S}_i} \bar{t}_{iy} & |\mathcal{S}_i| \end{pmatrix};$$

$$\mathbf{h}_i = \left(\sum_{t_i \in \mathcal{S}_i} \bar{t}_{ix} \cdot d_0(t) \quad \sum_{t_i \in \mathcal{S}_i} \bar{t}_{iy} \cdot d_0(t) \quad \sum_{t_i \in \mathcal{S}_i} d_0(t) \right)^T; l_i = \sum_{t_i \in \mathcal{S}_i} d_0(t)^2$$

3.2.2 Smooth Term

The smooth term controls the smoothness between patches. In this paper, the smooth term uses border pixels to control the continuity between patches, and uses the center pixels to control the normal direction consistency between patches, as follows:

$$E_{smooth} = \sum_{i=1}^q \left(\sum_{\mathcal{S}_j \in N(\mathcal{S}_i)} P(i, j) \sum_{t \in \mathbf{E}(\mathcal{S}_i, \mathcal{S}_j) \cup c_i} (a_i \bar{t}_{ix} + b_i \bar{t}_{iy} + c_i - a_j \bar{t}_{jx} - b_j \bar{t}_{jy} - c_j)^2 \right). \quad (7)$$

where, $N(\mathcal{S}_i)$ represents the neighbor patch set of \mathcal{S}_i ; $\mathbf{E}(\mathcal{S}_i, \mathcal{S}_j)$ represents pixels in \mathcal{S}_i which is adjacent to \mathcal{S}_j ; $c_i = (c_{ix}, c_{iy})^T$ represents the center pixel in \mathcal{S}_i ; $(\bar{t}_{ix}, \bar{t}_{iy})$ represents the centralized coordinates in \mathcal{S}_i ; $(\bar{t}_{jx}, \bar{t}_{jy})$ represents the centralized coordinates in \mathcal{S}_j ; $P(i, j)$ represents a penalty defined by adjacent relationship between \mathcal{S}_i and \mathcal{S}_j , as follows:

$$P(i, j) = P \cdot \exp\left(-\frac{|\overline{nd}_i - \overline{nd}_j|}{\sigma_d}\right) \cdot \left(1 - \exp\left(-\frac{num(i, j)}{\sigma_n}\right)\right). \quad (8)$$

where, \overline{nd}_i and \overline{nd}_j represents depth averages of adjacent pixels between S_i and S_j , respectively; $num(i, j)$ represents the number of adjacent pixels; σ_d and σ_n represents smooth factors; P represents the given penalty coefficient.

Equation (7) can be described in matrix form by defining $\tilde{\mathbf{x}} = (\vec{n}_0 \ \vec{n}_1 \ \dots \ \vec{n}_q)^T$:

$$E_{smooth} = \tilde{\mathbf{x}}^T \mathbf{G}_s \tilde{\mathbf{x}}. \quad (9)$$

where, \mathbf{G}_s represents the coefficient matrix of the quadratic term, as follows:

$$\mathbf{G}_s = \sum_{i=1}^q \left(\sum_{S_j \in N(S_i)} P(i, j) \cdot \sum_{t \in (E(S_i, S_j) \cup c_i)} \mathbf{g}_{sr}(i, j, t) \right). \quad (10)$$

where,

$$\mathbf{g}_{sr}(i, j, t) = \begin{pmatrix} \mathbf{0}_{3 \times 3} & \dots & \mathbf{0}_{3 \times 3} & \dots & \mathbf{0}_{3 \times 3} & \dots & \mathbf{0}_{3 \times 3} \\ \vdots & \ddots & \vdots & \dots & \vdots & \dots & \vdots \\ \mathbf{0}_{3 \times 3} & \dots & \boldsymbol{\sigma}_1(i, j, t)_{i,i} & \dots & \boldsymbol{\sigma}_3(i, j, t)_{i,j} & \dots & \mathbf{0}_{3 \times 3} \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ \mathbf{0}_{3 \times 3} & \dots & \boldsymbol{\sigma}_3(i, j, t)_{j,i}^T & \dots & \boldsymbol{\sigma}_2(i, j, t)_{j,j} & \dots & \mathbf{0}_{3 \times 3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ \mathbf{0}_{3 \times 3} & \dots & \mathbf{0}_{3 \times 3} & \dots & \mathbf{0}_{3 \times 3} & \dots & \mathbf{0}_{3 \times 3} \end{pmatrix};$$

$$\boldsymbol{\sigma}_1(i, j, t) = \begin{pmatrix} \overline{t_{ix}}^2 & \overline{t_{ix}} \cdot \overline{t_{iy}} & \overline{t_{ix}} \\ \overline{t_{ix}} \cdot \overline{t_{iy}} & \overline{t_{iy}}^2 & \overline{t_{iy}} \\ \overline{t_{ix}} & \overline{t_{iy}} & 1 \end{pmatrix}; \quad \boldsymbol{\sigma}_2(i, j, t) = \begin{pmatrix} \overline{t_{jx}}^2 & \overline{t_{jx}} \cdot \overline{t_{jy}} & \overline{t_{jx}} \\ \overline{t_{jx}} \cdot \overline{t_{jy}} & \overline{t_{jy}}^2 & \overline{t_{jy}} \\ \overline{t_{jx}} & \overline{t_{jy}} & 1 \end{pmatrix};$$

$$\boldsymbol{\sigma}_3(i, j, t) = \begin{pmatrix} -\overline{t_{ix}} \cdot \overline{t_{jx}} & -\overline{t_{ix}} \cdot \overline{t_{jy}} & -\overline{t_{ix}} \\ -\overline{t_{iy}} \cdot \overline{t_{jx}} & -\overline{t_{iy}} \cdot \overline{t_{jy}} & -\overline{t_{iy}} \\ -\overline{t_{jx}} & -\overline{t_{jy}} & -1 \end{pmatrix};$$

The global energy function can be redefined by combining Eqs. (6) and (9).

$$E(\mathbf{D}) = \tilde{\mathbf{x}}^T (\mathbf{G}_{data} + \mathbf{G}_s) \tilde{\mathbf{x}} - 2\mathbf{H}_{data}^T \tilde{\mathbf{x}} + l_{data}. \quad (11)$$

Computing the minimum value of Eq. (11) is equal to solving $(\mathbf{G}_{data} + \mathbf{G}_s) \tilde{\mathbf{x}} = \mathbf{H}_{data}$. Cholesky decomposition can be used to compute $\tilde{\mathbf{x}}$ directly.

3.3 Feather

In curved surface, obvious seam lines exist between patches. Feather algorithm is designed to smooth seam lines. Firstly, a buffer with the radius l is defined centered at seam lines between patches, as shown in Fig. 3(a). Only points in buffer are involved in feather process. \mathbf{p} is a pixel in the buffer. The distance from \mathbf{p} to the seam line is l' . The depth of \mathbf{p} can be decided by the plane function of S_i , which is defined as d_i . It can also be decided by the plane function of S_j , which is defined as d_j . The depth after feather is determined by d_i and d_j in Eq. (12). The feather result is shown in Fig. 3(b).

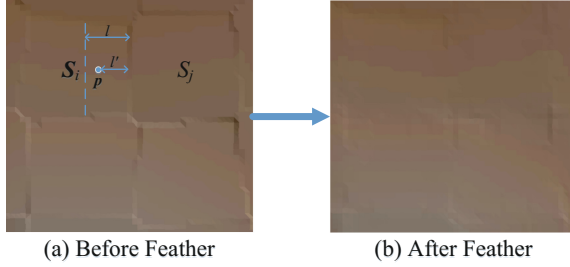


Fig. 3. Feather between patches.

$$d'(\mathbf{p}) = w \cdot d_i + (1 - w) \cdot d_j. \quad (12)$$

where, d' represents the depth after feather; w represents weight, $w = 0.5 + l'/2l$.

4 Experiments

Two experiments were designed to verify the correctness and validity of GDRP. The first experiment used GDRP to refine a disparity image produced by a state-of-the-art 1D label algorithm on Motorcycle images which was provided by Middlebury Benchmark, and compared the original matching accuracy with the refined accuracy, which aimed at testing the validity of GDRP in indoor reconstruction. The second experiment used GDRP to refine a DSM generated by INTS [12] on Wuhan university images, which aimed at testing the validity of GDRP in extensive outdoor reconstruction.

4.1 Indoor Experiment

We chose the disparity image of LW-CNN which ranked the 2nd in Middlebury Benchmark for indoor experiment. The optimization result is shown in Table 2. The first column lists the original image and the ground truth, respectively. The second column lists the original disparity image of LW-CNN and the disparity image after GDRP refinement. The fourth and the fifth rows show the original matching accuracy and the refined matching accuracy, respectively. The last row lists the running time of GDRP.

Table 2. Optimization of the motorcycle disparity image.



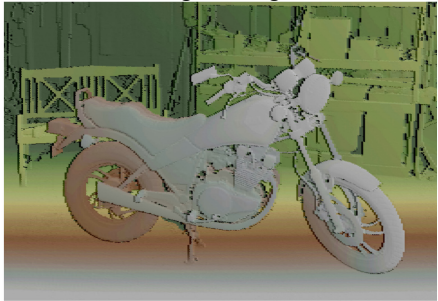

Original Image and Ground Truth		LW-CNN	
			
Original Image		Original Disparity Image	
			
Ground Truth		Refined Disparity Image	
Original Matching Accuracy:		0.97 pixels	
Refined Matching Accuracy:		0.97 pixels	
Running Time:		19.94 s	

Table 2 showed that obvious fronto-parallel bias existed in the original disparity image of LW-CNN. The refined disparity image was continuous and smooth, which showed that GDRP was able to remove fronto-parallel bias efficiently. The matching accuracy didn't change after refinement. It was because: (1) GDRP took original depths as the control, thus the refined accuracy after adjustment should be consistent with the accuracy of the control; (2) fronto-parallel bias was a very slight system error, which had little influence on accuracy assessment. GDRP needs no iterations. It can achieve refined results directly. In the case of single CPU @2.6 GHZ, the running time was only 19.94 s, which was much faster than current 3D label methods in Table 1. It showed that GDRP was fit for fast reconstruction.

4.2 Outdoor Experiment

INTS method [12] was used to reconstruct the DSM of Wuhan University, as shown in Fig. 4. Then, GDRP was used to refine the DSM. In order to show the refinement more clearly, local zoomed reconstruction results are shown in Table 3.

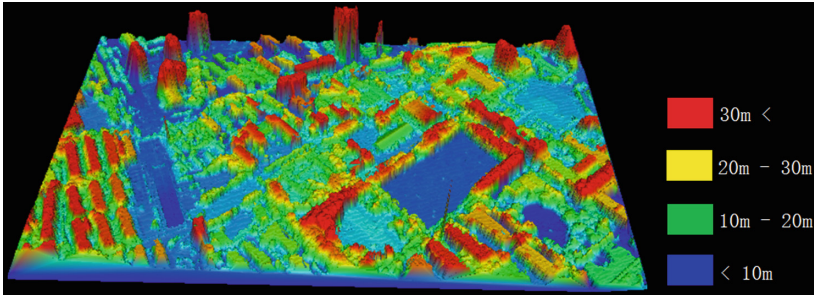


Fig. 4. DSM reconstruction of Wuhan University.

Table 3. Comparison of original DSM and refined DSM.

Original DSM	Refined DSM
<p>Original DSM Accuracy: Refined DSM Accuracy:</p>	<p>0.75 m 0.76 m</p>

INTS is a 1D label algorithm. There were obvious fronto-parallel bias in slanted planes such as roofs, as shown in left column of Table 3. GDRP could smooth fronto-parallel bias effectively, as shown in right column in Table 3. No significant difference of accuracy between original and refined DSMs was detected. It suggested that GDRP only made a small change on the surface to remove the fronto-parallel bias. The DSM accuracy cannot be improved, because GDRP takes the original DSM as the control without any stereo images. The advantage of GDRP lies in the time complexity

which is much lower than current 3D label. The combination of GDRP and state-of-the-art 1D label algorithms enables the efficient acquirement of continuous, smooth 3D reconstruction.

5 Conclusion

This paper proposed a new global depth refinement based on Patches. GDRP transformed the depth refinement into the minimum of a quadratic function, and achieved continuous, smooth depths without changing the original accuracy. It could remove fronto-parallel bias of 1D label algorithms efficiently. Compared with the current 3D label algorithms, GDRP was superior in low time complexity. However, GDRP cannot improve the accuracy. We will introduce stereo images into GDRP to remove fronto-parallel bias as well as improve depth accuracies in the future work.

Acknowledgements. This work was supported by the Chinese Parasol Entrepreneurial Partner Project of Wuhan Engineering Science & Technology Institute (Grant No. gkwt006).

References

1. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comput. Vision* **47**(1), 7–42 (2002)
2. Olsson, C., Ulen, J., Boykov, Y.: In defense of 3D-label stereo. In: *Computer Vision and Pattern Recognition*, pp. 1730–1737. IEEE press, Portland (2013)
3. Li, L., Zhang, S., Yu, X., Zhang, L.: PMSC: PatchMatch-based superpixel cut for accurate stereo matching. *IEEE Trans. Circ. Syst. Vid.* **PP**(99), 1–14 (2016)
4. Zhang, C., Li, Z., Cheng, Y., et al.: MeshStereo: a global stereo model with mesh alignment regularization for view interpolation. In: *International Conference on Computer Vision*, pp. 2057–2065. Santiago (2015)
5. Park, H., Mu Lee, K.: Look wider to match image patches with convolutional neural networks. *IEEE Signal Process. Lett.* **PP**(99), 1–5 (2016)
6. Kim, K.R., Kim, C.S.: Adaptive smoothness constraints for efficient stereo matching using texture and edge information. In: *International Conference on Image Processing*, pp. 3429–3434. IEEE press, Phoenix (2016)
7. Bontar, J., Lecun, Y.: Stereo matching by training a convolutional neural network to compare image patches. *J. Mach. Learn. Res.* **17**(1), 2287–2318 (2016)
8. Hirschmuller, H.: Stereo processing by semiglobal matching and mutual information. *IEEE Trans. Pattern Anal.* **30**(2), 328–341 (2008)
9. Yang, Q.X.: Stereo matching using tree filtering. *IEEE Trans. Pattern Anal.* **37**(4), 834–846 (2015)
10. Pham, C.C., Jeon, J.W.: Domain transformation-based efficient cost aggregation for local stereo matching. *IEEE Trans. Circ. Syst. Vid.* **23**(7), 1119–1130 (2013)
11. Kolmogorov, V., Zabih, R.: Computing visual correspondence with occlusions using graph cuts. In: *International Conference on Computer Vision*, pp. 508–515. Vancouver (2001)
12. Huang, X., Zhang, Y., Yue, Z.: Image-guided Non-local dense matching with three-steps optimization. In: *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, Prague, pp. 67–74 (2016)

13. Zhang, Y., Gong, M., Yang, Y.: Local stereo matching with 3D adaptive cost aggregation for slanted surface modeling and sub-pixel accuracy. In: International Conference on Pattern Recognition, pp. 1–4. Springer press, Tampa (2008)
14. Žbontar, J., LeCun, Y.: Stereo matching by training a convolutional neural network to compare image patches. *J. Mach. Learn. Res.* **17**, 1–32 (2016)
15. Bleyer, M., Rhemann, C., Rother, C.: PatchMatch stereo - stereo matching with slanted support windows. In: British Machine Vision Conference, Dundee, pp. 14.1–14.11 (2011)
16. Heise, P., Klose, S., Jensen, B., et al.: PM-Huber: PatchMatch with huber regularization for stereo matching. In: International Conference on Computer Vision, pp. 2360–2367. IEEE press, Sydney (2013)
17. Barnes, C., Shechtman, E., Finkelstein, A., et al.: PatchMatch: a randomized correspondence algorithm for structural image editing. *ACM Trans. Graph.* **28**(3), 341–352 (2009)
18. Klaus, A., Sormann, M., Karner, K.: Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In: International Conference on Image Processing, pp. 15–18. IEEE press, Hong Kong (2006)
19. Veldandi, M., Ukil, S., Govindarao, K.A.: Robust segment-based stereo using cost aggregation. In: British Machine Vision Conference, Nottingham, pp. 1–11 (2014)
20. Bleyer, M., Gelautz, M.: A layered stereo matching algorithm using image segmentation and global visibility constraints. *ISPRS J. Photogram.* **59**(3), 128–150 (2005)
21. Guney, F., Geiger, A.: Displets: resolving stereo ambiguities using object knowledge. In: Computer Vision and Pattern Recognition, pp. 4165–4175. IEEE press, Boston (2015)
22. Yamaguchi, K., McAllester, D., Urtasun, R.: Efficient joint segmentation, occlusion labeling, stereo and flow estimation. In: Fleet, D., Pajdla, T., Schiele, B., Tuytelaars, T. (eds.) ECCV 2014. LNCS, vol. 8693, pp. 756–771. Springer, Cham (2014). https://doi.org/10.1007/978-3-319-10602-1_49
23. Yamaguchi, K., Hazan, T., McAllester, D., Urtasun, R.: Continuous markov random fields for robust stereo estimation. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7576, pp. 45–58. Springer, Heidelberg (2012). https://doi.org/10.1007/978-3-642-33715-4_4
24. Tani, T., Matsushita, Y., Naemura, T.: Graph cut based continuous stereo matching using locally shared labels. In: Computer Vision and Pattern Recognition, pp. 1613–1620. IEEE press, Columbus (2014)
25. Xu, S., Zhang, F., He, X., et al.: PM-PM: PatchMatch with potts model for object segmentation and stereo matching. *IEEE Trans. Image Process.* **24**(7), 2182–2196 (2015)
26. Li, Y., Min, D., Brown, M. S., Do, M. N., Lu, J.: SPM-BP: sped-up PatchMatch belief propagation for continuous MRFs. In: International Conference on Computer Vision, pp. 4006–4014. IEEE press, Santiago (2015)
27. Besse, F., Rother, C., Fitzgibbon, A., Kautz, J.: PMBP: PatchMatch belief propagation for correspondence field estimation. *Int. J. Comput. Vis.* **110**(1), 2–13 (2014)
28. Bleyer, M., Rother, C., Kohli, P.: Surface stereo with soft segmentation. In: Computer Vision and Pattern Recognition, pp. 1570–1577. IEEE press, San Francisco (2010)
29. Lempitsky, V., Rother, C., Roth, S., Blake, A.: Fusion moves for markov random field optimization. *IEEE Trans. Pattern Anal.* **32**(8), 1392–1405 (2009)