Accurate and Fast Segment-based Cost Aggregation Algorithm for Stereo Matching

Da-Fang Chang, Sih-Sian Wu, Hsin-Yu Hou, Liang-Gee Chen, *Fellow, IEEE*
DSPIC Lab, Department of Electrical Engineering
National Taiwan University, Taiwan
b02901126@ntu.edu.tw, benwu@video.ee.ntu.edu.tw, hsinyuhou13579@gmail.com, lgchen@ntu.edu.tw

***Abstract*—An accurate segment-based cost aggregation method is proposed in this paper. Segment-based methods can speed up the procedure according to the assumption that pixels belonging the same segment should share the same physical property. One of the most critical problems of those methods is the limited pixels within the corresponding segment. To enlarge supporting region of aggregation efficiently, we include similar adjacent segments with a fixed weight instead of a complex one. With introducing the simplified spatial weight for adjacent segments, the proposed method outperforms existing segment-based cost aggregate method[11], the Adaptive Support-weight[3], and the cross-based cost aggregation[10]. Comparing the other aspect of performance, this algorithm also speeds up the computation time by 43%.**

***Keywords*—Stereo Matching, cost aggregation, Segment-based, Superpixel, SLIC.**

I. Introduction

Stereo Matching is a fundamental computer vision problem that has been researched for decades and is still active. It is closely connected to the applications like self-driving cars, 3D scene reconstruction , augmented reality and robotic vision. Matching methods generally perform the following four steps: matching cost computation, cost aggregation, disparity computation and disparity refinement. In this paper, we focus on the cost aggregation step. To accurately estimate the disparity near the depth discontinuities, many cost aggregation methods are proposed. Among those methods, Segment-based methods play an important role because of spatial information. These methods utilize the segmentation of images as the shape of support window for aggregation, and achieve good performance on edge-preserving problem.

Segment-based aggregation methods assume that depth discontinuities occur on boundaries of segments while the segmentation methods are generally based on the RGB differences. This assumption works fine when dealing with an object with similar color, since the object is segmented in the same region, and the scene structure is preserved. However, a problem emerges if the object has a high textured surface. Due to the basic idea of image segmentation, sizes of the segments on that surface tend to be extremely small. Therefore, the support windows are small as well, leading to the reduction of robustness of the matching. Besides, textureless regions, such as a white wall, also cause problem if there is no area restriction on segments. Segments would grow until reaching the boundaries of textureless regions and take a lot of computation time as a result.

978-1-5090-3649-3/17/$31.00

In local algorithms, the disparity estimation at a target pixel aggregates matching cost within a local support window. To estimate disparity accurately, especially near the depth discontinuity, many algorithms adapt their local windows to fit the size and shape of the edge. To reduce the image ambiguity, support pixels with higher similarity to the reference pixel are given higher weights.

This paper introduces a novel segment-based cost aggregation method which is based on the Simple Linear Iterative Clustering (SLIC) algorithm [6]. Unlike other segment-based methods taking only one segment into account, we extend the support region by considering adjacent segments with similar mean color. The problem caused by high textured surfaces would be solved accordingly because the support regions are enlarged. In addition, because SLIC tends to uniform the size of each segment, huge segments that slow down the computation speed would not appear in our algorithm. Consequently, the computation time would not be exceedingly high on textureless regions. Compared with other segment-based methods, the proposed method is able to handle different textured area without complicated computation, thereby achieving a more accurate and fast segment-based cost aggregation. Moreover, it outperforms other local methods like [3] and the cross-based aggregation method (CBCA) [10].

In summary, the contributions of this paper are:

1. An accurate and fast segment-based cost aggregation method is proposed.

2. A simple technique to efficiently enlarge supporting region.

The rest of this paper is organized as follows: Section II summarizes related researches. Our proposed method is presented in Section III. The experimental results and discussion are shown in Section IV. Finally, the paper is concluded in Section V.

II. Related Work

Scharstein and Szeliski [1] classified the stereo matching algorithm into two classes: local and global methods. Generally speaking, global methods provide better performance since they are able to obtain whole information of the image. However, they also require more complex computation like Belief-propagation (BP) [19], GraphCut [20] and Dynamic programming [21]. Semi-global Matching (SGM) is proposed by Hirschmuller [16]. Yang proposed Non-local cost aggreagtion [9] using minimum spanning tree to speed up aggregation

procedure. All the above methods need to store the whole cost volume of the image during the process, which is not practical for high resolution situation.

In contrast, local methods only obtain information from appropriate local window and weights. Yoon and Kweon proposed Adaptive Support-weight [3] which aggregates cost within a local support window with the spatial and the range weights. It achieves great performance but the complicated computation slows down the speed as a trade-off. Zhang et. al. [10] introduced cross-based cost aggregation using the irregular binary weight mask to raise the speed, the performance drops accordingly. Both mentioned methods are still actively used in recent research on stereo vision. Patchmatch stereo [17] uses a simplified form of [3] to overcome the edge-fattening problem. Convolutional neural network(CNN) based methods like MC-CNN [13] and LW-CNN [14] adopt cross-based method [10] as their cost aggregation step, which is influenced by Mei et. al. [22] on the acceleration of the aggregation method.

Recently, segment-based related methods are being increasingly popular. PatchMatch Filter [8] denotes each segment as a superpixel. It uses the SLIC method and assumes the smoothness of disparity within same superpixels. Segment-based algorithms are employed by Lee et. al. [4] and Song et. al. [5] as the cost aggregation part of their complete stereo matching system. The methods mentioned above support each pixel with exactly one segment, therefore some textured regions would be challenging for them without any refinement. In this paper, we do not directly compare our method with them, since we are mainly focused on the cost aggregation step, and their systems comprise all four stereo matching steps above-mentioned.

Another local method [11] also makes use of the information obtained from image segmentation, and combines it with the Adaptive Support-weight[3]. This method uses only one segment for aggregation as well. However, it defines a local window for more information. That is, pixels not belong to the support segment but within the support window are aggregated. Those pixels are given a weight function that is same as the one used in adaptive support-weight. This method outperforms [3], but takes even much more times than it because of the segmentation computation.

The proposed method is also based on the image segmentation information. We choose the SLIC algorithm [6] to decompose the input image. However, different from [3] and [11] that complex weighting function on support pixels is employed for edge-preserving, our method uses only one fixed number to weight pixels in adjacent segments. As a result, the computation complexity is substantially reduced. Adjacent Segments are taken into account in our algorithm, which leads to a better performance compared with [11].

III. Segment-based Cost Aggregation

A. Algorithm Overview

Fig. 1 illustrates the procedure of the proposed aggregation method. The basic idea is to utilize information obtained from the SLIC segmentation. The advantages of SLIC over other superpixel algorithms are the compactness and the regular

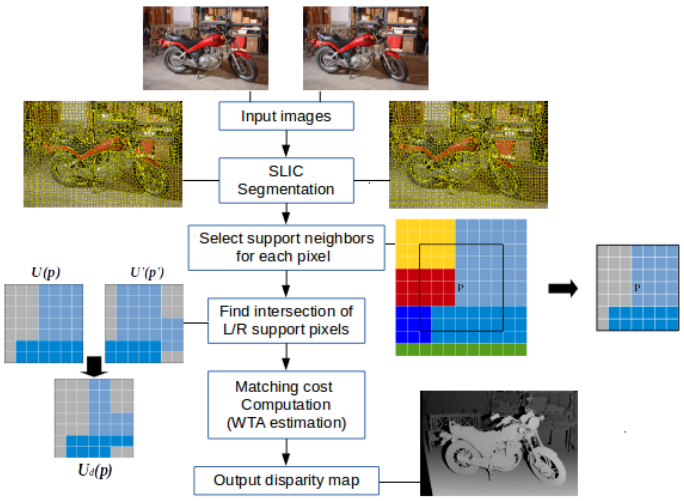


Fig. 1: Block diagram of the proposed algorithm.

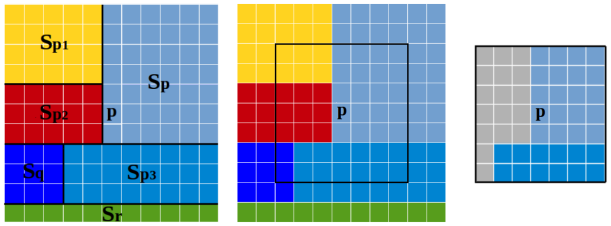


Fig. 2: Pixel P is the target pixel in the center, and S_p denotes the target segment. S_{p_1}, S_{p_2} and S_{p_3} are adjacent to the S_p, and only S_{p_3} has similar color with S_p. S_q and S_r are not neighboring segments of S_p.

shapes and sizes of the segments. Similar to [8] and [11] that hypothesize the smoothness of disparity in each segment, our algorithm is based on the assumption that depth discontinuities occur only on segment boundaries. Therefore, to select for each pixel an adaptive support, only neighboring pixels in the same segment as the target are considered. A fixed size support window is also adapted, and the support pixels for the target pixel are then determined with the segment and window constraints. Despite that the size of support windows for left and right image are the same, the shape of the segment inside the window is very likely to be different. Inspired by the cross-based cost aggregation (CBCA) [10], we perform an intersection of reliable pixels from left and right window to prevent outliers in the right image from reducing the accuracy of disparity estimation. With support pixels successfully being selected, the matching cost is then aggregated. The disparity is estimated using Winner-Take-All strategy and the depth map is thus constructed.

B. Select Support Pixels

The support pixels are selected according to the information of segments. For each pixel p in the image, there is a target segment Sp which $p \in Sp$. Pixels in the same segment as p are aggregated. However, due to the SLIC algorithm which tends to uniform the size of each segment, we may take few pixels into account in the textureless region like illustrated in Fig. 3,

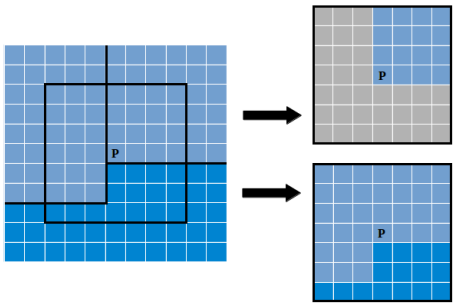


Fig. 3: The low textured condition case with segmentation. Upper figure is the supporting region without adjacent segments. Lower one is the actual supporting region detected by the proposed method.

and it would cause ambiguity when estimating the disparity. When the target pixel p is just on the boundary of segment, support pixels obtained from the target segment are very few. To enlarge the support region, segments with similar color are considered. As a result, accuracy on low textured region is improved. Therefore, to enlarge the support region for the pixel in textureless region, the neighboring segments that are similar in color with target segment are under consideration as well. That is, $\forall S_{p_n} \in N_p$, if

$$\sum_{i \in S_{pn}} \frac{\mid I_{ni} - I_p \mid}{\mid\mid S_{p_n} \mid\mid} \leq \tau, \tag{1}$$

then the segment S_{p_n} is considered as support region. A neighboring segments list N_p is pre-constructed for each segment, a threshold τ is defined to decide whether the neighboring segment should be considered or not, and I means the intensity of pixels. Those pixels that are selected in the neighboring segments are given a weight ω_n, so that they have less influence than pixels in center segment.

In addition to the segments, fixed-size window is also used in our method. The purpose of window is to limit and regularize the shape and size of support region for each pixel, so that the matching cost can be aggregated more efficiently. Fig. 2 demonstrates an example of selected support pixels. S_{p_1}, S_{p_2} and S_{p_3} are all neighboring segments of target segment Sp. Suppose that segments with different blue colors in Fig. 2 are all analogous; that is, they are all satisfy (1). However, since segment S_q is not adjacent to S_p, pixels that belong to S_q are not aggregated. Thus, S_p and S_{p_3} are the only segments considered. Therefore, despite that S_q is analogous with S_p in color, pixels in S_q are excluded. After employing a window on the target pixel, the aggregation step for one image is complete.

After excluding pixels outside the window, the support region for p is determined. An intersection of support regions $U(p)$ and $U'(p')$ is finally performed, which yields region $U_d(p)$ with

$$U_d(p) = \{(x,y) | (x,y) \in U(p), (x-d,y) \in U'(p')\}. \tag{2}$$

C. Matching Cost Aggregation

In our experiment, the raw matching cost $C_d(p)$ between pixel p and p' is computed as

$$c_d(p,p') = \alpha \mid I_p - I_{p'} \mid + (1-\alpha) Census(p,p',w_c). \tag{3}$$

As we mentioned in the previous section, support pixels in the neighboring segments are given a weight ω_n. Therefore, the normalized matching cost is aggregated as

$$C_d(p,p') = \frac{\sum_{p \in U_d(p), p' \in U_d(p')} \omega_{pro}(p,q) c_d(q)}{\sum_{p \in U_d(p), p' \in U_d(p')} \omega_{pro}(p,q)} \tag{4}$$

Compared with the equation in [3], which is

$$E(p,p') = \frac{\sum_{p \in N_p, p' \in N_{p'}} \omega(p,q)\omega(p',q') c_d(q)}{\sum_{p \in N_p, p' \in N_{p'}} \omega(p,q)\omega(p',q')} \tag{5}$$

and the weight

$$\omega(p,q) = exp(-(\frac{\Delta c_{pq}}{\gamma_c} + \frac{\Delta g_{pq}}{\gamma_p})), \tag{6}$$

, the proposed method uses only one fixed floating-point number ω_n to weight different pixels and avoids any exponential terms. The proposed weight $\omega_{pro}(p,q)$ is defined as follow

$$\omega_{pro}(p,q) = \begin{cases} In \quad the \quad same \quad segment & 1; \\ In \quad similar \quad adjacent \quad segments & \omega_n; \\ Others & 0. \end{cases} \tag{7}$$

IV. Experimental Results

We use 15 quarter size training image pairs from Middlebury [15] to evaluate local aggregation methods including adaptive support-weight [3], segment support [11], CBCA [10] and our method. The parameters are kept constant for all image pairs, i.e., window size = 11 $\times$ 11, ω_n = 0.75 and τ = 100. For [3] and [11], the window size is also set to 11 for fair comparison. The arm length L for [10] is set to 5, since the arms grow in four different directions. [11] is implemented with SLIC method in this experiment rather than the Mean-Shift algorithm[23] used in the original paper, so that both [11] and our method are based on same segmentation method.

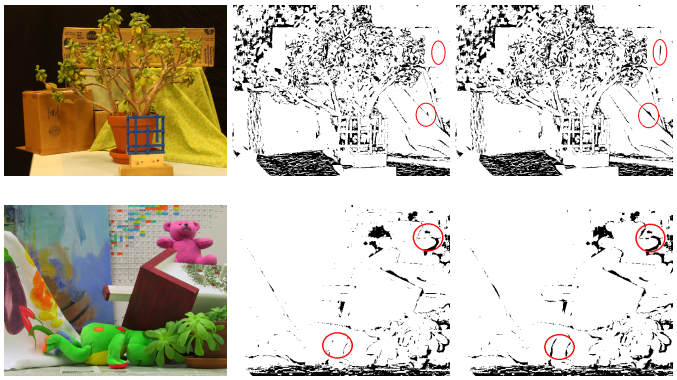
Table I presents the evaluation results for Middlebury dataset. Different from the old dataset used by [3], [11] and [10] in their experiments, this new dataset is more challenging, and the performance ranking is slightly different from the results of old dataset as a consequence. The error threshold for the disparity value is 1.0. In our experiment, [3] outperforms [11] slightly. The proposed method not only achieves a better performance than [3] and [11], but also takes less computation time. Compare to [11], the computation time of 15 image pairs is reduced by 43%. It outperforms [10] with an obvious improvement, but not as fast as [10] is.

Fig. 6 and 7 are the visual presentations of our results. The disparity maps of non-occlusion region for ArtL, Jadeplant, Pipes and Recycle (from top to bottom) are shown in Fig. 6, and the error maps are shown in Fig. 7 with errors are greater than 1.0. We can observe that in low textured regions

TABLE I: Error rate of Middlebury dataset within various local cost aggregation methods with the error threshold equals to 1.0.

labels	Adirodack	ArtL	Jadeplant	Motorcycle	MotorcycleE	Piano	PianoL	Pipes	Playroom
Adapt. Weights [3]	12.08_1	19.12_3	26.37_2	11.41_3	14.61_3	20.93_1	44.15_3	15.00_3	28.92_3
Ours	13.34_2	15.80_1	23.43_1	10.08_2	13.19_2	21.68_2	38.68_1	12.90_1	27.97_1
Segment suppor [11]	14.42_3	16.98_2	28.46_3	9.95_1	11.44_1	22.17_3	39.42_2	13.44_2	28.76_2
CBCA [10]	23.59_4	25.51_4	30.79_4	16.32_4	20.70_4	27.52_4	53.1_4	20.32_4	35.83_4

labels	Playtable	PlaytableP	Recycle	Shelves	Teddy	Vintage	Average	Weighted Avg.	Total time(sec)
Adapt. Weights [3]	34.83_1	20.40_2	16.78_2	42.72_1	8.20_1	36.48_2	23.17_2	20.32_2	19373.15_4
Ours	36.60_2	19.93_1	15.97_1	45.37_3	8.59_2	36.05_1	22.54_1	19.78_1	2988.69_2
Segment support [11]	37.90_3	22.01_3	17.22_3	45.25_2	9.61_3	37.00_3	23.60_3	20.79_3	5253.80_3
CBCA [10]	42.03_4	28.48_4	24.48_4	47.96_4	12.53_4	46.36_4	29.34_4	26.19_4	$1471,93_1$

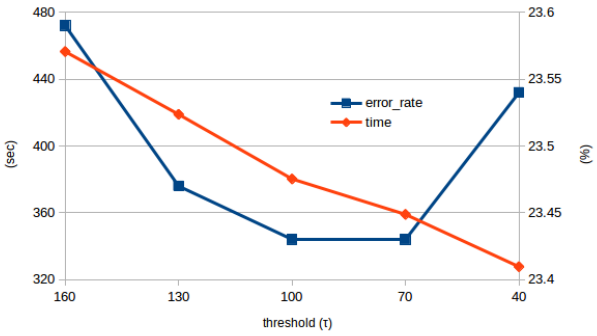


(a) Reference Image (b) $\tau = 100$ (c) $\tau = 160$

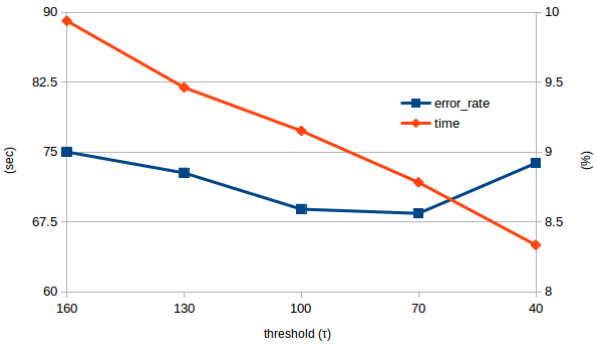
Fig. 4: (a)Reference image, (b) Error map when $\tau = 100$ and (c) Error map when $\tau = 160$. Red circled regions highlight the difference between two τ(threshold) values.

like the sculpture in ArtL and the background in Jadeplant, the proposed method estimates the disparity most correctly. The proposed method also properly handles the high textured regions like the background in Pipes, while CBCA fails on some pixels there. The background to the left of Recycle is a high textured region with repetitive patterns. From Fig. 7, the proposed method is able to deal with that situation without any error and outperforms the others.

We also evaluate the performance of the proposed method with different parameter settings. Fig. 4 shows the comparison between two different values of τ. From the red circles in the error map, it is easy to find that as τ goes higher, errors pixels in discontinuity area are increased. Fig. 5 demonstrates that the higher the threshold τ is, the speed is slower. This is intuitive because more neighboring segments are taken into account, and more computation is required. However, the performance does not improve when more pixels are considered. The reason is that higher τ accepts segments that are dissimilar to the target segment, which conflicts with our assumption that only similar segments are included. The performance is also undesirable when τ is too low. This situation restricts the assistance from neighboring segments, thus the high textured regions cause the ambiguity. The experimental results suggest that the proposed method can reach the best accuracy when the value of τ is set between 70 to 100.



(a) Jadeplant

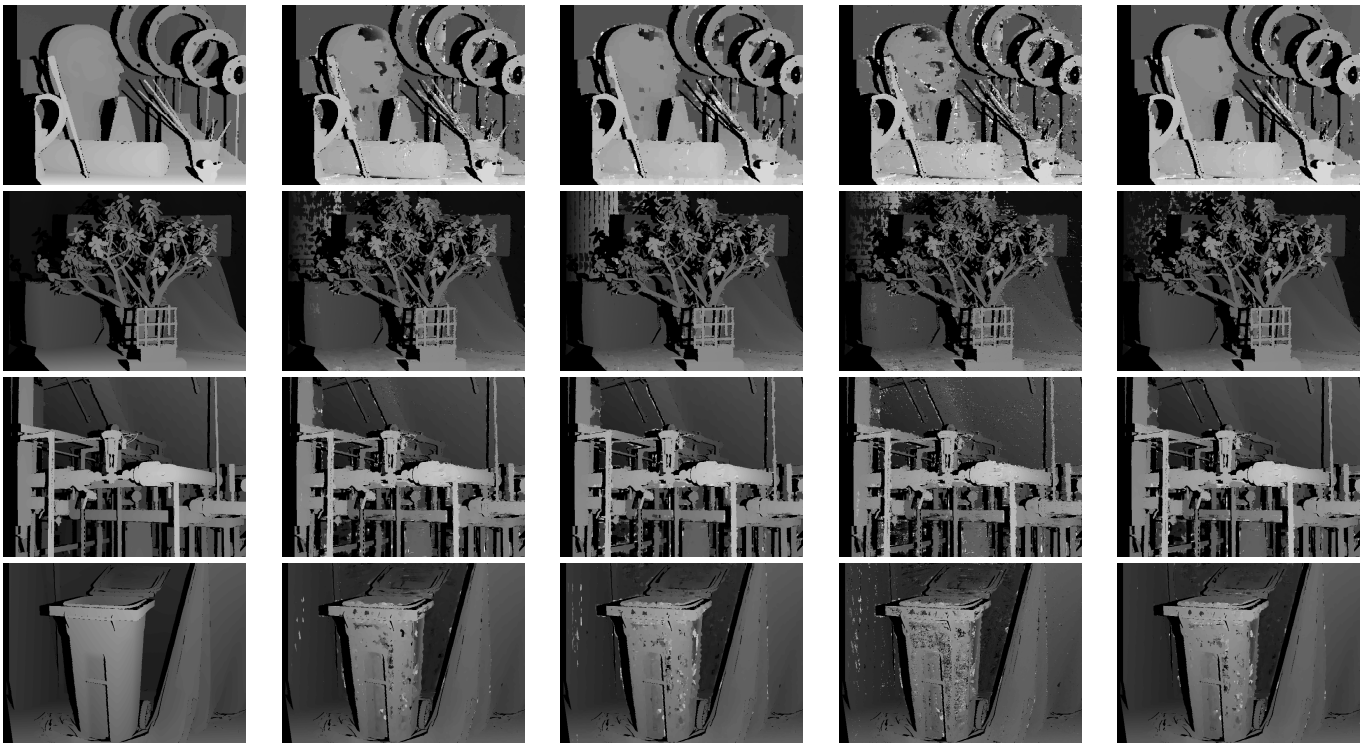


(b) Teddy

Fig. 5: Performance and run time evaluation of the proposed method when using different value of τ. ω_n is fixed at 0.75.

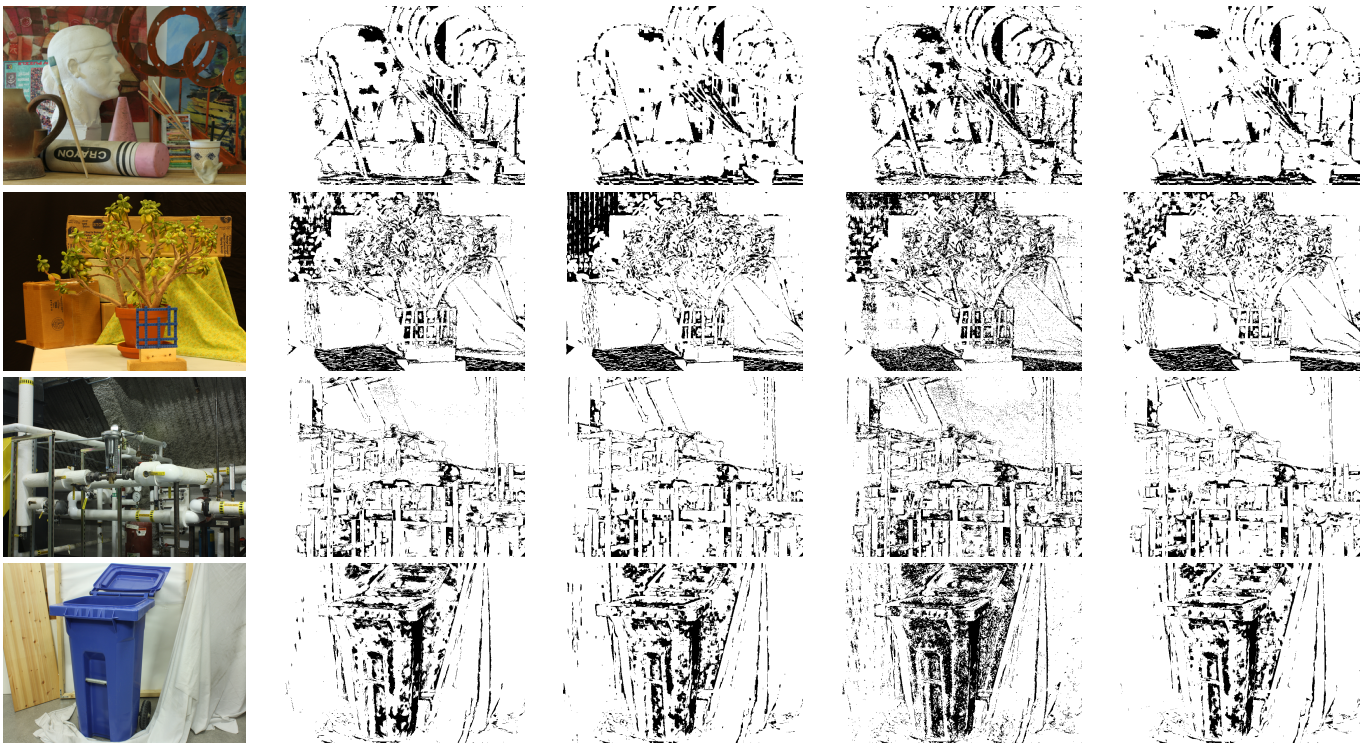
V. Conclusion

An accurate segment-based cost aggregation method is proposed. This algorithm leverages the smoothness of the segmented region and enlarge the support region efficiently. The support region includes similar adjacent segments with a fixed weight employed on them. The discussion about the threshold to determine whether the neighboring segments is included is made. According to the results, this algorithm outperforms Adaptive support-weight [3] and CBCA [10] and the existing segment-based cost aggregation method [11]. Furthermore, the proposed algorithm can further speed up the computation by 43% compared to [11].



(a) GT (b) Adapt. Weight (c) Seg. Support (d) Cross-based (e) Ours

Fig. 6: (a)Ground truth and (b-e) disparity maps of various algorithms on the Middlebury 3.0 benchmark in non-occlusion region.



(a) Reference Image (b) Adapt. Weight (c) Seg. Support (d) Cross-based (e) Ours

Fig. 7: (a) Left images and (b-e) bad pixels of various algorithms on the Middlebury 3.0 benchmark whose errors are greater than 1.0.

REFERENCES

[1] H. Kopka and P. W. Daly, *A Guide to LaTeX*, 3rd ed. Harlow, England: Addison-Wesley, 1999.

[2] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47:7-42, 2002.

[3] K.-J. Yoon and I.-S. Kweon. Adaptive support-weight approach for correspondence search. *TPAMI*, 28(4):650-656, 2006.

[4] Lee, Sehyung, et al. "Robust stereo matching using adaptive random walk with restart algorithm." Image and Vision Computing 37 (2015): 1-11.

[5] Song, Kechen, et al. "Noise robust image matching using adjacent evaluation census transform and wavelet edge joint bilateral filter in stereo vision." Journal of Visual Communication and Image Representation 38 (2016): 487-503.

[6] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Ssstrunk, SLIC Superpixels Compared to State-of-the-art Superpixel Methods, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, num. 11, p. 2274 - 2282, May 2012.

[7] 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," 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, 2015, pp. 2057-2065.

[8] J. Lu; Y. Li; H. Yang; D. Min; W. Eng; M. Do, "PatchMatch Filter: Edge-Aware Filtering Meets Randomized Search for Visual Correspondence," in IEEE Transactions on Pattern Analysis and Machine Intelligence , vol.PP, no.99, pp.1-14

[9] Q. Yang, "A non-local cost aggregation method for stereo matching," 2012 IEEE Conference on Computer Vision and Pattern Recognition, Providence, RI, 2012, pp. 1402-1409.

[10] K. Zhang, J. Lu and G. Lafruit, "Cross-Based Local Stereo Matching Using Orthogonal Integral Images," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 19, no. 7, pp. 1073-1079, July 2009.

[11] Tombari, Federico, Stefano Mattoccia, and Luigi Di Stefano. "Segmentation-based adaptive support for accurate stereo correspondence." Advances in Image and Video Technology (2007): 427-438.

[12] Keselman, Leonid, et al. "Intel RealSense Stereoscopic Depth Cameras." arXiv preprint arXiv:1705.05548 (2017).

[13] Zbontar, Jure, and Yann LeCun. "Stereo matching by training a convolutional neural network to compare image patches." Journal of Machine Learning Research 17.1-32 (2016): 2.

[14] H. Park; K. M. Lee, "Look Wider to Match Image Patches with Convolutional Neural Networks," in IEEE Signal Processing Letters , vol.PP, no.99, pp.1-5

[15] http://vision.middlebury.edu/stereo/data/scenes2014/

[16] Hirschmuller, Heiko. "Stereo processing by semiglobal matching and mutual information." IEEE Transactions on pattern analysis and machine intelligence 30.2 (2008): 328-341.

[17] M. Bleyer, C. Rhemann, and C. Rother. "Patchmatch stereo - Stereo matching with slanted support windows." In Proc. of BMVC, 2011.

[18] Besse, Frederic, et al. "Pmbp: Patchmatch belief propagation for correspondence field estimation." International Journal of Computer Vision 110.1 (2014): 2-13.

[19] Sun, Jian, Nan-Ning Zheng, and Heung-Yeung Shum. "Stereo matching using belief propagation." IEEE Transactions on pattern analysis and machine intelligence 25.7 (2003): 787-800.

[20] Boykov, Yuri, Olga Veksler, and Ramin Zabih. "Fast approximate energy minimization via graph cuts." IEEE Transactions on pattern analysis and machine intelligence 23.11 (2001): 1222-1239.

[21] Veksler, Olga. "Stereo correspondence by dynamic programming on a tree." Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on. Vol. 2. IEEE, 2005.

[22] Mei, Xing, et al. "On building an accurate stereo matching system on graphics hardware." Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on. IEEE, 2011.

[23] Comaniciu, Dorin, and Peter Meer. "Mean shift: A robust approach toward feature space analysis." IEEE Transactions on pattern analysis and machine intelligence 24.5 (2002): 603-619.