Robust segment-based Stereo using Cost Aggregation

Veldandi Muninder[1]
veldandi.muninder@nokia.com
Ukil Soumik[2]
soumik.ukil@nokia.com
Govindarao Krishna[2]
krishna.govindarao@nokia.com

[1] Nokia Technologies
Sunnyvale
California, USA

[2] Nokia Technologies
Bangalore
India

Abstract

A general approach in segment-based stereo methods is to segment an image and estimate a 3-D plane for each segment, or group of segments. Inherently, such methods are sensitive to segmentation parameters and intolerant to segmentation errors. We propose a novel algorithm for generating sub-pixel accurate disparities on a per-pixel basis, thus alleviating the problems arising from methods that estimate disparities on a per-segment basis. An initial disparity map, generated using any fast local method, is used in conjunction with a color-based segmentation map to generate a set of planes. The cost of assigning each plane to every pixel is computed, and the powerful spanning-tree based cost aggregation approach is used to assign a plane label to each pixel. The steps of plane estimation and assignment are repeated in an iterative framework to enhance the results. An evaluation on the Middlebury database demonstrates the robustness of our method to segmentation parameters and disparity initialization. We also show that the disparity map accuracy for the proposed method compares favorably with most state-of-the-art approaches, being ranked 1st in average percentage of bad pixels, and 11th overall.

1 Introduction

Stereo correspondence is an extensively researched topic in computer vision, given its many applications. A wide variety of local and global stereo matching algorithms have been proposed over the years, as listed in the comprehensive evaluation by Scharstein and Szeliski [15]. Recently, a new class of local algorithms using matching cost aggregation have come to the fore [13, 17, 20]. Compared to global methods, they are computationally more efficient, and compared to other local methods, they are generally more accurate and robust.

Cost aggregation is an important step in local stereo matching [15] and is traditionally performed locally over windows with constant disparity. A fixed support bilateral filter based cost aggregation proposed by Yoon and Kweon [20] was very effective in preserving depth edges. Recently, full image support based cost aggregation methods [9, 12, 13, 17] have gained considerable attention due to their high accuracy and low computational complexity. Most of these methods, however, output fronto-parallel disparity levels.

There are several stereo algorithms that output sub-pixel disparity maps, with every map being modeled by 3-D planes at a segment level. Some methods generate sub-pixel accuracy at a post processing step, as in Yang et al. [18]. Sub-pixel accuracy can be considered by either using an extended label space [7] or using plane equations as labels [6]. Many algorithms use color segmentation [5] combined with an initial disparity estimate to compute an initial set of plane equations, and then refine these equations [4, 8, 10, 16]. A few approaches have been recently proposed [3, 11, 21], that estimate the plane assignment at every pixel and output the disparity map with sub-pixel precision. In [21], plane equations are estimated using least squares from horizontal and vertical slants, and in [3] the plane equations are estimated using a randomized algorithm. After estimating the set of planes, the plane matching cost at every pixel is aggregated within a window. All methods that estimate sub-pixel disparity [4, 8, 10, 16] are computationally complex. For example, the recent method of Bleyer et al. [3] takes 1 minute on an average to compute a disparity map on the Middlebury.

In this paper, we propose a novel method to compute sub-pixel precision disparity maps using the minimum spanning tree (MST) based cost aggregation framework. The proposed method, although based on segments, is shown to be highly robust to segmentation errors and parameter variations.

We use the full-image-support based cost-aggregation framework of Yang [17] and generate a sub-pixel accurate disparity map by estimating a plane equation per pixel. Since the disparity at every pixel is modeled by a plane equation, the goal is to ensure that all pixels belonging to a planar surface are labeled with the same plane equation. We show that using a reduced and refined set of planes as candidate labels in the aggregation framework ensures homogeneous labeling within a color segment. Experimental results on the Middlebury set [14] demonstrate the high accuracy of the proposed method, both at pixel and sub-pixel precision. The robustness of our method to variations in the input color segmentation is also demonstrated.

The rest of the paper is organized as follows. In section 2 we discuss the motivation for the proposed method. The method is described in section 3, experimental results are demonstrated in section 4, and our conclusions are presented in section 5.

2 Segment-based Stereo

Most segment based stereo methods estimate disparity by modeling color segments as 3-D planes [4, 8, 10, 19]. Two main dependencies of these methods on the underlying segmentation algorithm are: size of segments used for estimating planes, and assignment of a single plane to the whole segment. Specifically, in the case of under-segmentation, there is a higher chance of merging multiple objects (with multiple plane surfaces) into a single segment. Consequently, planes estimated using these segments are erroneous. The effect propagates to the disparity map, wherein a larger segment encompassing multiple objects is incorrectly represented by a single disparity plane. In the over-segmentation case, which gives smaller color segments, the estimated planes may be unreliable, leading to an inaccurate disparity map. Popular segment based methods try to solve this problem by re-fitting the planes on grouped segments, in an iterative manner [4, 8, 10].

3 Proposed Method

Fig. 1 displays the flow diagram of the proposed method, with the following subsections describing each of these steps in detail.

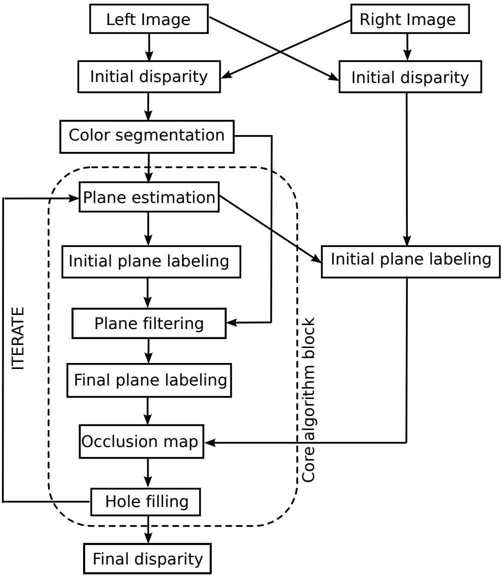


Figure 1: Algorithm flow diagram

3.1 Estimation of initial plane set

Our method takes left and right view disparity maps with fronto-parallel disparity labels as inputs. The disparity maps may be generated using any local or global algorithm. The left and right disparity maps are cross-checked for consistency, to flag occluded and inconsistent pixels. In addition, color segments are obtained on the left(reference) image using mean-shift segmentation [5]. Initial set of planes are determined using consistent disparities within a segment [8, 10]. The disparity plane corresponding to a segment is expressed by the following function:

$$d(p_x, p_y) = A.p_x + B.p_y + C \tag{1}$$

where (p_x, p_y) denotes the pixel location, d denotes the initial disparity map and A, B and C are the plane parameters. All the pixels having consistent disparity within a segment S are used to compute a least squares estimate of the plane parameters corresponding to that segment. Plane equations and plane labels are equivalent, and are used interchangeably, within the scope of this article.

3.2 Plane labeling using cost aggregation

The basis of our method is an assumption that the initial set of planes is a superset of the actual set of planes that describe the scene. The subsequent iterations of the core algorithm block enables the generation of a more accurate initial set of planes due to the feedback loop, as described in later sub-section. Plane based modeling of the scene is equivalent to assigning each pixel to its correct plane label. We compute a pixel-wise cost volume over this label set and generate a labeling using non-local cost aggregation, as described in [17]. An advantage of this formulation is that for a planar segment with similar colors, a single plane label can accrue more aggregated support than multiple fronto-parallel labels. This leads to a smoother output disparity in plane regions, as opposed to the step like effect when using only fronto-parallel labels.

Unlike traditional segment-based methods that compute the cost volume at a segment level [8, 10], we compute the cost volume at a pixel level. Although the method of Bleyer et al. [3] computes the cost volume at a pixel level, they use a random initial plane set and aggregate the subsequent cost volume over a fixed local support.

The pixel matching cost is a weighted sum of the absolute color difference and the absolute gradient difference. $\rho(p, p')$ computes the pixel dissimilarity between the pixels p and p' as

$$\begin{aligned}\rho(p,p') = (1-\alpha)\min(\left|I_p - I'_{p'}\right|, \tau_{abs}) \\ + \alpha\min(\left|\nabla I_p - \nabla I'_{p'}\right|, \tau_{grad})\end{aligned} \tag{2}$$

where $0 \leq \alpha \leq 1$, I and I' denote the left and right images, ∇I denotes gradient of image I, and τ_{abs} and τ_{grad} denote the clamping thresholds for color and gradient costs respectively. The matching cost $D(p,l)$ for a pixel p, located at (p_x, p_y) on the left image, for a plane label l with equation $A_l x + B_l y + C_l = z$, is given by

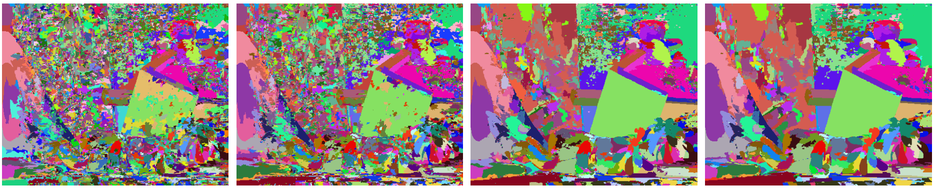
$$D(p,l) = \rho(p,q) \tag{3}$$

where q is the pixel located at $(p_x - A_l p_x - B_l p_y - C_l, p_y)$ on the right image. This cost volume is aggregated on a MST computed on the left image. Winner take all (WTA) on the aggregated cost volume is used to assign plane labels, leading to sub-pixel precision disparity for every pixel. Similarly, the right view disparity map is obtained by computing the cost volume using right view plane labels (Eq. 3), and aggregating the cost volume on the right image MST. Instead of computing candidate planes using the right image, these can be generated using the segmentation and disparity from the left view. Specifically, given a disparity map d, for a segment S in the left image, the plane equation is obtained using $\{x_i, y_i, d(x_i, y_i)\} \mid \forall (x_i, y_i) \in S$, whereas the corresponding plane equation for the right view is obtained using $\{x_i - d(x_i, y_i), y_i, d(x_i, y_i)\}$ (shift in x). Disparity maps for both views are generated from the plane labels, and the occlusion map is updated by consistency checking. Owing to the increased precision of the underlying disparity maps, a stricter sub-pixel threshold is used in the consistency check.

3.3 Filtering of planes and label update

The number of labels in the initial set is of the order of the number of segments, with a plane estimate for every segment. The subsequent labeling, described in subsection 3.2, is used along with the color segmentation to filter and generate a reduced set of planes. This framework of plane filtering followed by re-labeling leads to a more accurate disparity map.

In each color segment, the plane label retained is the mode of the distribution of plane labels within that segment. Unlike an analysis of the global distribution of plane labels, this local analysis ensures that only dominant planes within the color segments are retained.



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

Figure 2: Plane labeling after intermediate steps (Teddy):(a) Labeling - initial label set; (b) Labeling - reduced label set; (c) Labeling - reduced label set and support weighted matching cost; (d) Labeling - occlusion filling.

For example, a global histogram could wrongly discard locally prominent plane labels. The subsequent labeling, generated using cost aggregation and WTA, is more locally homogenous, as shown in Fig. 2(b), leading to a more accurate disparity map. The robustness of our method is due to the over completeness of the initial plane set, i.e. it contains all planes in the scene, in addition to some other insignificant or incorrect planes.

In addition to filtering of the initial plane set, segment analysis is also used to modify the plane matching cost. We weigh the pixel matching cost by a support factor, where the support factor is derived from the distribution of plane labels within the color segment, as follows:

$$D(p,l) = \rho(p,q)e^{\frac{-n_{l,s}}{\tau n_s}} \tag{4}$$

where n_s is the number of pixels in the segment s that contains p, $n_{l,s}$ is the number of pixels in the segment s that are assigned plane label l, and τ is a constant. This cost update adds a bias towards locally dominant labels, whilst suppressing labels with smaller support. As shown in Fig. 2(c), the labeling derived from the modified cost volume with the reduced set of labels is more locally homogeneous than those shown in Fig. 2(a) and 2(b).

3.4 Occlusion filling

The left and right disparity maps are cross-checked for consistency, and holes are filled using a method similar to [17]. Using the filtered labeled set from subsection 3.3, we build a new cost volume:

$$D_{fill}(p,l) = \begin{cases} |d-(A_l p_x + B_l p_y + C_l)|e^{\frac{-n_{l,s}}{\tau n_s}} \\ \qquad\qquad\qquad\qquad : \text{p is consistent} \\ 0 \qquad\qquad\qquad\quad\;\; : \text{otherwise} \end{cases} \tag{5}$$

The modification of data cost using a segment support factor is an enhancement over [17]. Aggregation of this cost is done on the reduced set of labels, followed by a WTA step to obtain a disparity map with occlusions filled. In this formulation, neighboring plane labels of similar colors propagate to fill the holes, leading to sub-pixel precision disparities at these locations. Additionally, the support factor encourages labeling the occlusion region with a dominant plane label in the color segment the occlusion belongs to. As shown in Fig. 2(d), the labeling derived from the occlusion filling step is more homogeneous compared to Fig. 2(c), especially near occlusions.

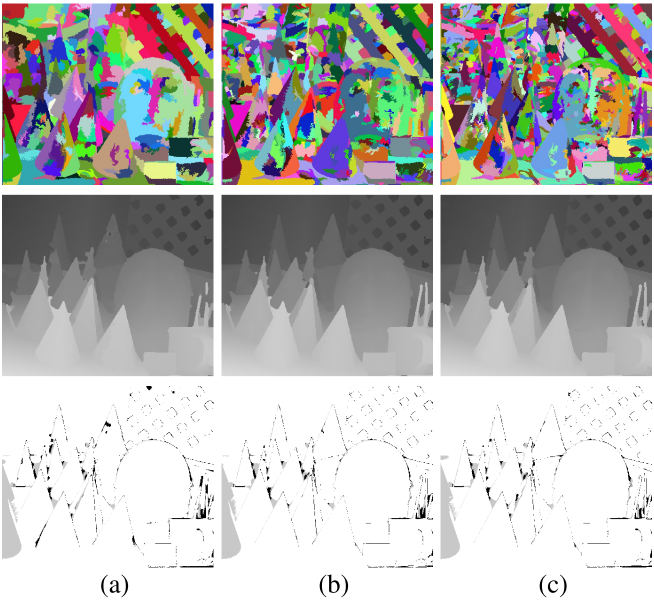


Figure 3: Effect of segmentation variance on disparity (Cones): (a) 266 segments, error = 2.58, rank = 23; (b) 507 segments, error = 2.10, rank = 3; (c) 836 segments, error = 2.16, rank = 6;

Algorithm	Rank	Avg% bad pixels	Tsukuba			Venus			Teddy			Cones		
			nocc	all	disc	nocc	all	disc	nocc	all	disc	nocc	all	disc
Our Method-*WTA*	21	4.75	2.36	2.83	10.5	0.13	0.28	1.82	4.70	5.96	12.8	2.32	6.47	6.87
Our Method-*MST*[1]	11	3.58	1.99	2.39	8.59	0.12	0.21	1.68	2.19	3.73	7.02	2.16	6.52	6.37
Patch Match[3]	28	4.59	2.09	2.33	9.31	0.21	0.39	2.62	2.99	8.16	9.62	2.47	7.80	7.11
MST[17]	43	5.48	1.47	1.85	7.88	0.25	0.42	2.60	6.01	11.60	14.30	2.87	8.45	8.10

Table 1: Results for our method with WTA and MST [17] initialization, PatchMatch [3] and MST [17] on the Middlebury benchmark set.

.

3.5 Iterative refinement

As shown in Fig. 1, the core algorithm block of plane labeling can be iterated on, in a feedback loop. The sub-pixel precision disparity map generated from the final plane labeling is used along with the initial color segments to re-estimate the set of planes. The higher accuracy of the input disparity leads to more accurate plane estimation, and subsequently to more accurate disparity maps. In Section 4 we demonstrate the improvement in accuracy over increasing iterations. While a convergence criteria based on change in absolute disparities between iterations can be used, we have empirically found that convergence is reached in 3 iterations.

4 Experimental Results

In this section we report results using the proposed method on the Middlebury set [14] and also on natural scenes. For all experiments, the parameters α, τ_{abs} and τ_{grad} in Eq. 2 are set to 0.89, 7, and 2, respectively. The parameter τ in Eq. 4 is set to 2, and to 4 in Eq. 5. The results reported in Table 1 are generated with the following mean-shift segmentation

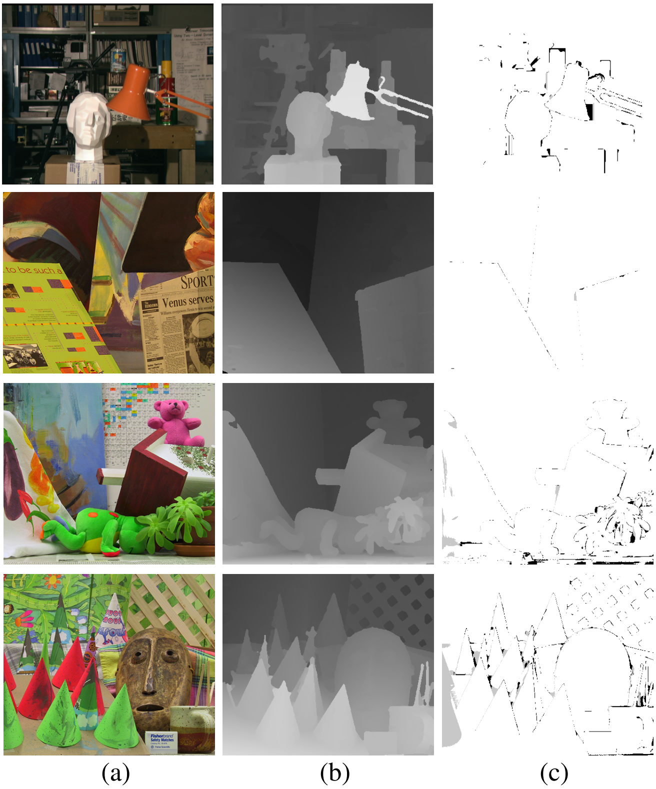


Figure 4: Results on the Middlebury benchmark set: (a) Left Image; (b) Disparity using proposed method(MST [17] initialization); (c) Error maps (absolute error ≥ 1.0)

parameters: $\sigma_{color} = 4.5$, $\sigma_{spatial} = 10$ and minimum segment size = 0.01% of image size.

The effect of segmentation parameter variation is demonstrated on the Cones image, as shown in Fig. 3. The minimum segment size parameter in mean-shift segmentation [5] is varied to generate varying segmentation maps. Our method is robust to these variations, resulting in accurate disparity maps in all instances. Observing the bottom right corner of the disparity maps in Fig. 3(a), 3(b), the pencils belong to a segment that spans multiple objects. Despite this leakage our algorithm is able to recover and assign correct disparities. The methods of [8] and [10] inherently generate labels on a per-segment basis, leading to a lower tolerance for such variations. In our method, the input segmentation is used only to generate an initial set of planes, and is not a cardinal constraint in the final plane labeling. Instead, the color appearance based cost aggregation framework ensures that plane labels have local coherence based on color similarity.

We show how the quality of the initial disparity map affects disparity estimation by considering two different methods for creating input fronto-parallel disparity maps. First, we initialize our method with a disparity map generated using simple WTA, without cost aggregation, on the cost volume of Eq. 2. The first row of Table 1 displays the result for 3 iterations of our algorithm with this initialization. The overall Middlebury rank with this initialization is 21. Next, we initialize our method with the disparity generated by [17]. Three

[1] These results are published in the permanent Middlebury stereo evaluation table under the name *SegAggr*.

	WTA initialization													
Iterations	Rank	Avg% bad pixels	Tsukuba			Venus			Teddy			Cones		
			nocc	all	disc	nocc	all	disc	nocc	all	disc	nocc	all	disc
1	115	9.27	2.55	2.91	11.0	0.95	1.15	4.58	8.98	10.7	21.7	12.2	17.1	17.3
2	41	5.27	2.63	3.11	10.4	0.19	0.31	2.56	5.06	6.92	13.6	2.93	7.51	7.94
3	21	4.75	2.36	2.83	10.5	0.13	0.28	1.82	4.70	5.96	12.8	2.32	6.47	6.87
	MST [17] initialization													
1	15	4.24	2.20	2.64	9.24	0.11	0.20	1.59	3.23	7.22	9.22	2.19	6.53	6.47
2	17	3.99	2.02	2.51	8.75	0.16	0.26	2.21	3.14	5.30	8.21	2.20	6.56	6.53
3	11	3.58	1.99	2.39	8.59	0.12	0.21	1.68	2.19	3.73	7.02	2.16	6.52	6.37

Table 2: Middlebury results over increasing iterations.

iterations of our algorithm using this initialization leads to an improvement in overall Middlebury rank from 43 to 11, as shown in the second row of Table 1. It is evident that our plane based labeling process significantly refines and enhances the input fronto-parallel disparity, while being robust to the quality of the input disparity maps. The results indicate that our method adds a refinement step that is robust and can be added to any local or global algorithm generating fronto-parallel disparities. Table 1 also shows results of methods MST [17] and PatchMatch [3] for reference. The disparity maps and errors with respect to ground truth data, for our method with MST initialization, are shown in Fig. 4.

To demonstrate the improvement in accuracy as a function of number of iterations, we tabulate results after each iteration of our algorithm in Table 2, for both WTA and MST [17] based disparity initialization. For both initialization schemes, increasing iterations demonstrates a drop in bad pixels, and an increase in overall rank in general. For WTA initialization the improvement over three iterations is more dramatic, with rank improving from 115 to 21, and the average percent of bad pixels reducing from 9.27 to 4.27. This is also borne out by observing of the top row of Fig. 5, where the drastic improvement in the disparity map from (a) to (d) is clear. For MST initialization, it should be noted that one iteration of our algorithm itself leads to an overall rank of 15. For the second iteration, even though the overall rank is slightly higher, the average percentage of bad pixels are reduced. After 3 iterations, we obtain an overall rank of 11 among the 148 submissions currently present in the Middlebury evaluation. Additionally, we report the lowest average percentage of bad pixels (3.58), of all methods in the evaluation. When tested with an error threshold of 0.75, suitable only for sub-pixel disparities, our methods ranks 1st. Observing the bottom row in Fig. 5, it can be seen that the proposed algorithm is able to recover the ground plane accurately in (d), while it is erroneous in (a). This improvement is possible because of the accurate plane estimation resulting from the feedback loop of the proposed framework. For both initializations, it was empirically observed that running the algorithm for more than 3 iterations resulted in a negligible reduction in the average number of bad pixels, and hence we stop at 3 iterations. The average run-time on the Middlebury set is 25 seconds on a 2.67 GHz Intel Core i7 CPU with 8 GB memory.

It should be noted from Table 1 that the method of [17] performs better on Tsukuba, and is worse on Teddy, Cones and Venus. Tsukuba contains only fronto-parallel disparities, hence, a fronto-parallel labeling is more accurate than a general plane based labeling. However, for scenes with planar surfaces (a common scenario), plane based labeling outperforms most fronto-parallel labeling methods. We also tested our algorithm on two publicly available real-world stereo video data sets : the *cycling* sequence from the RMIT database [2] and the *Ilkay* sequence from the Microsoft i2i database [1]. Fig. 6 shows the snapshots for both sequences, the disparity maps with the proposed method(with MST initialization), and disparity maps with the MST [17] method. For both examples, the proposed method significantly enhances the MST result. For instance, in the cycling scene, the accurate disparity for

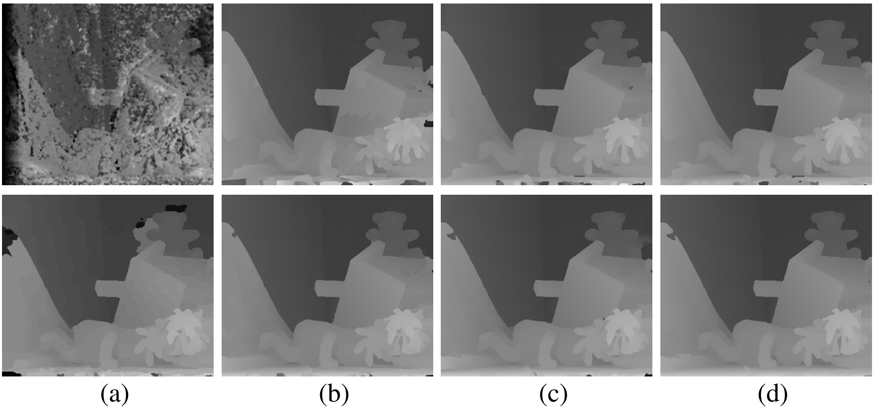


Figure 5: Results of our algorithm with increasing iterations (Teddy): WTA initialization(top), MST [17] initialization(bottom). (a) initial disparity map; (b) 1 iteration; (c) 2 iterations; (d) 3 iterations.

the ground plane is clearly evident in (c).

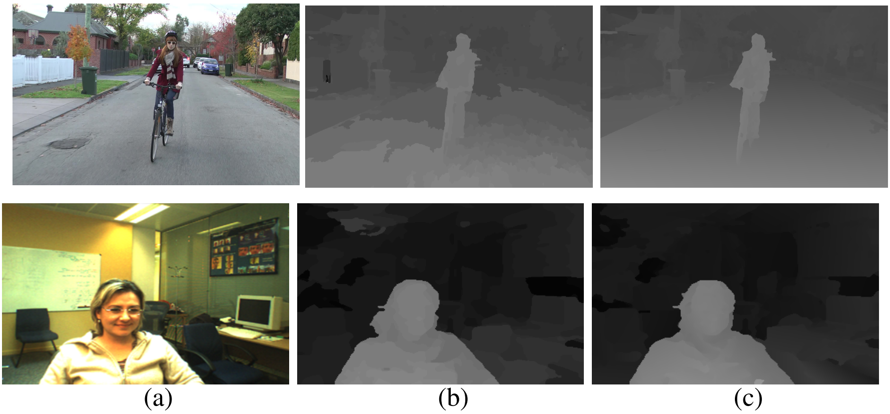


Figure 6: Snapshots of Ilkay [1] and cycling [2] stereo video sequences. (a) Reference frame; (b) disparity map computed by MST [17]; (c) disparity map computed by proposed method.

5 Conclusion

While segmentation based disparity estimation algorithms have advantages, their sensitivity to the initial segmentation is a major drawback. Our method uses plane assignment on a per pixel basis, giving it robustness to segmentation errors and segmentation algorithm parameters. By considering only the dominant plane in each color segment and removing other plane candidate labels, we iteratively improve the accuracy of the estimated disparity maps. The proposed method has the lowest average percentage of bad pixels among all the methods in the Middlebury benchmark, and gives good results even with poor initial disparity maps. This indicates that our method can be used either as a standalone disparity estimation method, or to refine and enhance fronto-parallel disparity maps generated with other methods. Future work will investigate the extension of this framework to higher order modeling of surfaces.

References

[1] Database of monocular and stereo video sequences with labelled foreground and background layers. www.research.microsoft.com/en-us/projects/i2i/data.aspx.

[2] An uncompressed stereoscopic 3d HD video library. www.rmit3dtv.com/download.php.

[3] M. Bleyer, C. Rhemann. C, and C. Rother. Patchmatch stereo - stereo matching with slanted support windows. *BMVC*, pages 1–11, 2011.

[4] M. Bleyer, C. Rother, P. Kohli, D.Scharstein, and S.Sinha. Object stereo - joint stereo matching and object segmentation. *CVPR*, pages 3081–3088, 2011.

[5] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *PAMI*, 24(5):603–619, 2002.

[6] D. Gallup, J. Frahm, P. Mordohai, Q. Yang, and M. Pollefeys. Real-time plane-sweeping stereo with multiple sweeping directions. *CVPR*, pages 1–8, 2007.

[7] S. Gehrig and U. Franke. Improving sub-pixel accuracy for long range stereo. *ICCV*, pages 1–7, 2007.

[8] L. Hong and G. Chen. Segment-based stereo matching using graph cuts. *CVPR*, pages 74–81, 2004.

[9] A. Hosni, M. Bleyer, C. Rhemann, M. Gelautz, and C. Rother. Real-time local stereo matching using guided image filtering. *in Proc IEEE-ICME*, pages 1–6, 2011.

[10] A. Klaus, M. Sormann, and K. Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. *ICPR*, pages 15–18, 2006.

[11] J. Lu, H. Yang, D. Min, and M. N. Do. Patch match filter: Efficient edge-aware filtering meets randomized search for fast correspondence field estimation. *CVPR*, pages 1854–1861, 2013.

[12] X. Mei, X. Sun, W. Dong, H. Wang, and X. Zhang. Segment-tree based cost aggregation for stereo matching. *CVPR*, pages 313–320, 2013.

[13] C. Rhemann, A. Hosni, M. Bleyer, C. Rother, and M. Gelautz. Fast cost-volume filtering for visual correspondence and beyond. *CVPR*, pages 3017–3024, 2011.

[14] D. Scharstein and R. Szeliski. Middlebury stereo evaluation. http://vision.middlebury.edu/stereo/eval/.

[15] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1–3):7–42, 2002.

[16] H. Tao, H.S. Sawhney, and R. Kumar. A global matching framework for stereo computation. *ICCV*, pages 532–539, 2001.

[17] Q. Yang. A non-local cost aggregation method for stereo matching. *CVPR*, pages 1402–1409, 2012.

[18] Q. Yang, R. Yang, J. Davis, and D. Nister. Spatial-depth super resolution for range images. *CVPR*, pages 1–8, 2007.

[19] Q. Yang, L. Wang, R. Yang, H. Stewenius, and D. Nister. Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling. *PAMI*, 31(3):492–504, 2009.

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

[21] Y. Zhang, M. Gong, and Y. Yang. Local stereo matching with 3d adaptive cost aggregation for slanted surface modeling and sub-pixel accuracy. *ICPR*, pages 1–4, 2008.