FAST SEGMENTATION-BASED DENSE STEREO FROM QUASI-DENSE MATCHING

Yichen WEI[1], *Maxime LHUILLIER*[2] *and Long QUAN*[1]

[1]Department of Computer Science, Hong Kong University of Science and Technology
[2]LASMEA-UMR 6602 UBP/CNRS

ABSTRACT

We propose a segmentation-based dense stereo algorithm within an energy minimization framework. The cost function includes a new consistency term to take into account an initial quasi-dense disparity map and handles occlusions explicitly. Based on quasi-dense matching and color segmentation, optimization is performed efficiently by assuming a constant disparity for each region. The assumption is made robust by over-segmentation and a dynamic region splitting method done by graph cut. The efficiency and accuracy of the algorithm are demonstrated on standard stereo data. Experiment results show that the algorithm compares favorably with other state-of-the-art stereo algorithms.

1. INTRODUCTION

Two main problems in dense stereo are lack of texture and occlusion in the image. We propose a segmentation-based algorithm that handles the two problems appropriately within an energy minimization framework. Experiment results show that the algorithm compares favorably with other state-of-the-art stereo algorithms. It has the following features: (1) Color segmentation is used to capture disparity discontinuities and a constant disparity is assumed for each region. The assumption leads to an efficient implementation and is robust due to over-segmentation and a dynamic region splitting step. (2) A quasi dense matching algorithm is integrated as pre-processing step. Based on the quasi dense correspondences, the dense disparity map is initialized efficiently, and a new consistency term is introduced in the cost function to make the matching more reliable in textureless areas. (3) The optimization is performed using a fast greedy method generalized from the method in [16]. The algorithm handles occlusion explicitly while only computes one disparity(depth) map from two or more input images.

Related Work Segmentation-based stereo matching has received a lot of attention recently [16, 6, 2, 12]. It is advantageous in that the disparity smoothness within the region and discontinuities on the region boundary can be efficiently assumed. However, it suffers from the difficulty of appropriate segmentation and causes problems when disparity discontinuities do not coincide with region boundary. This can be solved using iterative segmentation[2, 12] or simply ignored[16, 6]. Our solution is a trade-off. We show that using over-segmentation and an efficient dynamic region splitting method can almost capture all the disparity discontinuities. Our approach is closer to [16, 6] but differs in the initialization step, smoothness handling and optimization method. Our approach is quite different from [2, 12] which perform the segmentation and fit the motion parameters of each region iteratively, therefore, more computationally expensive.

Recently, noticeable progress has been made in stereo by formulating the problem within an energy minimization framework and solving the minimization by graph cut algorithm [13, 4, 8, 9, 3]. Our algorithm differs from above approaches in two aspects: (1) The energy function includes a new consistency term and handles occlusion explicitly while only computes one disparity map; (2) The optimization is done by a simple and efficient greedy algorithm. Graph cut is used in the dynamic region splitting.

Organization The paper is organized as follows. Section 2 formulates and discusses our approach. Section 3 gives the algorithms and implementation details. Experiment results are reported in Section 4. Section 5 concludes this paper.

2. FORMULATION AND APPROACH

Preliminaries Our approach will be formulated using two horizontally rectified images. The generalization to multiple input images is straightforward.

Let I_0 denote the reference(left) image and I_1 be the second(right) image. The algorithm computes a disparity function d over I_0 such that every pixel p in I_0 corresponds to pixel $p+d_p$ in I_1, where d_p is the disparity of p, a horizontal displacement vector. Figure 1(a) and (b) show the reference image and ground truth disparity map on Tsukuba data, respectively.

To handle occlusion, a visibility function $visible(p, d)$ is defined. It returns $true$ if p is visible when warping I_0 to the viewpoint of I_1 according to the disparity function d, i.e., $visible(p, d) = false$, if $\exists q, q+d_q = p+d_p \wedge d_q > d_p$, otherwise $visible(p, d) = true$.

A robust quasi-dense matching algorithm [11] is integrated as pre-processing. It computes the correspondence information for only those sufficiently textured areas. Matching is propagated from the most reliably matched pixels to their neighbors. Propagation is stopped when texture cue is not sufficient. More details can be found in [11]. Let $\mathcal{Q}$ be the set of correspondences computed, $\mathcal{Q} = \{(p, q) \mid p \in I_0, q \in I_1\}$. One may think that $\mathcal{Q}$ partially defines the disparity function d. Figure 1(c) shows a quasi dense disparity map. Note that $\mathcal{Q}$ satisfies the uniqueness constraint, i.e., each pixel can be involved in $\mathcal{Q}$ at most once.

Definition of a new cost function Previous algorithms[13, 4, 8, 9, 2, 12] formulate the dense stereo matching as an energy minimization problem where the energy function typically includes the following two terms: $E_{data}(d)$ and $E_{smoothness}(d)$. The data term, $E_{data}(d)$, measures how consistent the disparity function d agrees with the input images. The smoothness term, $E_{smoothness}$, encodes the smoothness assumption imposed by the algorithm.

Our cost function is defined with the following three terms

$$E(d) = E_{data}(d) + E_{smoothness}(d) + E_{consistency}(d). \tag{1}$$

Our algorithm tries to compute a disparity function d that minimizes (1).

The first two terms play the similar role as mentioned earlier. In addition, a new term $E_{consistency}$ is introduced to measure the agreement of the disparity function d with the pre-computed quasi dense correspondences.

The data term E_{data} differs from the previous ones in that it handles occlusion explicitly,

$$E_{data}(d) = \sum_p D(p, d),$$

where $D(p, d)$ is 0, if $visible(p, d) = false$, and can be any other robust matching cost measure(SSD, SAD, normalized correlation, etc.) otherwise. In the implementation, we use the robust truncated absolute difference, $D(p, d) = min(0, |I_0(p) - I_1(p + d_p)| - K)$, where K is a positive constant.

The smoothness term $E_{smooothness}$ encourages smooth disparities over the 4-connected neighborhood system $\mathcal{N} = \{(p, q) \mid |p_x - q_x| + |p_y - q_y| = 1\}$,

$$E_{smoothness}(d) = \sum_{(p,q) \in \mathcal{N}} S(d_p, d_q),$$

where $S(d_p, d_q)$ is 0, if $d_p = d_q$, and returns a positive penalty $\lambda_{smoothness}$ otherwise.

The consistency term penalizes the disparities that are inconsistent with $\mathcal{Q}$,

$$E_{consistency}(d) = \sum_p C(p, d),$$

where $C(p, d)$ returns a positive penalty $\lambda_{consistency}$ if (i) $visible(p, d) = true$, and (ii)$\exists p', (p', p+d_p) \in \mathcal{Q} \wedge p \neq p'$. Otherwise, $C(p, d)$ is 0.

Segmentation-based representation While the above formulation is completely independent, a segmentation-based representation is favored due to the following considerations:

- In practice, disparity discontinuities usually coincide with intensity edges that can be readily captured by color segmentation [5]. The disparity smoothness within a region is assumed explicitly. Computational complexity is therefore reduced significantly.
- The form of energy function (1) does not allow using the efficient graph cut algorithm[10] in the optimization due to the occlusion handling. Instead, the method in [16] is generalized and simplified. It handles occlusion based on segmentation and computes an approximate solution very fast.

Pixels in a region R are assumed to have the same disparity d_R. Note here the same notation d is used, which will not cause any ambiguity. Our algorithm actually assigns each region a disparity. The assumption enables very efficient computation but is essentially only valid for fronto-parallel surfaces. It causes problems when involving large slanted surfaces. However, we claim that, by using over-segmentation and taking a further dynamic region splitting, the assumption becomes a good approximation in practice. The region splitting is done by graph cut and will be described in Section 3. Figure 1(d) and (e) show examples of color segmentation and region splitting, respectively.

Although the smoothness constraint is imposed inside each region, the smoothness term in (1) is still meaningful since it regularizes the computation for small regions.

Discussions on the consistency term This term is introduced to exploit the fact that only textured pixels can be matched reliably and they should assist or constrain the matching process of other textureless pixels. This idea has also been exploited in [19].

Combined with the segmentation representation, this term makes a textureless region R be matched more reliably. Even when a few pixels in R can be pre-matched correctly and appear in $E_{consistency}$, they tend to fix d_R at its correct value. Experiment results show that the consistency term really helps in textureless and occluded areas.

The main problem is that there are outliers in the pre-computed correspondences, caused by the so called *foreground fattening* problem and typically distributed near the

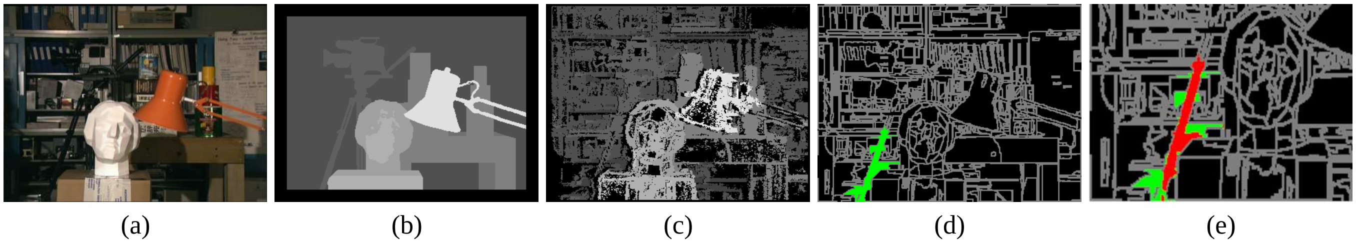

Fig. 1. Tsukuba data. (a)reference image. (b)ground truth disparity map. (c)disparity map partially defined by the quasi dense correspondence. (d)color segmentation. (e)an example of dynamic region splitting. The green colored region in (d) is split into several smaller regions in either red or green in (e). This clearly demonstrates that the splitting step helps to capture disparity discontinuities further where the color segmentation fails.

surface boundary on the textureless background. That is the reason of using condition (i) in function $C(p, d)$. Under this condition, outliers in the occluded area will not take effect since they should be invisible in case of a correct disparity function.

3. ALGORITHM AND IMPLEMENTATION

In the pre-processing, the color segmentation [5] is applied to I_0 and quasi dense matching algorithm [11] is applied to the image pair. Afterwards, the disparity function d is initialized based on the quasi dense correspondences and then the cost function (1) is optimized by a greedy algorithm to obtain the final disparity function. The initialization and optimization steps are elaborated in the following.

3.1. Initialization

In this phase, the problem is to initialize the disparity d_R for each region R. The method is straightforward based on the pre-matched pixels. For each region R, let $disp^R_{1st}$ and $disp^R_{2nd}$ be the two disparities that receives the most and second most votes from the pre-matched pixels in R. Let $pureness(R)$ be the ratio of difference in the number of votes for $disp^R_{1st}$, $disp^R_{2nd}$ and the total number of pre-matched pixels in R. If $pureness(R)$ is smaller than a pre-defined threshold (0.8 in the implementation), the region R will be split dynamically. The pureness testing and splitting are performed iteratively until all regions are initialized. A region R becomes initialized when (i) there is no pre-matched pixels in R, or (ii) R becomes smaller than a pre-defined threshold or (iii) $pureness(R)$ is larger than the pre-defined threshold. In case (i) d_R is assigned the smallest possible disparity. In case (ii), (iii), d_R is assigned $disp^R_{1st}$.

Dynamic Region Splitting For computational consideration, it is assumed that all pixels in R have only two different disparities: $disp^R_{1st}$ and $disp^R_{2nd}$. The splitting is a process of assigning each pixel one of the two disparities. New regions consist of connected pixels with the same assigned disparity. Figure 1(e) shows one example.

The disparity assignment is a bi-labelling problem that can be addressed as an energy minimization problem in MAP-MRF framework and solved exactly via graph cut[7, 4]. The implementation of graph cut in [3] is used, which is efficient for vision applications. Refer to [4] for details.

3.2. Optimization

In this phase, the problem is to minimize the energy function (1). We use a simple greedy algorithm that is similar to α-expansion algorithm in [4, 8, 9].

For every possible disparity α and each region R, d_R is changed to α and energy decrease $\delta E(d, R, \alpha) = E(d^R_\alpha) - E(d)$ is computed, where d^R_α is a disparity function by changing d_R to α. If $\delta E(d, R, \alpha) < 0$, R is recorded. After all the regions have been tested, the disparities of all recorded regions are changed to α. This process is performed over all possible disparities iteratively and stops when the cost function can not be decreased anymore or the maximum number of iterations has been reached.

The method is greedy and local in that it only checks one region at a time and does not consider the interaction of regions simultaneously, therefore it does not compute a global solution. However, in practice, it computes a good local solution, provided that the disparity function d is appropriately initialized.

Implementation The algorithm focuses on the computation of energy decrease $\delta E(d, R, \alpha)$ for each (R, α) pair. $E(d)$ is computed only once for each α. A straightforward way is to first compute $E(d^R_\alpha)$ and then $\delta E(d, R, \alpha)$, but this is too expensive. A more efficient but equivalent method in [16] is generalized and simply described as follows: an image $I_1^{'}$ is created by warping I_0 to the viewpoint of I_1 according to the current disparity function d. This is done only once for each α. Each position in $I_1^{'}$ records the two top-most pixels and relevant information such as the matching cost and region label. Since only one region can have its disparity changed at a time, $I_1^{'}$ records all the information that are necessary to compute the visibility change and energy decrease. $\delta E(d, R, \alpha)$ consists of three terms, δE_{data}, $\delta E_{smoothness}$ and $\delta E_{consistency}$. δE_{data} and $\delta E_{consistency}$ can be efficiently computed based on the information stored in $I_1^{'}$, while $\delta E_{smoothness}$ is computed in a brute-force way. Refer to [16] for more details.

Time complexity of the optimization algorithm is $O(nIDN)$, where n is the number of input images, I is the number of iterations, D is the number of possible disparities and N is the number of pixels in the reference image.

4. EXPERIMENTS

The algorithm is evaluated on the test bed proposed by Scharstein and Szeliski[14, 1]. The evaluation measure is the percentage of wrong disparities differing from the true value more than 1 pixel. This measure is calculated over three different areas in the image, classified as untextured(untex), discontinuous(disc) and the entire image(all). The ground truth disparity map and the stereo data sets used in the experiments are available on the web[1].

Comparison with graph cut algorithms The graph cut algorithms in [4, 9] are independently implemented for comparison on the same platform, referred to as GC and GCMulCam, respectively. The disparity(depth) map of the Tsukuba data using either two or five input images are shown in Figure 2. Each disparity(depth) map is obtained using the best parameters. Corresponding quality measures and running time are given in Table 1. One can verify the correctness of our implementation, by either inspecting the disparity map or comparing the quality metrics with those of the original implementation provided in Table 2.

Our algorithm slightly outperforms GCMulCam with 2 input images and GC with 5 input images, and achieves comparable results in other cases. It performs particularly well on the long thin lamp pole, which benefits from the occlusion handling and color segmentation. It is seen that GC achieves better result using two input images than using five input images, perhaps due to the lack of occlusion handling.

The greedy optimization algorithm is faster than GCMulCam and comparable with GC. If the preprocessing time(about 1 minute) is taken into account, our algorithm is comparable with GCMulCam and slower than GC. Note that GCMulCam computes the depth maps of multiple input images simultaneously and its running time increases at least linearly with the number of input images, since the number of nodes in its graph construction increases linearly.

Results on other data sets The algorithm is also compared with other stereo algorithms. Results are given in Table 2 and Figure 3.

For Tsukuba data, we obtain comparable results with the best algorithms. For the other three data sets involved in Table 2, Venus and Sawtooth data contain large slanted surfaces, and Map data is too textured to make a successful color segmentation. However, quite good results are still obtained that are slightly poorer than the best global algorithms but better than most local algorithms[14, 1]. Note that those occluded and textureless regions in Sawtooth and Venus are matched correctly and no obvious foreground fattening is observed.

#img	algo	all	untex	disc	time(sec)
2	Ours	1.77	0.36	8.66	2.7(0.1)
	GC	1.73	0.86	8.85	7.9
	GCMulCam	2.16	1.27	11.27	18.5
5	Ours	1.36	0.61	7.92	10.3(0.3)
	GC	2.86	2.52	15.80	8.2
	GCMulCam	2.45	3.92	5.79	82.9

Table 1. The quality measures of graph cut algorithms are obtained from our implementation. The left most column is the number of input images. The right most column is the running time of optimization step. For our algorithm, the value in the brackets is the running time of initialization.

The two rightmost columns of Figure 3 show the results on other two data sets, Cones and Teddy, which are only for qualitative evaluation. The disparity discontinuities are successfully identified in most areas and most fine structures are recovered. The black areas in the left of the disparity maps are due to the large disparity range of the two data sets(55 and 52 pixels, respectively).

Parameter Setting In all the experiments, most parameters are fixed, including those in the pre-processing. However, since there are many different components involved in the algorithm, three parameters are selected empirically, namely $\lambda_{smoothness}$, $\lambda_{consistency}$ and λ_{local}. The first two appears in the energy function and the last one is used as the smoothness factor in the region splitting done by a local graph cut. Results shown above are obtained using best parameters.

An undesirable property is that the parameter selection is sensitive to the extent of texture and color segmentation result. This is the main limitation of the algorithm. For example, small $\lambda_{smoothness}$ is favorable for Tsukuba data because of the moderate texture and good segmentation, but large $\lambda_{smoothness}$ is used for Map data, on the contrary. One future work is to choose the parameters automatically according to texture and segmentation information.

5. CONCLUSION

In this paper, a dense stereo algorithm is presented. It integrates several different components into an energy minimization framework. Color segmentation is used to impose smoothness constraint and capture the disparity discontinuity. Quasi dense correspondences are used in initialization as well as in the energy function. The energy

	Tsukuba			Sawtooth			Venus			Map	
algo	all	untex	disc	all	untex	disc	all	untex	disc	all	disc
Our algorithm	1.77	0.36	8.66	1.61	0.38	5.52	2.29	4.08	9.79	0.68	9.00
Layered[12]	1.58	1.06	8.82	0.34	0.00	3.35	1.52	2.96	2.62	0.37	5.24
Belief prop[15]	1.15	0.42	6.31	0.98	0.30	4.83	1.00	0.76	9.13	0.84	5.27
GCMulCam[9]	1.85	1.94	6.99	0.62	0.00	6.86	1.21	1.96	5.71	0.31	4.34
GC+occl[8]	1.27	0.43	6.90	0.36	0.00	3.65	2.79	5.39	2.54	1.79	10.08
GC[4]	1.86	1.00	9.35	0.42	0.14	3.76	1.69	2.30	5.40	2.39	9.35
Multi-cut[2]	8.08	6.53	25.33	0.61	0.46	4.60	0.53	0.31	8.06	0.26	3.27
Var. win.[18]	2.35	1.65	12.17	1.28	0.23	7.09	1.23	1.16	13.35	0.24	2.98
Comp. win.[17]	3.36	3.54	12.91	1.61	0.45	7.87	1.67	2.18	13.24	0.33	3.94
Cooperative[20]	3.49	3.65	14.77	2.03	2.29	13.41	2.57	3.52	26.38	0.22	2.37
Max flow[13]	2.98	2.00	15.10	3.47	3.00	14.19	2.16	2.24	21.73	3.13	15.98

Table 2. Comparison with other algorithms on four data sets. The evaluation is done on the web[1].

function differs from others in that it handles occlusion explicitly and includes a new consistency term. It is optimized approximately by a fast greedy algorithm based on segmentation. An additional region splitting step makes the algorithm more robust. Experiment results show that the proposed algorithm is comparable with best state-of-the-art methods, both in accuracy and efficiency. The main limitation is that several parameters need to be set empirically.

6. ACKNOWLEDGMENTS

This project is supported by the Hong Kong RGC grant HKUST6188/02E.

7. REFERENCES

[1] http://www.middlebury.edu/stereo/.

[2] S. Birchfield and C. Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *ICCV*, 1999, pages 489–495.

[3] Y. Boykov and V. Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in computer vision. In *EMMCVPR*, 2001.

[4] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. In *ICCV*, 1999, pages 377–384.

[5] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. In IEEE *Trans. of PAMI*, Vol.24:603–619, 2002.

[6] F. Ernst, P. Wilinski, and K. V. Overveld. Dense structure-from-motion: An approach based on segment matching. In ECCV, 2002, volume 2, pages 217–231.

[7] D. M. Greig, B. T. Porteous, and A. H. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society, Series B*, 51:271–279, 1989.

[8] V. Kolmogorov and R. Zabih. Computing visual correspondence with occlusions via graph cuts. In *ICCV*, 2001, volume 2, pages 508–515.

[9] V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. In *ECCV*, 2002, volume 3, pages 82–96.

[10] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? In *ECCV*, 2002, volume 3, pages 65–81.

[11] M. Lhuillier and L. Quan Match Propogation for Image-Based Modeling and Rendering. In IEEE *Trans. of PAMI*, volume 24, pages 1140–1146, 2002.

[12] M. H. Lin and C. Tomasi. Surfaces with occlusions from layered stereo. In *CVPR*, 2003.

[13] S. Roy and I.J. Cox. A maximum-flow formulation of the N-camera stereo correspondence problem. In *ICCV*, 1998, pages 492–499.

[14] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In *IJCV*, volume 47, pages 7–42, April 2002.

[15] J. Sun, H. Y. Shum, and N. N. Zheng. Stereo matching using belief propagation. In *ECCV*, 2002, volume 2, pages 510–524.

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

[17] O. Veksler. Stereo matching by compact windows via minimum ratio cycle. In *ICCV*, 2001.

[18] O. Veksler. Fast variable window for stereo correspondence using integral images. In *CVPR*, 2003.

[19] Z. Zhang and Y. Shan. A progressive scheme for stereo matching. *SMILE*, 2000, Lecture Notes.

[20] C. L. Zitnick and T. Kanade. A cooperative algorithm for stereo matching and occlusion detection. In IEEE *Trans. of PAMI*, volume 22, pages 675–684, 2000.

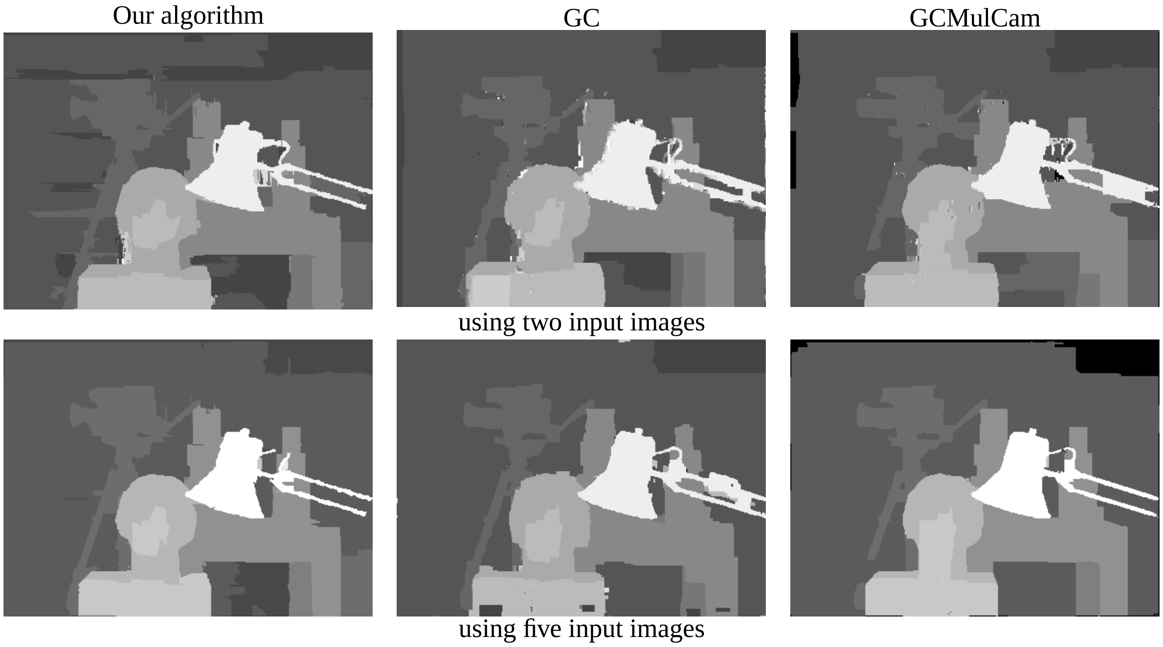

Fig. 2. Comparison with graph-cut algorithms. The three columns from left to right show the disparity maps computed by our algorithm, graph cut(GC)[4], graph cut for multiple cameras(GCMulCam)[9], respectively. The two rows show the result computed from two and five input images(center, left, right, top, bottom), respectively.

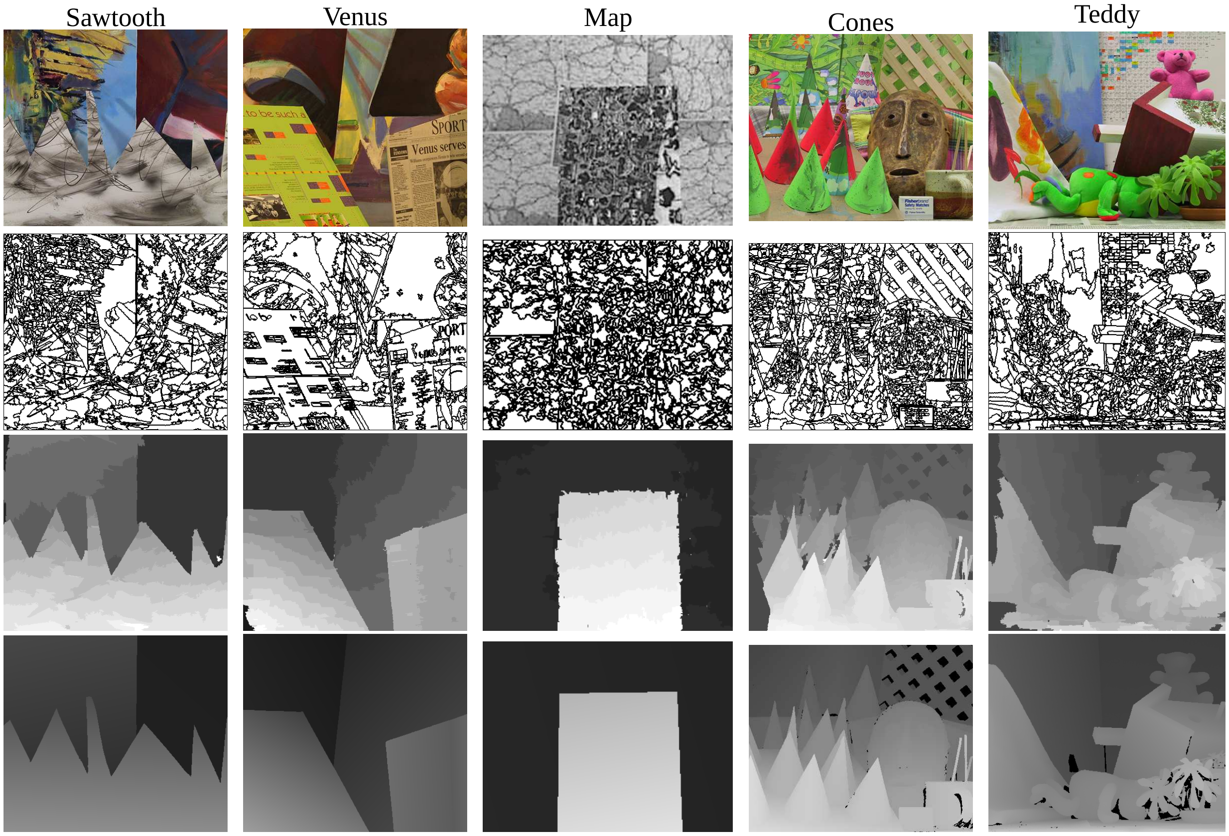

Fig. 3. Our results on other five data sets. From top to bottom, the four rows show the reference image, initial color segmentation, disparity map produced by our algorithm and the ground truth, respectively.