A novel adaptive support window based stereo matching algorithm for 3D reconstruction from 2D images

Jargalsaikhan Iveel and Sumam David, SMIEEE
Department of Electronics and Communication Engineering
National Institute of Technology Karnataka, Surathkal
Mangalore, India
bji.1986@gmail.com

***Abstract*—Three dimensional scene reconstruction, sometimes referred as view synthesis, is a problem in the area of Stereo Vision, which is the most widely used method for gathering depth information from 2D scenes. Stereo vision finds many applications in automated systems such robotics, tracking object in 3D space and constructing a 3D spatial model of a scene. There are many human-machine control applications, such vision based remote control system which exploits stereo vision to control the machine in a touch-free environment.**

In this paper, we present a new stereo vision matching algorithm using an adaptive support window. In many area based algorithms, the selection and computation of the size and shape of the window is the most crucial factor for obtaining high quality disparity map. We developed an adaptive support window based stereo matching using similar color regions selected by the seed growing algorithm. The proposed approach is tested on Middlebury stereo images and results were promising.

I. Introduction

The goal of the stereo vision system is to calculate depth by measuring the disparity between the two dimensional imaged positions of the point in a stereo pair of images taken from disparate locations. Stereo vision can provide accurate, efficient distance measurements over a large range of depths using off-the shelf camera systems. Intuitively stereo is the *simplest three dimensional vision method* to understand, since it is regarded as the most important way in which humans capture depth information.

Stereo vision is highly important in fields such as robotics, to extract information about the relative position of 3D objects in the vicinity of autonomous systems. Other applications for robotics include object recognition, where depth information allows for the system to separate occluding image components, such as one chair in front of another, which the robot may otherwise not be able to distinguish as a separate object by any other criteria.Scientific applications for digital stereo vision include the extraction of information from aerial surveys, for calculation of contour maps or even geometry extraction for 3D building mapping, or calculation of 3D heliographical information such as obtained by the NASA STEREO project. Also stereo vision based system is used for human-machine interface. Most human-machine application is built for controlling the system with help of visual input. One of these systems is hand tracking for human-machine interface.

Local stereo matching algorithms center a support window on a pixel of the reference image. This window is then displaced in the second view along the corresponding epipolar line in order to find the matching point of maximum correlation.

The major challenge in local stereo algorithms is to find appropriate window size and shape for stereo correspondence search. The window should be large enough to capture sufficient intensity variation for the low textured areas, and on the other hand, the same window should be small enough to not include pixels of different disparities in order to avoid the edge blurring effect at disparity discontinuities. In practice, there is no golden middle between these conflicting requirements.

The performance of local stereo matching enhanced with the introduction of the novel segmentation-based support aggregation schemes [1],[2],[3]. These methods can deliver results close to the quality of global approaches. They assume that pixels of homogeneous region share the same disparity value. Main disadvantage of this segmentation based method is computational complexity , hence it needs pre-processing for faster image segmentation.

In the recent years , the selection of the size and shape of the matching window has been the main focus of research in the area-based stereo matching algorithms. Kanade and Okutomi [4] first proposed an adaptive window method which starts with an initial disparity estimation and updates it iteratively for each point by choosing the size and shape of the matching window till it converges. In this method, the matching primitive is intensity and disparity variance. The disadvantages of this method are computational complexity, speed and sensitivity to initial disparity estimate. Veksler [5] made the first attempt to construct non-rectangular matching windows. The method selects window shape by optimizing over a large class of compact windows by using the minimum ratio cycle algorithm. Even though this method performs very well, but it is computationally too complex to implement in real time. Yoon and Kweon [6] proposed locally adaptive

978-1-61284-671-2/11/$26.00 ©2011 IEEE

support weight approach which computes the support weights for each pixel in the support window based on their color, dissimilarity and spatial distance from the center pixel. These weights regulate the pixels influence in the matching process. This approach gives very good results but it is computationally very expensive and is also prone to image noise. Raj Kumar and Siu-Yeung Cho [7] proposed adaptive binary window approach. This method uses support window which is formed via color similarity with different size and shapes in order to select pixels with same disparity. But in practice, the color similarity condition is enough to select similar disparity region

Our approach is based on the work of Yoon and Kweon [6] and improved this support window computation by using not only color similarity property but also considering the spatial information.

The outline of this paper is organized as follows:

Section II described the details of the proposed algorithm. Section III shows the implementation result and results obtained using the proposed approach. Finally, Section IV presents the conclusions and can be drawn from this investigation.

II. ALGORITHM

The proposed algorithm is divided into two stages in order to achieve final depth map of given stereo images. First step is initial disparity map computation whereas second one is rectification process to handle the ambiguous error within disparity map calculation. This algorithm works with pre-rectified stereo images and experimentally Middlebury stereo images [8] are used in the implementation.

First step is the most important processing part which deals with support window construction. The support window construction is the foremost factor for achieving a good result in any region-based methods. Here seed-growing technique was exploited for the selection of pixels with same disparity which worked fast and quite well.

A. Support window construction

As window-based stereo aggregation methods assume that all pixels within the window have same disparity, they fail when windows straddle depth discontinuities. As discussed by Scharstein and Szeliski [9], this results in a foreground fattening effect, as pixels near discontinuities become bimodal and will display a strong preference towards the foreground disparity, even if they are in the background.

Therefore choosing the optimum size and shape of support window is very crucial for obtaining good results. We assume that depths vary smoothly within any image segment with homogeneous color. Based on this assumption, we can disregard or diminish the influence of those pixels within support window which fall outside the homogeneous region that contains the central pixel under consideration.

In the paper, we propose a new support window construction technique which is an extension of simple binary window approach [7]. Yoon and Kweon [6] have shown that support window should be constructed with not only with color similarity but also with strong relationship with its spatial position. Therefore the improvement of our algorithm is that we considered spatial information of each support pixel relation with window's center pixel so that probability to form support region over the same disparity area will increase.

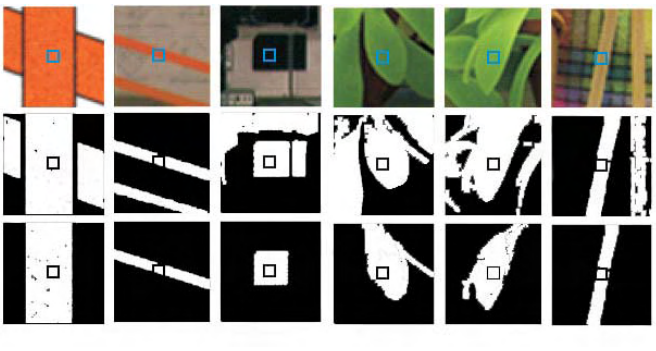


Fig. 1: Support window for the selected patches of the Middlebury stereo images[8]. (First row) Image crops. (Second row) Support window computed by [6]. The support selection method based on only color similarity so it takes wrong support pixels from different disparity regions. (Third row) Our support window selection method. Due to enforcing additional spatial constraint, such wrong selections are avoided (Note. White pixel represents the active matching region)

For example: If there is no path between a pixel p of the support window and the window's center c along which the color varies very little then a pixel p may belong to different disparity although these pixels may satisfy color similarity condition which is adopted in the selection of support pixels [6]. Therefore we put forward two initial conditions for selection of the pixels of the support window: (1) color variation between a pixel p and c should be within threshold value *(visual property)* (2) a path should be exist between these points along any path which the color varies within D_{th} threshold range *(connectivity property)* which further results in accurate selection of the region which belong same disparity range. The support window selection using proposed properties is shown in Figure.1 with comparison to the support window computed by [6].

In order to construct a better support window, we used simple seed growing algorithm which is usually used in image segmentation technique. Most important reason of choosing this algorithm is that it can be implemented very efficiently using FIFO data structure compared to the other techniques. Thus it gives advantage for implementing the proposed algorithm in the real time hardware architecture.

B. Seed growing algorithm

The first step in seed growing algorithm is to select a set of seed points. For our case, initial seed point is a pixel of reference stereo image which is located in the center of the support window. Seed point selection is based on criteria that the color distance between the selected seed point and the candidate point,which should be any adjacent point, is to be

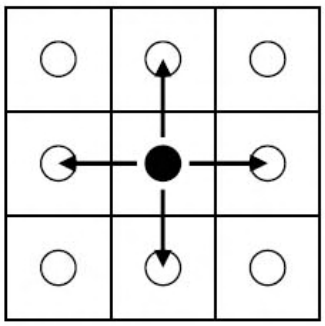


Fig. 2: Seed pixel growing towards 4 adjacent neighbors

within a range of D_{th} which is threshold value. For color distance calculation, we used the Euclidean color distance function $\Delta d(p,c)$, in the RGB color space, which is expressed as below:

$$\Delta d(p,c) = \sqrt{\sum (I_{ch}(p) - I_{ch}(c))^2} \tag{1}$$

where I_{ch} intensity value of respective color channel $ch \in \{r,g,b\}$ also p is a pixel of support window while c is a center pixel.

The seed growing algorithm is implemented as given below

- ***Initialization*** Set the support window size W, Color distance Threshold value D_{th}
- ***Step 1.*** Construction of the *waiting list* with initial seed of *center pixel* c.
- ***Step 2.*** Select seed pixel from *waiting list* of pixels and label it as *current pixel* q. If the list has no new items,terminate the algorithm.
- ***Step 3.*** Calculate the color distance value between *center pixel* c and *candidate pixel* p which is adjacent to the *current pixel* q, using equation (1). If $\Delta d(p,c) \leq D_{th}$, add pixel a to the *waiting list* as new item. Otherwise compare other adjacent pixel as shown in Fig. 2
- ***Step 4.*** When comparison is over, go to ***Step 2.***

This algorithm gives continuity similarity region which is considered as homogeneous region where disparities are assumed to be varied smoothly.Then all pixels within this region will participate in matching process. Further we will refer this region as *active matching region* of the support window as shown in Fig.1.

C. Support window weight

Instead of using common correlation techniques such as *Sum of Squared Difference* (SSD) or *Sum of Absolute Difference* (SAD), here we present a improved support window weight generation technique to increase the accuracy of the correlation over the Adaptive binary window approach (ABW). Fig. 3 shows the support window weight computed by this technique.

In highly textured areas, active matching region becomes very small which increases ambiguity error. Because in the matching process, there will be few pixels within active region will be aggregated. In order to overcome this situation, new window weight technique is proposed which aggregates not only pixels within an active region but also outside the region. We compute the matching cost by searching the number of pixels that their intensity difference less than a threshold value T. The matching cost of a pixel (x,y) can be expressed as:

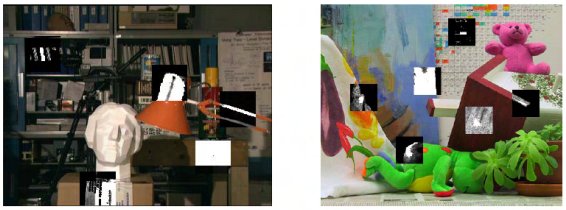


Fig. 3: Support window weight computation on Tsukuba (left) and Teddy (right) images. The weights are computed not only on the active region (which is pure white) but also on outside of this region therefore it can cover many variations of intensity value

$$C(x,y,d) = \sum_{(\eta,\xi)\in\omega} G(I_r(x+\eta, y+\xi), I_t(x+\eta+d, y+\xi) + \\ + \sum_{(\lambda,\mu)\in\overline{\omega}} H(I_r(x+\lambda, y+\mu), I_t(x+\lambda+d, y+\mu) \tag{2}$$

where $(\eta,\xi) \in \left[-\frac{W}{2}, \frac{W}{2}\right]$, W is the size of support window, d is the disparity value and ω represents the pixels in the *active matching region* and $\overline{\omega}$ presents the pixels outside the active matching region, $I_r(x,y)$, $I_t(x,y)$ is a color pixel respectively from $reference$ and $target$ image and G() is and H() the intensity functions which are defined as:

$$G(p,q) = \begin{cases} 1 & |p-q| < T \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

$$H(p,q) = \begin{cases} exp(-\frac{\Delta d(p,c)}{\gamma_c}) & |p-q| < T \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where $exp(-\frac{\Delta d(p,c)}{\gamma_c})$ is the weight function based on the color similarity between pixel p and center pixel c of reference image. Fig. 3 shows computation corresponding weights of certain chosen pixels.

In our approach, noisy pixel which has strong difference value will be controlled by the intensity function $G(p,q)$ and $H(p,q)$ because it doesn't use the intensity difference value directly.

We used simple *Winner-Takes-All* (WTA) strategy for choosing the best disparity value that has highest matching cost. The disparity of a pixel *x,y* can be estimated as:

$$D_c = argmax_d C(x,y,d) \tag{5}$$

Upon finding the appropriate match, we use same disparity value to initialize all the pixels within the corresponding active matching region in the support window as an initial disparity map. We compute the active matching windows and

Fig. 4: Shows the initial disparity map computed by the first stage of algorithm and its corresponding output after the disparity rectification process

perform the matching for all uninitialized pixels to estimate their disparity values. There are some pixels which are part of more than one active matching region. If these regions have different disparity values then the disparity values of these pixels become ambiguous.

D. Disparity rectification

The disparity rectification process is to assign values to the erroneous point from the computation from the initial disparity map calculation step. In this regard, the plane fitting method is the most popular for disparity map refinement used by every top ranking algorithms in Middlebury evaluation table. While this method increases the accuracy of the algorithm, it requires color segmented image as an input. To compute the color segmented image in real-time is a challenging task in itself.

Therefore we used the same algorithm used for initial disparity calculation but difference is that here disparity value assignment is active region based rather pixel based. Fig. 4 shows the rectification process performed after the initial disparity calculation on Tsukubu images.

This disparity rectification process sometimes causes noisy results at the edge boundaries. Therefore we used a median filter (with 3x3 window size) to obtain the final output of the depth map. Fig. 5 shows the filtered result.

Fig. 5: Noise removal by the Median filter with 3x3 support size on the final output of the algorithm shown in Figure 4

III. Implementation

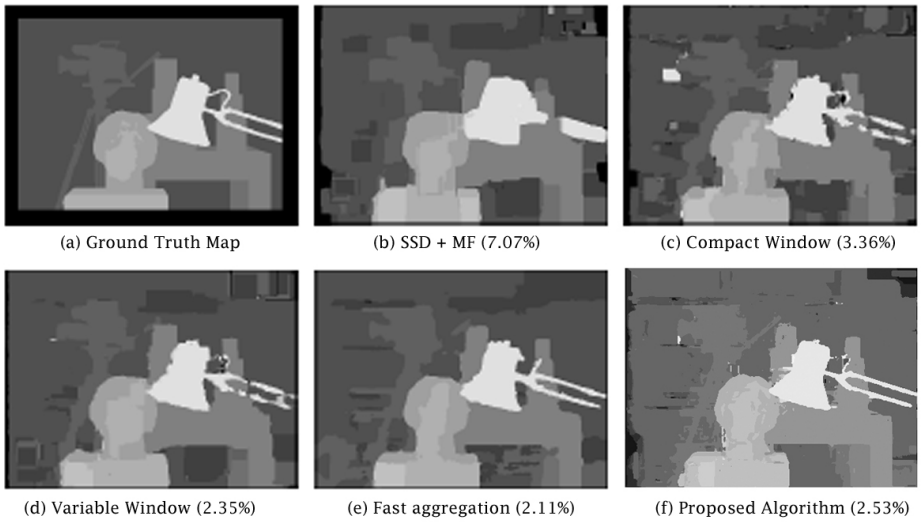
The proposed algorithm has been implemented using OpenCV Library, Microsoft Visual Studio code development environment on a 1.7 GHz Intel Pentium processor system with 1 GB RAM. The performance of the proposed approach is evaluated using Middlebury stereo benchmark [10] which is widely accepted by the stereo vision community. The disparity map computation took around 125 seconds for a 384 x 288 reference image (Tsukuba with disparity range from 0 to 15). We used same parameters for all the Middlebury stereo dataset. In the test run, the algorithms parameters were set to the constant values of matching window size W = 20, color distance threshold value D_{th} = 20.

Figure 8 shows the qualitative results of our approach for all four images. These image pairs along with their ground truth disparity map have been taken from the Middlebury database. The performance of the proposed approach for the Middlebury dataset is summarized in Fig. 7. The data shown in Fig. 7 represent the percentage of the bad pixels with an absolute disparity error lesser than one for different regions: they are non-occluded ($nocc$), whole image (all) and pixels near discontinuities ($disc$). Middlebury rank of the proposed method is 54 out of 109 for the tsukuba stereo pairs which is very high rank for the local stereo method.

Figure 6 compares the performance of the proposed algorithm with other support window based matching algorithms. Fig.6 (a) shows the ground truth image. Fig.6 (b-f) show the results of a fixed window (SSD + min filter)[9], compact window[5], variable window[11], fast aggregation[3], and our adaptive support window based algorithm. The results clearly show that our algorithm works very better at discontinuities compared to other methods.

IV. Conclusion and Future work

In this paper, new adaptive support window based approach is presented. Main purpose of the work was to achieve a high quality and hardware implementation friendly stereo matching



(a) Ground Truth Map (b) SSD + MF (7.07%) (c) Compact Window (3.36%)

(d) Variable Window (2.35%) (e) Fast aggregation (2.11%) (f) Proposed Algorithm (2.53%)

Fig. 6: Results on the Tsukuba image using different support window approach. (a) Ground truth map (b) SSD + MF, (c) Compact Window, (d) Variable Window, (e) Fast aggregation and (f) Proposed Approach.

Error Threshold = 1 (Error Threshold...) — Sort by nonocc — Sort by all — Sort by disc

Algorithm	Avg. Rank	Tsukuba ground truth nonocc	Tsukuba all	Tsukuba disc	Venus ground truth nonocc	Venus all	Venus disc	Teddy ground truth nonocc	Teddy all	Teddy disc	Cones ground truth nonocc	Cones all	Cones disc	Average Percent Bad Pixels
HBpStereoGpu [101]	81.2	3.37 79	5.34 84	13.6 75	1.12 65	2.06 70	14.1 81	12.2 91	19.0 88	27.2 95	6.29 80	14.2 80	16.4 86	11.2
BP+MLH [40]	82.0	4.17 89	6.34 95	14.6 81	1.96 84	3.31 85	16.8 85	10.2 78	18.9 87	24.0 86	4.93 63	15.5 86	12.3 65	11.1
H-Cut [76]	82.4	2.85 69	4.86 80	14.4 79	1.73 81	3.14 83	20.2 91	10.7 81	19.5 89	25.8 91	5.46 75	15.6 87	15.7 83	11.7
SAD-IGMCT [52]	84.9	5.81 102	7.14 100	22.6 104	2.61 87	3.33 86	25.3 99	9.79 74	15.5 72	25.7 90	5.08 66	11.5 58	15.0 81	12.5
FLTG-DDE [90]	86.1	3.03 74	5.28 83	15.0 84	3.39 92	5.02 93	25.0 98	11.0 85	19.5 90	26.3 93	5.78 76	16.0 89	14.2 76	12.5
our method	88.8	2.53 63	3.27 56	9.02 54	4.14 95	5.27 95	14.1 80	17.8 102	25.2 102	30.4 99	25.6 107	32.3 107	32.0 106	16.8
DP [1b]	90.5	4.12 88	5.04 81	12.0 70	10.1 105	11.0 105	21.0 94	14.0 92	21.6 92	20.6 72	10.5 97	19.1 96	21.1 94	14.2
DPVI [67]	90.7	4.76 92	5.83 89	16.6 89	4.89 97	5.66 97	22.9 96	11.0 86	16.2 78	23.4 83	9.64 92	15.6 88	23.5 101	13.3
Bipartite [78]	91.9	2.54 64	4.41 74	13.6 76	6.62 98	7.46 98	18.6 90	16.9 99	24.1 98	30.2 99	15.1 105	21.8 103	23.0 100	15.4
PhaseBased [31]	95.1	4.26 90	6.53 96	15.4 86	6.71 99	8.16 99	26.4 101	14.5 93	23.1 93	25.5 89	10.8 99	20.5 101	21.2 95	15.3
RegionalSup [38]	96.0	3.99 86	6.05 91	14.2 78	8.14 101	9.68 102	36.8 106	18.3 104	26.7 104	32.1 100	9.16 91	19.3 97	19.9 92	17.0
IMCT [62]	96.3	4.54 91	5.90 90	19.8 99	3.16 90	3.83 89	23.2 97	18.0 103	23.1 94	35.3 103	12.7 100	18.5 95	27.9 104	16.3

Fig. 7: The comparison of the proposed algorithm with other algorithms listed on Middlebury Evaluation table for absolute disparity error lesser than 1.0. The complete set of results can be found at http://vision.middlebury.edu/stereo/eval/

algorithm. In order to meet such requirement, a new stereo matching technique was proposed, which uses visual and spatial property to select optimum size and shape of support window.

From the simulation results, it is found that the algorithm performs very well in disparity discontinuity region. But main problem was ambiguous error especially in the region with low texture as seen in each local methods also. In order to overcome it, the matching cost for each pixel in the error regions was re-evaluated and assigned the corresponding value. Actually this error handling process is not so sophisticated because it is also subjected to the ambiguous error. Therefore in the future, the main improvement should be concentrated for handling the problems in sparse regions.

It is worthy to note that the proposed algorithm can be used for the tracking the object in the 3D space due to its accurate shape reconstruction. The object track is used for the human-machine interface which plays important role in vision-based remote control system.

REFERENCES

[1] V. Kolmogorov, A. Criminisi, A. Blake, G. Cross, and C. Rother, "Bi-layer segmentation of binocular stereo video," *CVPR*, vol. 2, pp. 407–414, 2001.

[2] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, 2002.

[3] F. Tombari, S. Mattoccia, and L. D. Stefano, "Segmentation-based adaptive support for accurate stereo correspondence," *IEEE Pacific-Rim Symposium on Image and Video Technology*, pp. 427–438, 2007.

[4] T. Kanade and M. Okutomi, "A stereo matching algorithm with adaptive window: Theory and experiment," *IEEE transactions on Pattern analysis and Machine Intelligence*, vol. 16, no. 9, pp. 920–932, 1994.

[5] O.Veksler, "Stereo matching by compact window via minimum ratio cycle," *International Journal of Computer Vision*, vol. 1, pp. 556–561, 2002.

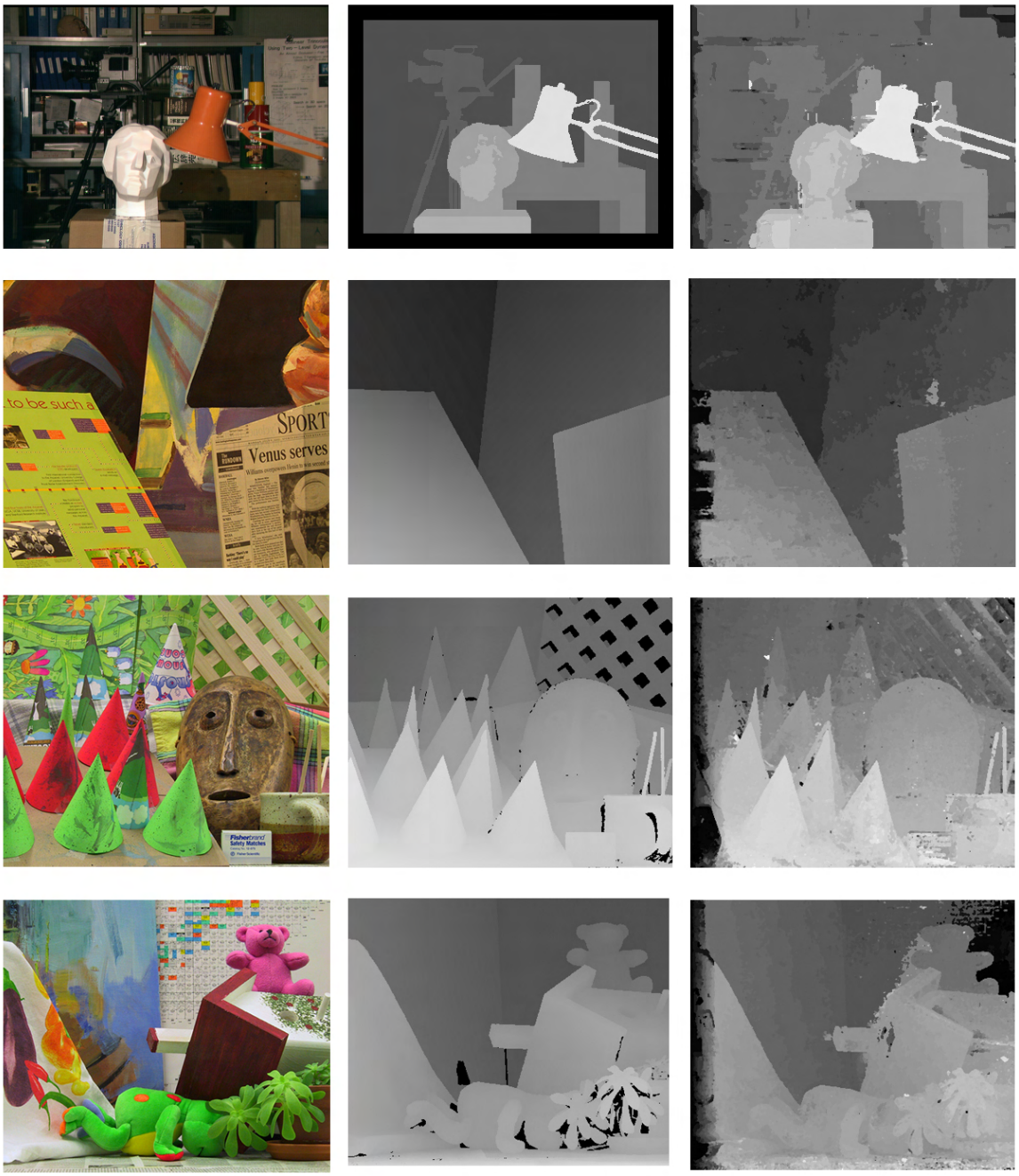


Fig. 8: The proposed algorithm evaluation on the Middlebury dataset (Tsukuba, Venus, Conus and Teddy) The first column shows the input images, the second shows the ground truth disparity map and the third column shows the proposed approach results

[6] K.J.Yoon and I.S.Kweon, "Adaptive support weight approach for stereo correspondence search," *IEEE Transactions on Pattern analysis and Machine Intelligence*, vol. 28, no. 4, pp. 650 – 656, 2005.
[7] R. K. Gupta and S.-Y. Cho, "Real-time stereo matching using adaptive binary window," *ISVC Proceedings of the 6th international conference on Advances in visual computing*, vol. 2, pp. 129–138, 2010.
[8] http://vision.middlebury.edu/stereo/data/.
[9] D.Scharstein and R.Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithm," *International Journal of Computer Vision*, vol. 47, no. 1/2/3, pp. 7–42, 2002.
[10] http://vision.middlebury.edu/stereo/eval/.
[11] O.Veksler, "Fast variable window for stereo correspondence using integral images," *IEEE Conf.Computer Vision and Pattern Recognition*, vol. 1, pp. 556–561, 2003.