A Correlation-Based Approach for Real-Time Stereo Matching

Conference Paper · November 2010

DOI: 10.1007/978-3-642-17274-8_13 · Source: DBLP

2 authors:

Raj Kumar Gupta
Nanyang Technological University

David Cho
University of Nottingham Ningbo China

A Correlation-based Approach for Real-Time Stereo Matching

Raj Kumar Gupta and Siu-Yeung Cho

Forensics and Security Laboratory, School of Computer Engineering
Nanyang Technological University, Singapore
{rajk0005, assycho}@ntu.edu.sg

Abstract. In this paper, we present a new area-based stereo matching algorithm that computes dense disparity maps for a real time vision system. While many stereo matching algorithms have been proposed in recent years, correlation-based algorithms still have an edge due to speed and less memory requirements. The selection of appropriate shape and size of the matching window is a difficult problem for correlation-based algorithms. We use two correlation windows (one large and one small size) to improve the performance of the algorithm while maintaining its real-time suitability. Unlike other area-based stereo matching algorithms, our method works very well at disparity boundaries as well as in low textured image areas and computes a sharp disparity map. Evaluation on the benchmark Middlebury stereo dataset has been done to demonstrate the qualitative and quantitative performance of our algorithm.

1 Introduction

The estimation of depth from a pair of stereo images is one of the most challenging problems in the field of computer vision and is vital for many areas like robotics and virtual reality. An accurate disparity map can help robots navigate in the real environment while in virtual reality; disparity maps play an important role in 3D reconstruction from the image sets. A fast and accurate matching at object boundaries is necessary for proper rendering and reconstruction of an object in virtual environment. The 3D reconstruction problem can be viewed as stereo correspondence problem which includes finding a set of points in one image that can be identified as the same points in another image.

A large number of algorithms have been proposed to compute dense disparity map. A detailed overview on these stereo matching algorithms can be found in [1]. In general, the stereo matching algorithms can be divided into two categories: local methods and global methods. Local algorithms are statistical methods and are usually based on correlation. Global algorithms make explicit smoothness assumption and then solve it through various optimization techniques. Global algorithms are computationally very expensive which makes them impractical for real-time systems.

Local stereo matching algorithms can be subdivided into two broad categories: area-based algorithms and feature-based algorithms. Area-based algorithms use neighboring pixels of a given pixel to find the suitable match in another image by using the intensity values of the pixels. These algorithms mostly use normalized cross-correlation (NCC) [2] or sum of absolute differences (SAD) [3],[4] or sum of squared differences (SSD) [5],[6],[7],[8] technique during the window matching process. The performance of area-based algorithms is highly influenced by the shape and size of the image region used during the matching process. Feature-based algorithms rely on feature extraction and match local cues (e.g. edge, corners). Through, these algorithms work very fast but they generate sparse disparity maps.

In this paper, a real-time correlation-based stereo matching method is presented that computes accurate disparity map from a stereo pair of images. The proposed algorithm uses two correlation windows (one large and one small size) to compute the disparity map. While large correlation window gives good results at non-textured image regions, the small window improves the performance at depth discontinuities. To demonstrate the performance of the proposed method, we have evaluated our algorithm using Middlebury datasets [9]. The rest of the paper organized as follows: In Section 2, we briefly cover the related literature; Section 3 describes the proposed algorithm. Section 4 contains the experimental results on the Middlebury dataset and a detailed comparison with other real-time stereo matching algorithm. The last section presents our conclusion and discusses the future work.

2 Related Work

The correlation-based methods find the corresponding match by measuring the similarity between two image areas. Most common correlation-based methods use Cross Correlation or the Sum of Squared or Absolute differences. The performance of these methods is strongly influenced by the size and shape of the matching image area. Usually, rectangular matching windows [3],[6],[7],[10],[11] are used to achieve high computational performance. The size of the matching window determines the number of pixels to be used for correlation. For the reliable disparity estimation, the matching window must be large enough to cover enough intensity variations, but small enough to cover only the pixels having the same disparity value. This requirement raises the need of different shape and size windows at different pixels within the same image as no fixed window size works well. While the large size window blurs the object boundaries, the small size window results are unreliable in low textured image regions. The pixels near to disparity discontinuity require windows of different shapes to avoid crossing the disparity.

Kanade and Okutomi [8] proposed an adaptive window-based method which starts with an initial estimation of the disparity map and updates it iteratively for each point by choosing the size and shape of a window till it converges. It uses the intensity and disparity variance to choose the window with the least uncertainty. This method is sensitive to the initial disparity estimations.

Boykov et al. [12] proposed a variable window algorithm, which is considerably fast and suitable for a real-time implementation. However, this method suffers from different systematic errors. Fusiello et al. [5] proposed a simple multiple window approach. For each pixel and disparity, the correlation with a limited number of windows is performed and it retains the disparity with the best correlation value. Since it uses a limited number of windows, it cannot cover the complete windows range of required shapes and sizes.

Hirschmuller et al. [10] proposed a multiple window-based approach that uses different size windows, mainly focused on reducing the errors at depth discontinuities. The algorithm uses a border correction filter to improve matches at object borders. Many other multiple window approaches [7],[11],[13] have been proposed which use multiple windows of different sizes or use windows of different orientations to compute the matching cost.

Veksler [14] proposed an algorithm which chooses the appropriate window shape by optimizing over a large class of compact window by using the minimum ratio cycle algorithm. The window cost is computed as the average window error and bias to larger windows. While this method works very well, it is not efficient enough for real-time system. In [15], Veksler introduced another approach by computing the correlation value of several different window sizes for the pixel of interest and selects the window size with least matching error. However, this algorithm needs many user defined parameters for matching cost computation.

Yoon and Kweon [16] proposed a locally adaptive support weight approach which computes the support weights for each pixel in the support window based on their color dissimilarity and the spatial distance from the center pixel. These weights regulate the pixel's influence in the matching process. This approach gives very good results but it is computationally very expensive and also prone to image noise. The reported computation time of this algorithm on a fast machine is around one minute which makes it unsuitable for real-time systems.

3 Algorithm

In this section, we briefly describe the proposed algorithm. The proposed algorithm consists of the following four processing modules: (1) Initial matching; (2) Unreliable pixel detection; (3) Disparity interpolation; (4) Disparity refinement. In initial matching step, we compute the initial disparity map by using two different sizes of correlation windows. In unreliable pixel detection step, we use left-right consistency check to identify unreliable pixels. The left-right consistency check enforces the uniqueness constraint and identifies those pixels which have unreliable disparity. In disparity interpolation step, we estimate the disparity for unreliable pixels identified by left-right consistency check [6]. In final step, we refine the disparity map by using the reference image to improve the accuracy at depth discontinuities. We assume rectified image pair as an input i.e. the epipolar lines are aligned with the corresponding scanlines.

3.1 Initial Matching

To compute the initial disparity map, we use the sum of absolute difference (SAD) based matching approach by using the large correlation window. The matching cost $C(x, y, d)$ of pixel (x, y) for disparity value d is given as follows:

$$C(x, y, d) = \sum_{i=-\omega/2}^{i=\omega/2} \sum_{j=-\omega/2}^{j=\omega/2} |I_l(x+i, y+j) - I_r(x+i, y+j-d)| + \xi, \qquad (1)$$

where $I_l(x, y)$ and $I_r(x, y)$ are the intensities of the pixel in left and right image, respectively. ω represents the size of the matching window. The matching cost is computed for all possible disparity values and the disparity value with minimum matching cost is selected as shown in Fig. 1(a). However, in non-textured image regions, the matching window can have many such minima as shown in Fig. 1(b). It becomes very hard to determine the correct disparity values of the pixels that resides in such image regions. We use the disparity values of neighboring pixels to determine the disparity of such pixels. We propose a penalty term ξ based on gradient and the disparity values of neighboring pixels to estimate the disparity value in non-textured image regions accurately. The penalty term ξ is given as:

$$\xi = T(|d - d'|)\left(1 - \frac{|I_l(x, y) - I_l(x, y')|}{255}\right), \qquad (2)$$

where d' is the disparity of the neighboring pixel (x, y') and T is the constant. The value of ξ changes according the change in gradient and becomes higher in low gradient image regions (e.g. low or non-textured areas) and lower in high gradient image regions (e.g. depth discontinuities). Fig. 1(c) demonstrates the matching costs computed for different disparity values in non-textured region of Tsukuba image at point $(205, 230)$ without using ξ. We can see that there is ambiguity due to the number of local minima. This problem can be solved by using gradient and disparity values of the neighboring pixels. Fig. 1(d) demonstrates the matching cost computed by using equation (1). We can see that the disparity value can be easily determined now by using *winner-takes-all* approach. The matching costs C_l and C_r are computed for both left and right neighbors of a pixel respectively. The disparity of a pixel (x, y) can be computed as:

$$d_c(x, y) = \min\left(arg \min_d C_l(x, y, d), arg \min_d C_r(x, y, d)\right). \qquad (3)$$

After computing the disparity values using the large correlation window, we use the small correlation window near the depth discontinuities. The use of large correlation window blurs the objects boundaries and the actual positions of these boundaries are usually within the distance of half the size of correlation window [10]. We compute the disparity values of all such pixels that reside near the depth discontinuities within the range of half the size of large correlation window. While computing the disparity of such pixels, we restrict the evaluation

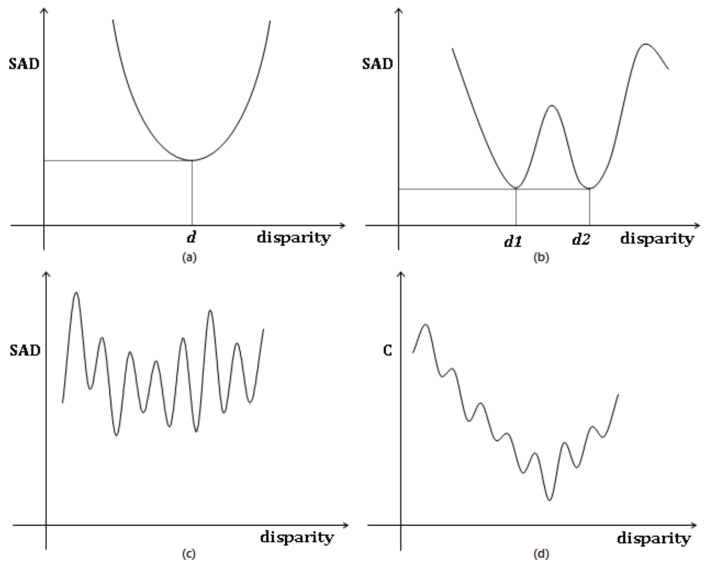

Fig. 1. Shows the problem in disparity selection. In (a), the disparity can be easily determined as d due to unique minimum value. It becomes ambiguous in case of multiple local minima as shown in (b). (c) shows the matching cost calculated at point $(205, 230)$ of Tsukuba image. (d) displays the matching cost calculated by using the penalty term ξ for the same image point. The disparity can be uniquely determined in (d)

of the small correlation window only within those disparity values that are carried by neighboring pixels. The disparity of such pixels can be computed as:

$$d_c(x, y) = arg \min_{d \in N} C(x, y, d), \tag{4}$$

where $d_c(x, y)$ is the disparity of the pixel (x, y) and N represents the disparity values of the neighboring pixels. The cost $C(.)$ is computed without using the penalty term as mentioned in Equation 1.

3.2 Unreliable Pixel Detection

The left-right consistency check is a very effective way to detect the unreliable pixels. The left-right consistency check is based on uniqueness constraint that assumes the one-to-one mapping between the stereo image points. We compute left and right initial disparity maps by using the method described in Section 3.1 by choosing the left and right image as reference image respectively. For each pixel of the left disparity map, we check whether it carries the same disparity value as its matching point in the right disparity map. A valid correspondence should match in both directions. A simple test of left-right cross checking can be written as:

$$|d_l(x, y) - d_r(x, y")| < 1, \tag{5}$$

where (x, y) and $(x, y")$ are the correspondence pair of pixels and $d_l(x, y)$ and $d_r(x, y")$ are left and right disparities for the points (x, y) and $(x, y")$, respectively. All the pixels that fail to satisfy the Equation 5 are marked as unreliable pixels.

3.3 Disparity Interpolation

The left-right consistency check filters out occluded pixels as well as the unreliable pixels. In this step, we assign new disparity values to all such pixels with the help of their reliable neighboring pixels. For each unreliable pixel, we search for valid pixels in its 8 neighboring pixels. For all these valid pixels, we compute the distance between the intensities of unreliable pixel and its valid neighbor in the reference image. We assign the disparity value of the valid pixel which has the minimum distance from unreliable pixel in reference image.

3.4 Disparity Refinement

To remove the outliers for each color segment, plane fitting method is used widely. It is assumed that the neighboring pixels which have the same intensity values will also have the same disparity values. While this method increases the accuracy of the algorithm, it requires the color segmented image as an input which makes it computationally very expensive. Here, we use the disparity refinement approach that has been proposed in [17]. The algorithm uses the reference color information to refine the disparity map without performing any image segmentation.

4 Experiments

The Middlebury stereo benchmark dataset [9] has been used to evaluate the performance of the proposed algorithm. In our experiment, the small window size is chosen as 3 x 3 for all test images while the large window size is set to 9 x 9 for all images. Figure 3 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 [9]. The performance of the proposed method for the Middlebury dataset is summarized in Table 1. The values shown in Table 1 represent the percentage of the bad pixels with an absolute disparity error greater than one for different regions: they are non-occluded (*nocc*), whole image (*all*) and pixels near discontinuities (*disc*). The last column of the table shows the overall performance of the algorithm for all four images.

Fig. 2 demonstrates the performance of the proposed algorithm with different window sizes. While small window size is fixed to 3 x 3, we change the size of the first window (described as large window in proposed algorithm) used during initial matching operation. The window size has been changes from 5 x 5 to 21 x 21. It shows the percentage error in non-occluded, whole image and near depth discontinuities for Tsukuba, Venus, Teddy and Cones images on using different

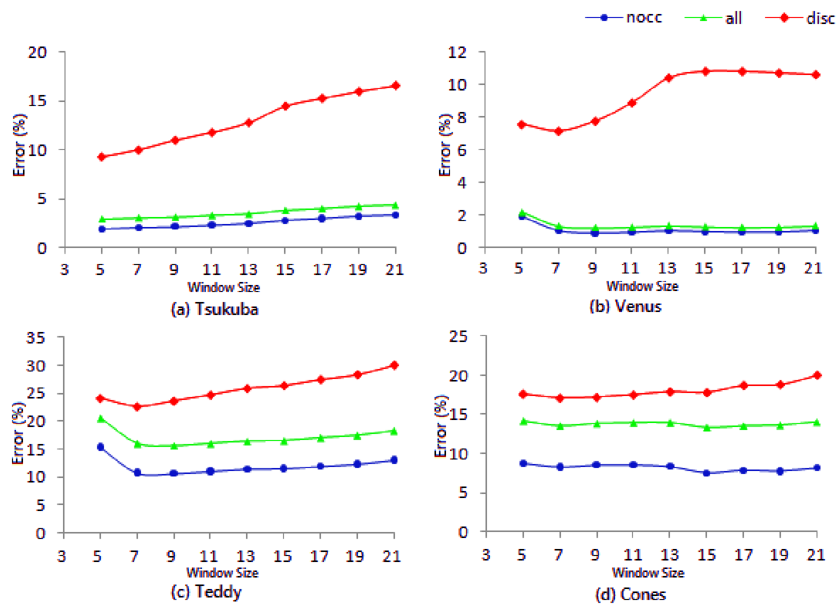

Fig. 2. Shows the percentage error in non-occluded (*nocc*), whole image (*all*) and near depth discontinuities (*disc*) for different window sizes for all four images

window size. The error graphs show that the change in the size of matching window does not affect the performance of the algorithm significantly.

Fig. 4 shows the qualitative results for new Middlebury dataset images. These test images are taken from both Middlebury 2005 and 2006 datasets. These datasets consist of variety of image features i.e. complex geometry (dolls and Moebius), repetitive patterns (Aloe and Cloth1) and non-textured image regions (Books and Cloth2). The images in these datasets also have large disparity ranges, resulting in large occlusions. Due to large occlusions and higher percentage of untextured surfaces, the new Middlebury datasets are much more challenging as compared to standard stereo benchmark dataset which contains images such as Teddy and Cones. The experimental results clearly show that the proposed algorithm works very well in case of repetitive patters, object boundaries, as well as in occluded and non-textured image regions. These experimental results are obtained by taking same parameters (window size and constant value T) for all the images.

5 Conclusions and Future Work

In this paper, we present a new correlation-based stereo matching approach. The algorithm uses two correlation windows (one large and one small size) to compute the disparity map. While large correlation window gives good results

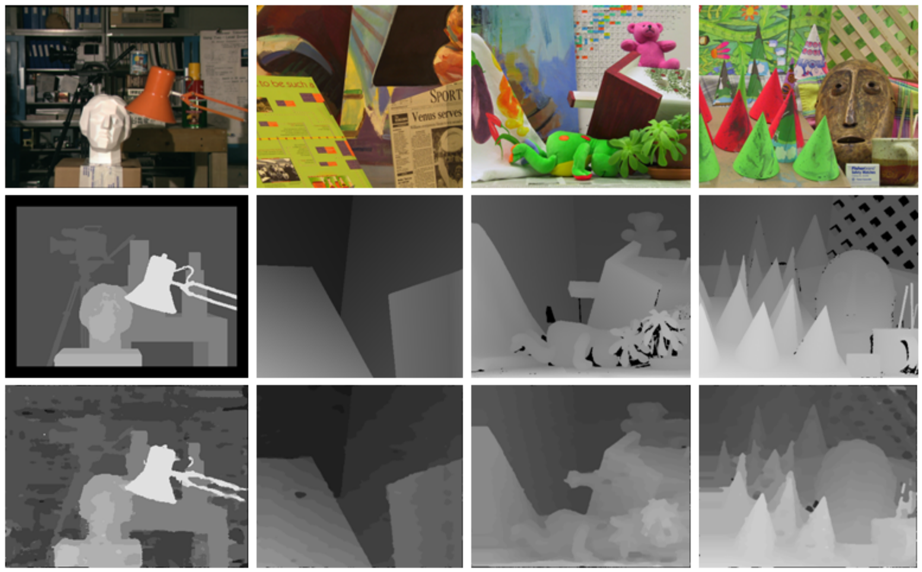

Fig. 3. Results on the Middlebury data set (Tsukuba, Venus, Cones and Teddy). The first row shows the left images, the second row shows the corresponding ground truth disparity maps and the third row shows the results obtained by using our algorithm

at non-textured image regions, the small window improves the performance at depth discontinuities. The algorithm use simple mathematical operations and can be easily implemented on GPU. Although, the proposed method works very fast, the parallel implementation of the algorithm can reduce the computation time significantly. The computation time can also be reduced by using the sliding window approach at the time of correlation. In our future work, we plan to extend this work with all these investigation to improve the efficiency of the algorithm.

References

1. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. Int. Journal on Computer Vision **47** (2002) 7–42
2. Faugeras, O., Hotz, B., Mathieu, M., Viville, T., Zhang, Z., Fua, P., Thron, E., Moll, L., Berry, G.: Real-time correlation-based stereo: algorithm, implementation and applications. INRIA Technical Report No. 2013 (1993)
3. Yoon, S., Park, S.K., Kang, S., Kwak, Y.: Fast correlation-based stereo matching with the reduction of systematic errors. Pattern Recognition Letters **26** (2005) 2221–2231
4. Stefano, L., Marchionni, M., Mattoccia, S.: A fast area-based stereo matching algorithm. Image and Vision Computing **22** (2004) 983–1005
5. Fusiello, A., Roberto, V., Trucco, E.: Efficient stereo with multiple windowing. IEEE Conf. Computer Vision and Pattern Recognition (1997) 858–863

Table 1. Comparison of the proposed method with other real-time algorithms listed on Middlebury evaluation table for absolute disparity error > 1. The complete set of results can be found at http://vision.middlebury.edu/stereo/eval/

Algorithm	**Tsukuba**			**Venus**			**Teddy**			**Cones**			**%**
	nocc	*all*	*disc*	*nocc*	*all*	*disc*	*nocc*	*all*	*disc*	*nocc*	*all*	*disc*	**Error**
RTBFV[18]	1.71	2.22	6.74	0.55	0.87	2.88	9.90	15.0	19.5	6.66	12.3	13.4	7.65
RTABW[17]	1.26	1.67	6.83	0.33	0.65	3.56	10.7	18.3	23.3	4.81	12.6	10.7	7.90
Our Results	*2.25*	*3.08*	*11.6*	*0.92*	*1.31*	*7.53*	*10.7*	*15.7*	*23.6*	*8.25*	*13.5*	*16.6*	*9.59*
RTCensus[19]	5.08	6.25	19.2	1.58	2.42	14.2	7.96	13.8	20.3	4.10	9.54	12.2	9.73
RTGPU[20]	2.05	4.22	10.6	1.92	2.98	20.3	7.23	14.4	17.6	6.41	13.7	16.5	9.82
DCBGrid[21]	5.90	7.26	21.0	1.35	1.91	11.2	10.5	17.2	22.2	5.34	11.9	14.9	12.5
SSD+MF[1]	5.23	7.07	24.1	3.74	5.16	11.9	16.5	24.8	32.9	10.6	19.8	26.3	15.7

6. Fua, P.: A parallel stereo algorithm that produces dense depth maps and preserves image features. Machine Vision and Applications **6** (1993) 35–49
7. Adhyapak, S., Kehtarnavaz, N., Nadin, M.: Stereo matching via selective multiple windows. Journal of Electronic Imaging **16** (2007)
8. Kanade, T., Okutomi, M.: A stereo matching algorithm with an adaptive window: Theory and experiments. IEEE Trans. Pattern Analysis and Machine Intelligence **16** (1994) 920–932
9. (http://vision.middlebury.edu/stereo)
10. Hirschmuller, H., Innocent, P., Garibaldi, J.: Real-time correlation-based stereo vision with reduced border errors. Int. J. on Computer Vision **47** (2002) 229–246
11. Okutomi, M., Katayama, Y., Oka, S.: A simple stereo algorithm to recover precise object boundaries and smooth surfaces. Int. Journal on Computer Vision **47** (2002)
12. Boykov, Y., Veksler, O., Zabih, R.: A variable window approach to early vision. IEEE Trans. Pattern Analysis and Machine Intelligence **20** (1998) 1283–1294
13. Jeon, J., Kim, C., Ho, Y.S.: Sharp and dense disparity maps using multiple windows. Lecture Notes Computer Science (2002) 1057–1064
14. Veksler, O.: Stereo matching by compact window via minimum ratio cycle. IEEE Int. Conf. Computer Vision **1** (2001) 540–547
15. Veksler, O.: Fast variable window for stereo correspondence using integral images. IEEE Conf. Computer Vision and Pattern Recognition **1** (2003) 556–561
16. Yoon, K.J., Kweon, I.S.: Adaptive support-weight approach for correspondence search. IEEE Trans. Pattern Analysis and Machine Intelligence **28** (2006) 650–656
17. Gupta, R., Cho, S.Y.: Real-time stereo matching using adaptive binary window. 3D Data Processing, Visualization and Transmission (2010)
18. Zhang, K., Lu, J., Lafruit, G., Lauwereins, R., Gool, L.V.: Real-time accurate stereo with bitwise fast voting on cuda. ICCVW (2009)
19. Humenberger, M., Zinner, C., Weber, M., Kubinger, W., Vincze, M.: A fast stereo matching algorithm suitable for embedded real-time systems. CVIU (2010)
20. Gong, M., Yang, Y.: Near real-time reliable stereo matching using programmable graphics hardware. IEEE Conf. Computer Vision and Pattern Recognition (2005)
21. Richardt, C., Orr, D., Davies, I., Criminisi, A., Dodgson, N.: Real-time spatiotemporal stereo matching using the dual-cross-bilateral grid. European Conference on Computer Vision (2010)

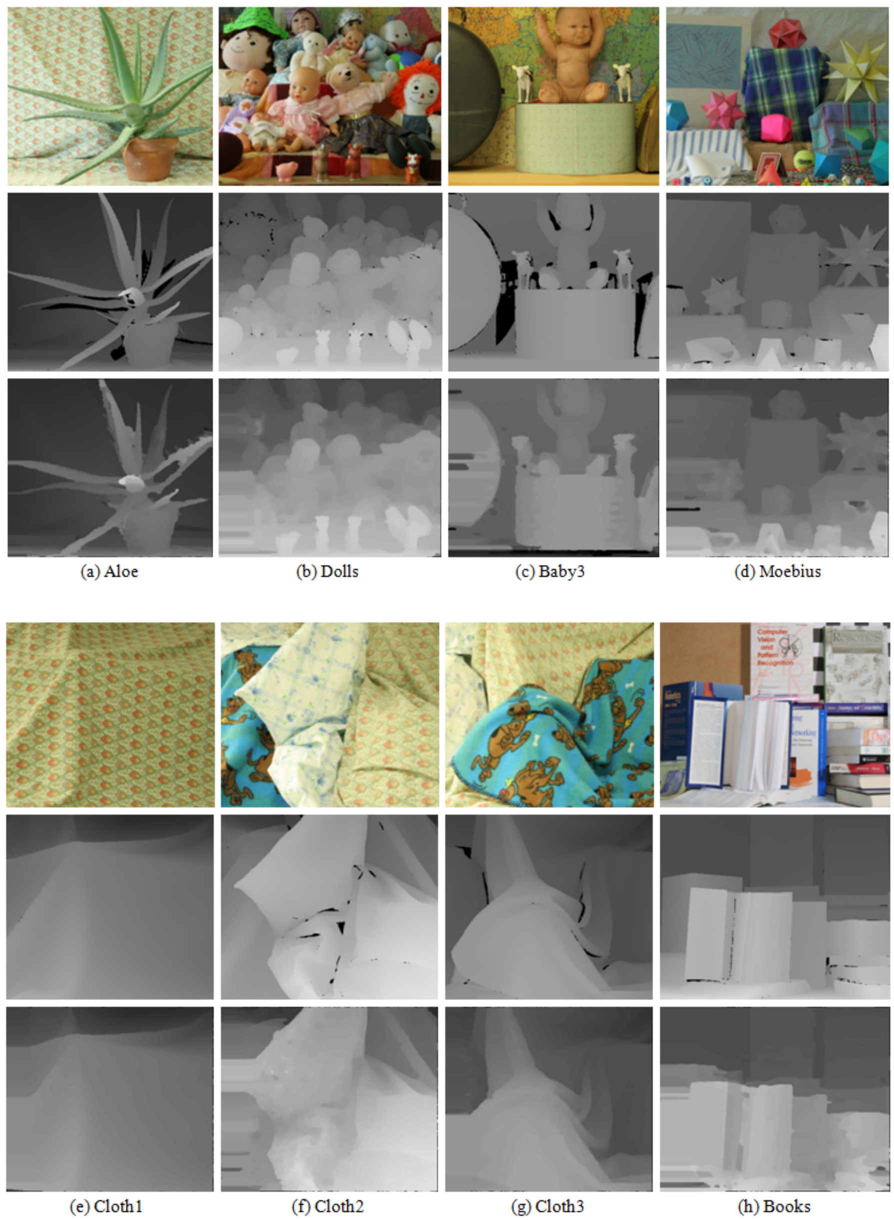

Fig. 4. Results on the new Middlebury dataset. The first row shows the left images, the second row shows the corresponding ground truth disparity maps and the third row shows the results obtained by using the proposed algorithm