Published in IET Computer Vision
Received on 17th April 2011
Revised on 17th July 2012
Accepted on 7th January 2013
doi: 10.1049/iet-cvi.2011.0077

ISSN 1751-9632

Window-based approach for fast stereo correspondence

Raj Kumar Gupta, Siu-Yeung Cho

Forensics and Security Laboratory, School of Computer Engineering, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639 798, Singapore
E-mail: davidcho@pmail.ntu.edu.sg

Abstract: In this study, the authors present a new area-based stereo matching algorithm that computes dense disparity maps for a real-time vision system. Although many stereo matching algorithms have been proposed in recent years, correlation-based algorithms still have an edge because of speed and less memory requirements. The selection of appropriate shape and size of the matching window is a difficult problem for correlation-based algorithms. In the proposed approach, two correlation windows are used to improve the performance of the algorithm while maintaining its real-time suitability. The CPU implementation of the proposed algorithm computes more than 10 frame/s. Unlike other area-based stereo matching algorithms, this method works very well at disparity boundaries as well as in low textured image areas and computes a dense and sharp disparity map. Evaluations on the benchmark Middlebury stereo datasets have been performed to demonstrate the qualitative and quantitative performance of the proposed 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 such as robotics and virtual reality. An accurate disparity map can help robots navigate in the real environment, whereas in virtual reality, disparity maps play an important role in three-dimensional (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 from 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. Local stereo-matching algorithms can be subdivided into two broad categories: area-based algorithms and feature-based algorithms. Area-based algorithms use neighbouring pixels of a given pixel to find the suitable match with another image by using the intensity values of the pixels. These algorithms mostly use normalised cross-correlation (NCC) [2] or sum of absolute differences (SAD) [3, 4] or sum of squared differences (SSD) [5–8] 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.

Global algorithms [9–12] make explicit smoothness assumption and then solve it through various optimisation techniques. Most of the global algorithms are based on energy minimisation [9, 11, 12]. Global algorithms give very good results but these methods are computationally expensive which makes them impractical for real-time systems. In the past few years, many efforts have been made to develop real-time algorithms for disparity estimation. A classical category of these algorithms are based on dynamic programming [10]. Dynamic programming-based algorithms perform scanline-based optimisation for all scanlines in the image independently. Although dynamic programming produces better results on occlusion boundaries, the inconsistency between scanlines leads to the well-known streaking artifacts. This problem has been solved recently by multiple-label dynamic programming [13, 14]. However, it still lacks the level of accuracy. Felzenszwalb and Huttenlocher [9] proposed an efficient belief propagation algorithm that uses the hierarchical approach to reduce the computation time and memory usages of belief propagation approach. Despite the reduction, the algorithm still requires a lot of memory to form different hierarchical levels.

In this paper, a fast correlation-based stereo matching method is presented that computes accurate disparity map from a stereo pair of images in real-time. The proposed algorithm uses two correlation windows (one large and one small size) to compute the disparity map. Although 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 [15]. The rest of the paper is organised 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 datasets (2003, 2005 and 2006) 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. Many 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 [6, 7, 16–18] 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. As the large size window blurs the object boundaries, the small size window results are unreliable in low textured image regions. The pixels near to the disparity discontinuity require windows of different size and shape 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.* [19] 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.* [17] 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, 18, 20] have been proposed which use multiple windows of different sizes or use windows of different orientations to compute the matching cost.

Veksler [21] proposed an algorithm which chooses the appropriate window shape by optimising 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. Although this method works very well, it is not efficient enough for real-time system. In [22], 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 colour dissimilarity and the spatial distance from the centre pixel. These weights regulate the pixel's influence in the matching process. This approach gives very good results but it is computationally expensive and also prone to image noise. The reported computation time of this algorithm on a fast machine is around 1 min which makes it impractical for real-time systems.

3 Proposed algorithm

In this section, we briefly describe the proposed algorithm. Fig. 1 demonstrates the block diagram of the proposed algorithm. The algorithm consists of the following four processing modules: (i) initial matching; (ii) unreliable pixel detection; (iii) disparity interpolation and (4) disparity refinement. In the initial matching step, we compute the initial disparity map by using two different sizes of correlation windows W_l and W_s, respectively. In the 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 the disparity interpolation step, we estimate the disparity for unreliable pixels identified by left–right consistency check [6]. In the 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, that is, the epipolar lines are aligned with the corresponding scanlines.

3.1 Initial matching

To compute the initial disparity map, we use the SAD-based matching approach by using the large correlation window W_l.

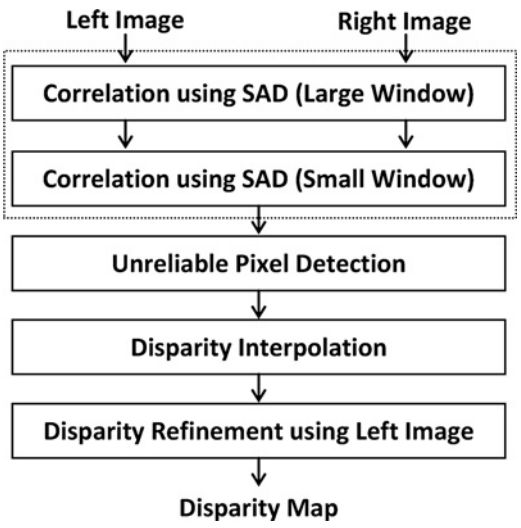

Fig. 1 *Block diagram of the proposed algorithm*

doi: 10.1049/iet-cvi.2011.0077

The matching cost $C(x, y, d)$ of pixel (x, y) for disparity value d is given as follows

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

where $I_l(x, y)$ and $I_r(x, y)$ are the intensities of the pixel in left and right images, respectively. ω represents the size of the matching window W_l. Fig. 2 shows the SAD characteristics under different scenario of matching costs. In Fig. 2*a*, only one minimum matching cost can be found, so the disparity can be determined uniquely as d. However, if more than one minimum matching costs are computed by the matching window, as shown in Fig. 2*b*, which is a common case of non-textured image regions. 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 neighbouring pixels to determine the disparity of such pixels. We propose a penalty term ξ based on gradient and the disparity value of neighbouring pixel in the same scanline to estimate the disparity value in non-textured image regions accurately. The penalty term ξ is given as

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

d' is the disparity of the left or the right neighbouring pixel (x, y') in the same scanline and T is the constant. In our experiments, we fix the value of T to 8. The value of ξ changes according to the change in image gradient. It increases in low gradient image regions (e.g. low or non-textured areas) and decreases in high gradient image regions (e.g. depth discontinuities)

$$1 \geq \left(1 - \frac{|I_l(x, y) - I_l(x, y')|}{255}\right) \geq 0 \quad (3)$$

Fig. 2*c* demonstrates the SAD values computed for different disparity values in non-textured region of Tsukuba image at point (205, 230). We can observe that there is ambiguity because of the number of local minimas. This problem can be solved by using gradient and disparity values of the neighbouring pixels. Fig. 2*d* demonstrates the matching cost computed by using (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 neighbours of a pixel, respectively. The disparity of a pixel (x, y) can be computed as follows

$$d_c(x, y) = \min\left(\arg\min_d C_l(x, y, d), \arg\min_d C_r(x, y, d)\right) \quad (4)$$

After computing the disparity values using the large correlation window, we use the small correlation window W_s 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

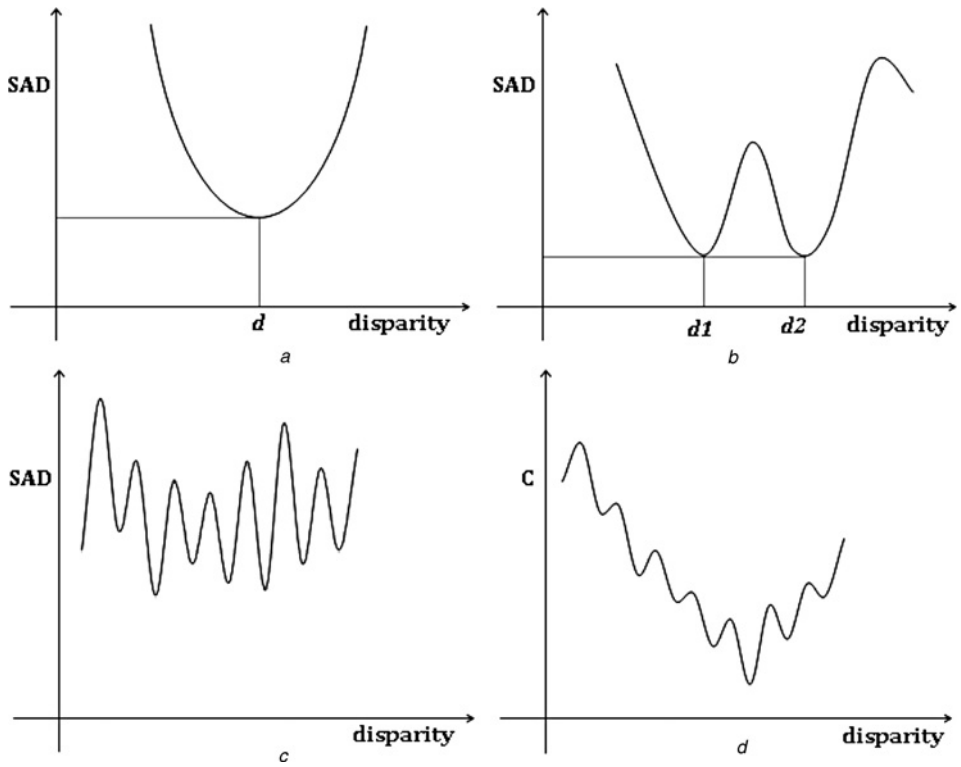

Fig. 2 *Problems in disparity selection*

a Disparity can easily be determined as d for unique minimum value
b Disparity becomes ambiguous in case of multiple minima
c 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
Disparity can be uniquely determined in d

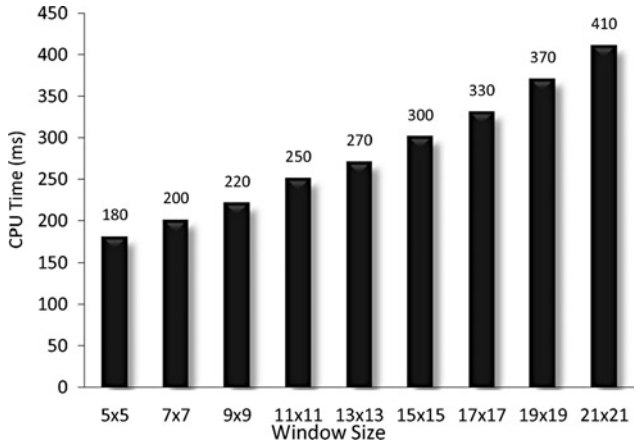

Fig. 3 *Computation time of the proposed algorithm for different window sizes W_l (5 × 5 to 21 × 21) on Tsukuba image (image size 384 × 288 with 16 disparity labels)*

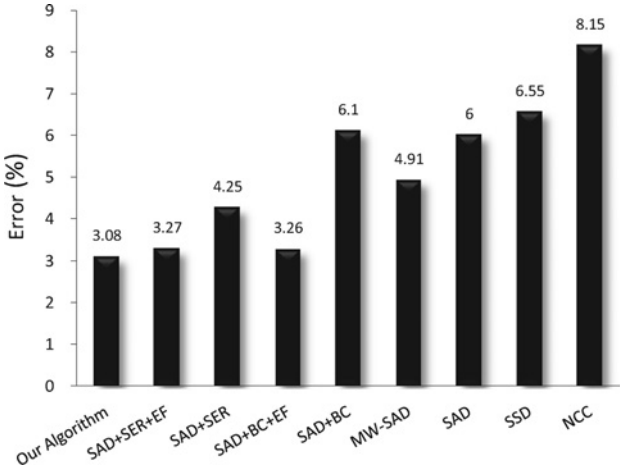

Fig. 4 *Comparison the performance of the proposed algorithm with other correlation-based algorithms*

Graph shows the percentage error in whole image (all) on Tsukuba image

怎么知道这个地方是视差不连续区域呢？

distance of half the size of correlation window [17]. 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 W_l. While computing the disparity of such pixels, we restrict the evaluation of the small correlation window only within those disparity values that are carried by neighbouring pixels. The disparity of such pixels can be computed as

$$d_c(x, y) = \arg\min_{d \in N} C(x, y, d) \tag{5}$$

where $d_c(x, y)$ is the disparity of the pixel (x, y) and N represents the disparity values of the neighbouring pixels. Since we are computing the disparity of the pixels that reside near the depth discontinuity regions, we compute the cost $C(\cdot)$ without using the penalty term ξ as mentioned in (1). Restricting the disparity search within the disparity values of the neighbouring pixels not only helps to avoid false local minima, but it also speeds up the matching process.

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 [23] 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 images 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

$$\left|d_l(x, y) - d_r(x, y'')\right| < 1 \tag{6}$$

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 (6) are marked as unreliable pixels.

3.3 Disparity interpolation

In Section 3.2, the left–right consistency check filters out the 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 neighbouring pixels. For each unreliable pixel, we search for pixels with reliable disparity value in its eight neighbouring pixels. For all these reliable pixels, we compute the distance between the intensities of unreliable pixel and its reliable neighbour in the reference image. The distance between two points p and q can be

Table 1 Comparison of the proposed method with other local real-time stereo-matching algorithms listed on Middlebury evaluation table for absolute disparity error >1

Algorithm	Tsukuba			Venus			Teddy			Cones			percentage error
	nocc	all	disc	nocc	all	disc	nocc	all	disc	nocc	all	disc	
RTBFV [25]	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 [24]	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
Results-1	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 [26]	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 [27]	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 [28]	5.90	7.26	21.0	1.35	1.91	11.2	10.5	17.2	22.2	5.34	11.9	14.9	10.9
Results-2	2.80	4.20	13.4	1.64	2.44	14.8	11.0	16.4	24.9	8.89	14.6	17.7	11.1
SAD + IGMCT [29]	5.81	7.14	22.6	2.61	3.33	25.3	9.79	15.5	25.7	5.08	11.5	15.0	12.5
SSD + MF[1]	5.23	7.07	24.1	3.74	5.16	11.9	16.5	24.8	32.9	10.9	19.8	26.3	15.7

Complete set of results can be found at http://vision.middlebury.edu/stereo/eval/

doi: 10.1049/iet-cvi.2011.0077

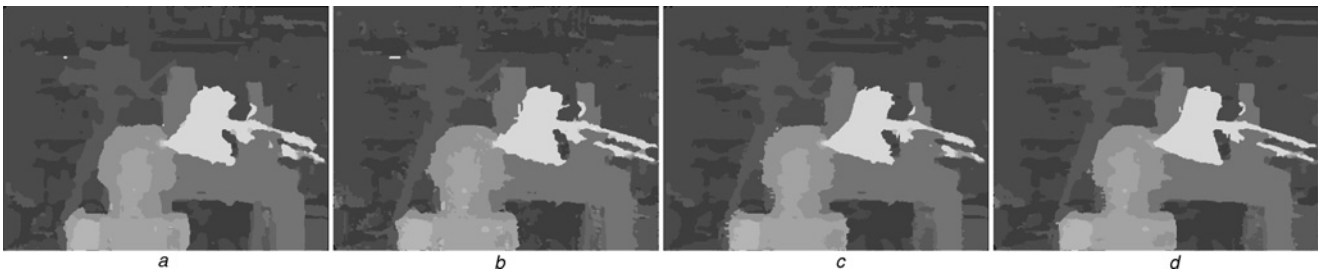

Fig. 5 *Results of the proposed algorithm*

a Without using small correlation window W_s and the disparity refinement step
b Without using the disparity refinement step
c Without using small correlation window W_s
d With all four steps on Tsukuba image

Table 2 Quantitative results obtained with and without using small correlation window W_s by the proposed algorithm

Algorithm	Tsukuba			Venus			Teddy			Cones			percentage error
	nocc	all	disc	nocc	all	disc	nocc	all	disc	nocc	all	disc	
with window W_s	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
without window W_s	3.22	4.25	17.1	1.61	2.16	15.2	11.3	16.3	27.5	8.42	13.9	19.6	11.7

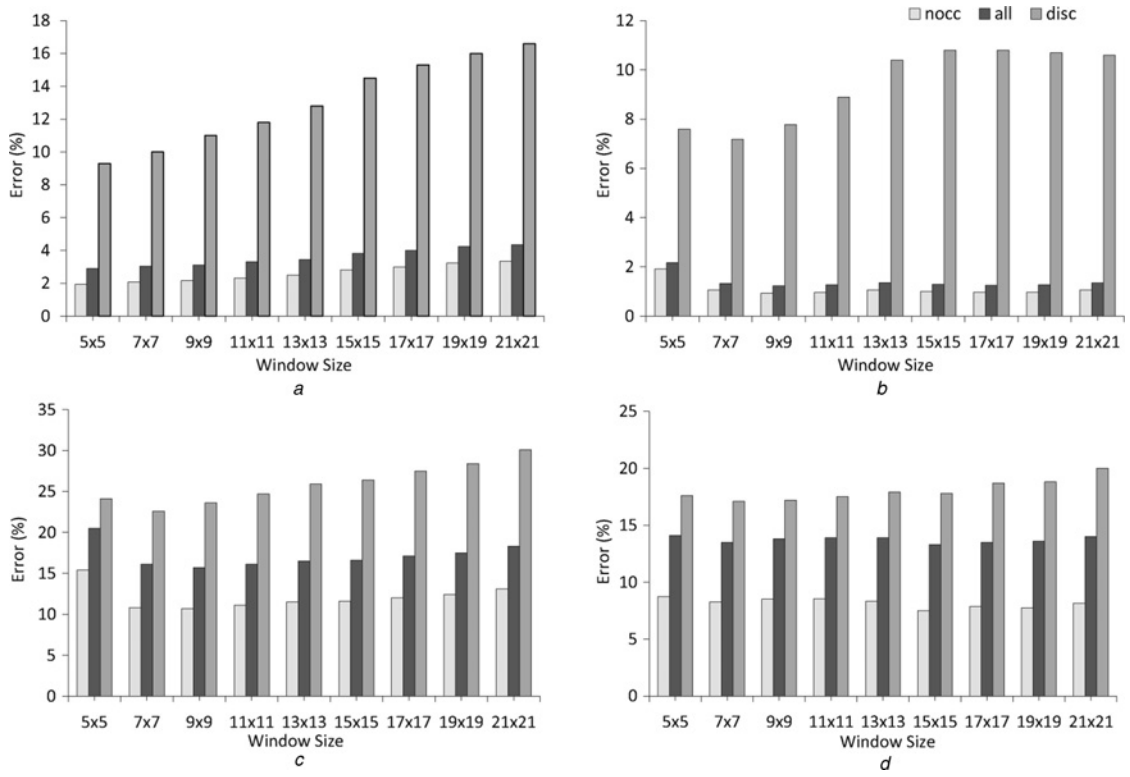

Fig. 6 *Percentage error in non-occluded (nocc), whole image (all) and near depth discontinuities (disc) for different window sizes W_l (5 × 5 to 21 × 21) for all four images (Tsukuba, Venus, Teddy and Cones)*

a Tsukuba
b Venus
c Teddy
d Cones

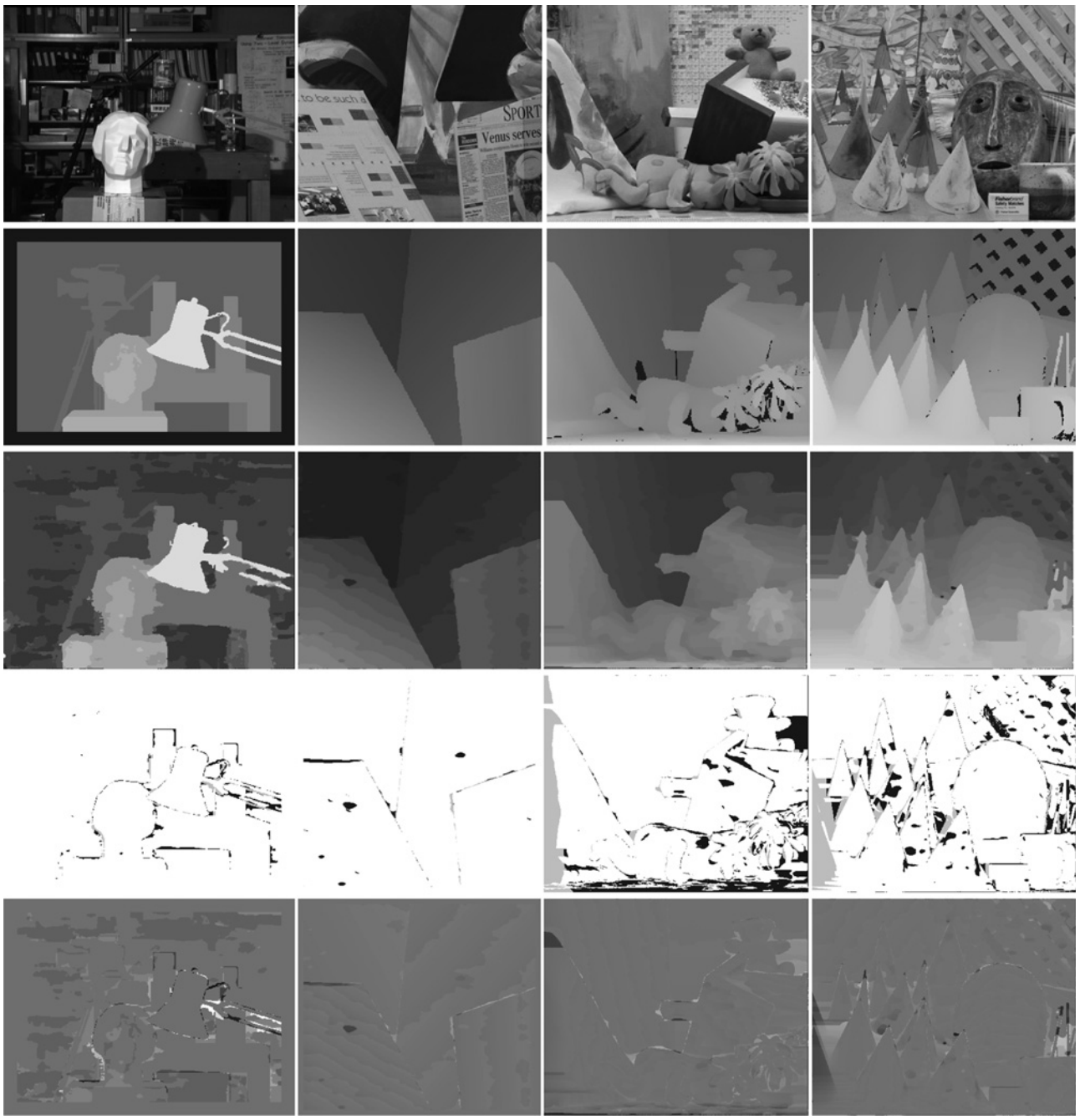

Fig. 7 *Results on the Middlebury dataset (Tsukuba, Venus, Teddy and Cones)*

First row shows the left images, second row shows the ground truth disparity map, third row shows the results obtained by using our algorithm and the fourth and fifth rows show the bad pixel maps and signed disparity error for error threshold value 1.0, respectively

calculated as

$$D(p, q) = \sqrt{\left(r_p - r_q\right)^2 + \left(g_p - g_q\right)^2 + \left(b_p - b_q\right)^2} \quad (7)$$

where r_p, g_p, b_p and r_q, g_q, b_q are the intensity values of points p and q, respectively. We assign the disparity value of the reliable pixel, which has the minimum distance from unreliable pixel in reference image.

3.4 Disparity refinement

To remove the outliers for each colour segment, plane fitting method is used widely. It is assumed that the neighbouring pixels, which having the same intensity values will also have the same disparity values. Although this method increases the accuracy of the algorithm, it requires the colour-segmented image as an input, which makes it computationally expensive. Here, we use the disparity refinement approach that has been proposed in [24]. The algorithm (see Algorithm 1) uses the colour information of the reference image to refine the disparity map without performing any image segmentation.

Algorithm 1: Disparity map refinement by using the colour information of the reference image.

1: for each scanline in reference image do
2: for all p in scanline do
3: index ← 0;

doi: 10.1049/iet-cvi.2011.0077

Fig. 8 *Few image frames extracted from the stereo vision system running in our laboratory*
First and second rows show the left and the right images, whereas the third row shows the corresponding disparity maps

4: mdif ← ∞;
5: for q from $(p-w)$ to $(p+w)$ do
6: if $p \neq q$ then
7: dif ← $d(p, q)$;
8: if mdif > dif then
9: index ← q;
10: mdif ← dif;
11: end if
12: end if
13: end for
14: disp(p) ← min(disp(p), disp(q));
15: end for
16: end for

4 Experiments

The Middlebury stereo benchmark dataset [15] has been used to evaluate the performance of the proposed algorithm. In our experiment, the small window size W_s is chosen as 3×3 for all test images, whereas the large window size W_l is set to 9×9. The proposed algorithm has been implemented using C language and compiled with GNU gcc compiler performing under a 3 GHz computer running with Linux operating system. We used the sliding window approach proposed in [2] to speed up the matching cost computation of large correlation window W_l. The serial CPU implementation of the proposed algorithm achieved more than 8 million disparity evaluations per second, whereas the parallel CPU implementation was able to achieve more than 10 million disparity evaluations per second. However, the computation time of the algorithm depends on the window size.

Fig. 3 shows the computation time (in ms) of the proposed algorithm for different window sizes. The computation time is estimated for the serial CPU implementation of the proposed algorithm on Tsukuba image. It has been shown that the increase in window size also increases the computation time of the proposed algorithm. However, the real-time performance can be achieved for a large range of matching

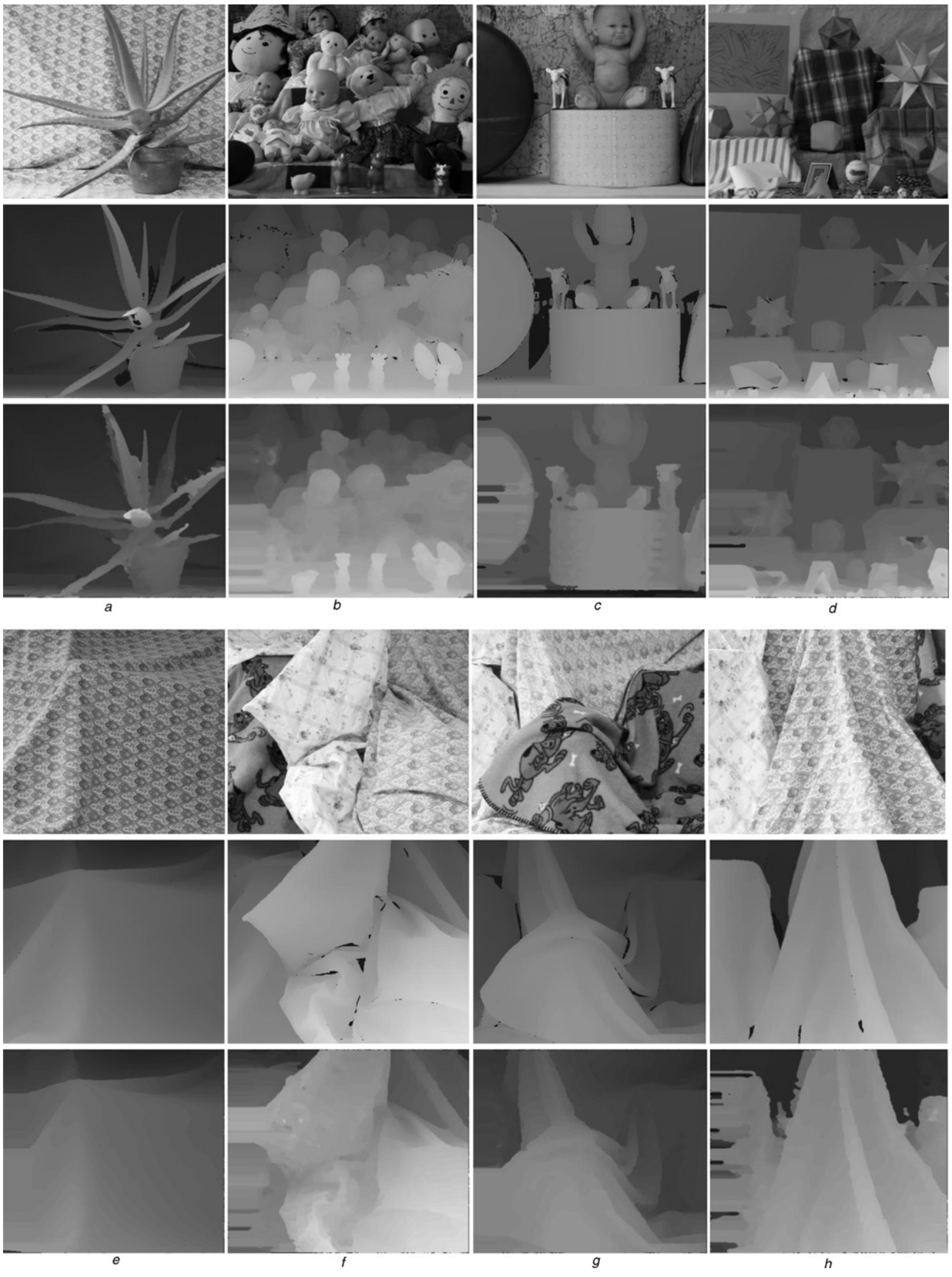

Fig. 9 *Results on the new Middlebury dataset*

First row in each column represents the left image, second row shows the ground truth image and third row shows our results

a Aloe
b Dolls
c Babys
d Moebius
e Cloth 1
f Cloth 2
g Cloth 3
h Cloth 4

doi: 10.1049/iet-cvi.2011.0077

Fig. 10 *Results on the new Middlebury dataset*

First row in each column represents the left image, second row shows the ground truth image and third row shows our results
a Books *d* Medi1 *g* Rock2
b Rock1 *e* Wood1 *h* Reindeer
c Baby2 *f* Flowerpot

windows of different sizes. The quantitative performance of the proposed method for standard Middlebury dataset (Tsukuba, Venus, Teddy and cones) is summarised 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. In the table, the results of the proposed algorithm is listed as Results-1. Results-2 shows the results with the same parameters without performing the disparity refinement (Section 3.4). As shown in Table 1, the proposed algorithm is listed at third place out of eight local real-time stereo-matching algorithms.

Fig. 4 compares the performance of the proposed algorithm with other fixed size window-based correlation algorithms. It shows the percentage error in whole image (all) computed on Tsukuba image. For Tsukuba image pair, we compared the proposed algorithm with standard correlation-based algorithms (SAD, SSD and NCC) and other window-based fast stereo-matching algorithm such as multi-window (MW-SAD) [5], SAD with border correction (SAD + BC) and SAD with border correction and error filtering (SAD + BC + EF) [17], SAD with systematic error reduction (SAD + SER) and SAD with systematic error reduction and error filtering (SAD + SER + EF) [3]. These algorithms use different parameters and different number of windows of different sizes; so to make a fair comparison between them, for each of these algorithms, the results are computed for the best parameter combinations (i.e. which gives the lowest error) reported by their corresponding authors. It is shown that the proposed algorithm yielded an error of 3.08% which is superior to the other fixed size window-based correlation algorithms.

Fig. 5 shows the results of the proposed algorithm for (a) without using small correlation window W_s and the disparity refinement step, (b) without using the disparity refinement step, (c) without using small correlation window W_s and (d) with all four steps on Tsukuba image. The results show that the use of small correlation window W_s and disparity refinement step improves the accuracy at object boundaries. Table 2 shows the quantitative results obtained with and without using small correlation window W_s by the proposed algorithm. It can be seen that the small correlation window W_s plays an important role in proposed algorithm and improves the accuracy significantly.

Fig. 6 demonstrates the performance of the proposed algorithm with different window sizes. As the small

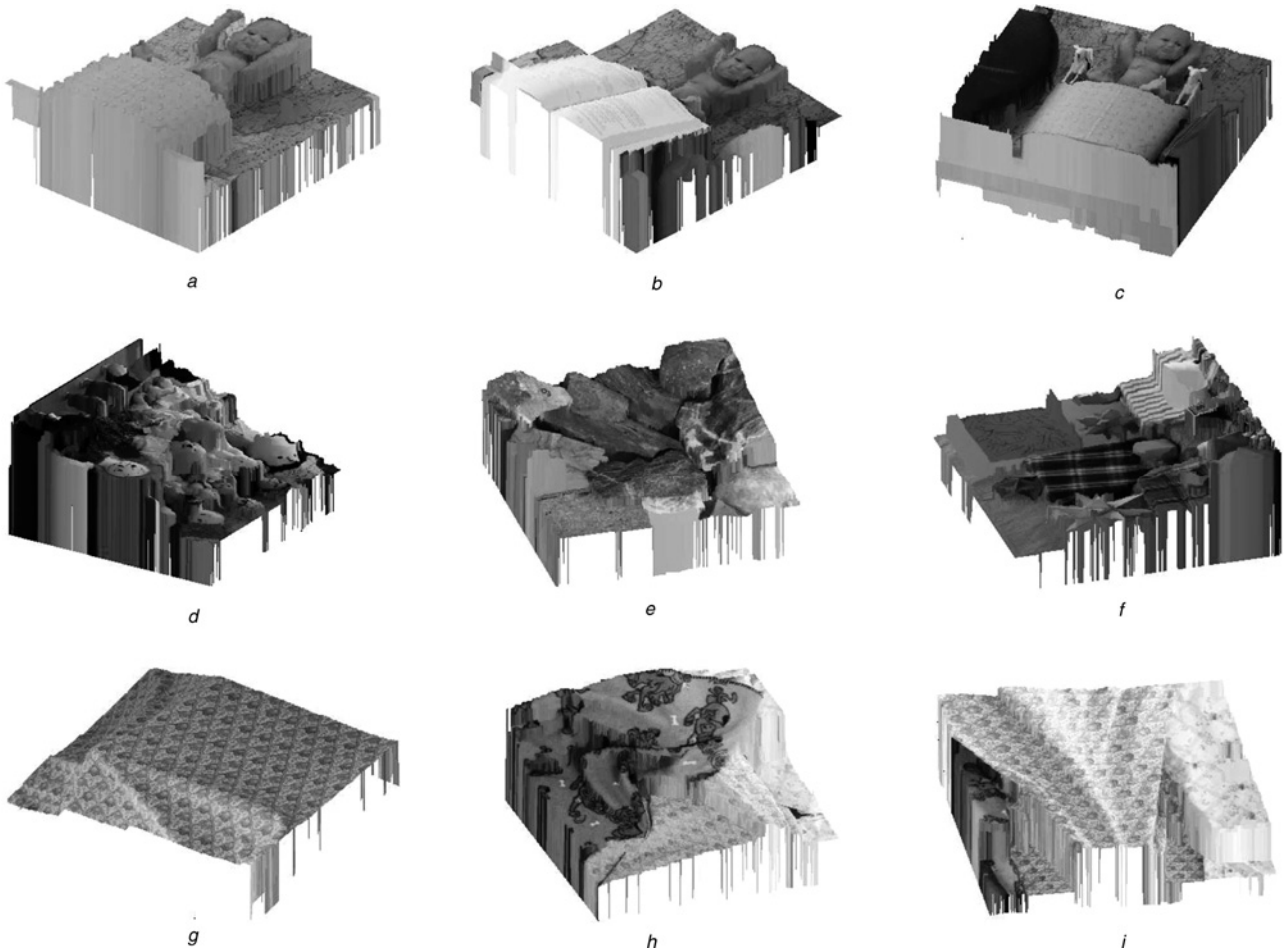

Fig. 11 *3D reconstruction of the left images by using the disparity maps computed by using the proposed algorithm*

a Baby1 *b* Baby2 *c* Baby 3 *d* Dolls *e* Rock1 *f* Moebius *g* Cloth1 *h* Cloth3 *i* Cloth4

doi: 10.1049/iet-cvi.2011.0077

window size W_s is fixed to 3×3, we change the size of the first window (described as large window W_l in proposed algorithm) used during initial matching operation. The window size W_l has been changed from 5×5 to 21×21 pixels. In the graphs of figure, the percentage error in non-occluded, whole image and near depth discontinuities for Tsukuba, Venus, Teddy and Cones images on using different window sizes are shown. The error graphs show that unlike other window-based approaches, the proposed algorithm is not very sensitive to matching window size and the change in the size of matching window does not affect the performance of the algorithm significantly.

参数的敏感性验证

Fig. 7 shows the qualitative results for standard Middlebury dataset. The first and second rows show the left images and the corresponding ground truth images. Third row shows the disparity maps computed by using the proposed algorithm. The fourth and fifth rows show the error maps and signed disparity errors for all four images, respectively. These error maps have been calculated by using Middlebury evaluation system for the error threshold value set as one. The black pixels in the error maps represent the error in non-occluded image regions, whereas the grey pixels represent the errors in occluded regions. The white image area represents the correct matches.

Fig. 8 shows few image frames that have been extracted from the stereo vision system implemented using the proposed algorithm running in our laboratory. The first and second rows show the left and the right images, respectively, whereas the third row shows the corresponding disparity maps. The disparity range of these images is around 60 pixels. The Middlebury datasets contain the images that have been taken in ideal lighting environment, however the conditions in real environment may be quite different. The disparity maps in Fig. 8 show that the proposed algorithm works well for both the Middlebury datasets as well as in real environment.

Figs. 9 and 10 further show 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, that is, complex geometry (dolls, Moebius and Reindeer), repetitive patterns (Aloe and Cloth images) and non-textured image regions (Wood1, Books and Medi1). The images in these datasets also have large disparity ranges (more than 60 pixels), resulting in large occlusions. Owing to large occlusions, repetitive patterns and higher percentage of untextured surfaces, the new Middlebury datasets are much more challenging as compared to the previous stereo dataset which contains images such as Teddy and Cones. The disparity maps shown in these figures have some streaking artifacts to the left side of the image. These image regions are occluded and the pixels in these image regions have no corresponding match in the right image. We have propagated the disparity in such image regions from their right neighbouring pixels.

Fig. 11 shows the 3D reconstruction of a few images (Baby1, Baby2, Baby3, Dolls, Rock1, Moebius, Cloth1, Cloth2 and Cloth3) that have been taken from new Middlebury datasets, by using the disparity maps computed by using the proposed algorithm. The experimental results clearly show that the proposed algorithm works very well in case of repetitive patterns, 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

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 dense disparity map. Although large correlation window gives good results at non-textured image regions, the small window improves the performance at depth discontinuities. The algorithm uses simple mathematical operations and can be easily implemented on GPU. The CPU implementation of the proposed algorithm computes dense disparity maps at a speed of more than 10 frame/s in case of 320×240-sized image pair with disparity value 16. The computation time can be significantly reduced by hardware specific optimisation or GPU implementation of the algorithm. The proposed method can be used in real-time applications to reconstruct the 3D structures with great accuracy at object boundaries.

6 References

1 Scharstein, D., Szeliski, R.: 'A taxonomy and evaluation of dense two-frame stereo correspondence algorithms', *Int. J. Comput. Vis.*, 2002, **47**, (1), pp. 7–42
2 Faugeras, O., Hotz, B., Mathieu, M., *et al.*: '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 Recognit. Lett.*, 2005, **26**, pp. 2221–2231
4 Stefano, L., Marchionni, M., Mattoccia, S.: 'A fast area-based stereo matching algorithm', *Image Vis. Comput.*, 2004, **22**, (12), pp. 983–1005
5 Fusiello, A., Roberto, V., Trucco, E.: 'Efficient stereo with multiple windowing'. IEEE Conf. Computer Vision and Pattern Recognition, 1997, pp. 858–863
6 Fua, P.: 'A parallel stereo algorithm that produces dense depth maps and preserves image features', *Mach. Vis. Appl.*, 1993, **6**, (1), pp. 35–49
7 Adhyapak, S., Kehtarnavaz, N., Nadin, M.: 'Stereo matching via selective multiple windows', *J. Electron. Imag.*, 2007, **16**, (1), pp. 013012
8 Kanade, T., Okutomi, M.: 'A stereo matching algorithm with an adaptive window: theory and experiments', *IEEE Trans. Pattern Anal. Mach. Intell.*, 1994, **16**, (9), pp. 920–932
9 Felzenszwalb, P., Huttenlocher, D.: 'Efficient belief propagation for early vision'. IEEE Conf. Comput. Vis. Pattern Recognit., 2004, vol. 1, pp. 261–268
10 Birchfield, S., Tomasi, C.: 'Depth discontinuities by pixel-to-pixel stereo', *Int. J. Comput. Vis.*, 1999, **25**, pp. 1073–1080
11 Boykov, Y., Veksler, O., Zabih, R.: 'Fast approximate energy minimization via graph cuts', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2001, **23**, (11), pp. 1222–1239
12 Sun, J., Zheng, N., Shum, H.: 'Stereo matching using belief propagation', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2003, **25**, (7), pp. 787–800
13 Criminisi, A., Blake, A., Rother, C., Shotton, J., Torr, P.H.: 'Efficient dense stereo with occlusions for new view-synthesis by four-state dynamic programming', *Int. J. Comput. Vis.*, 2007, **71**, (1), pp. 89–110
14 Sara, R.: 'How to teach stereoscopic matching?' ELMAR, 2010, pp. 445–453
15 http://vision.middlebury.edu/stereo, accessed 16th July 2012
16 Yoon, K.-J., Kweon, I.S.: 'Adaptive support-weight approach for correspondence search', *IEEE Trans. Pattern Anal. Mach. Intell.*, 2006, **28**, (4), pp. 650–656
17 Hirschmuller, H., Innocent, P., Garibaldi, J.: 'Real-time correlation-based stereo vision with reduced border errors', *Int. J. Comput. Vis.*, 2002, **47**, pp. 229–246
18 Okutomi, M., Katayama, Y., Oka, S.: 'A simple stereo algorithm to recover precise object boundaries and smooth surfaces', *Int. J. Comput. Vis.*, 2002, **47**, (1), pp. 261–273
19 Boykov, Y., Veksler, O., Zabih, R.: 'A variable window approach to early vision', *IEEE Trans. Pattern Anal. Mach. Intell.*, 1998, **20**, (12), pp. 1283–1294
20 Jeon, J., Kim, C., Ho, Y.S.: 'Sharp and dense disparity maps using multiple windows'. Lecture Notes Computer Science, 2002, pp. 1057–1064

21 Veksler, O.: 'Stereo matching by compact window via minimum ratio cycle'. IEEE Int. Conf. Comput. Vis., 2001, vol. 1, pp. 540–547
22 Veksler, O.: 'Fast variable window for stereo correspondence using integral images'. IEEE Conf. Computer Vision and Pattern Recognition, 2003, vol. 1, pp. 556–561
23 Marr, D., Poggio, T.: 'A computational theory of human stereo vision'. Proc. Royal Society of London, Series B, 1979, vol. 240, no. 1156, pp. 301–328
24 Gupta, R., Cho, S.-Y.: 'Real-time stereo matching using adaptive binary window' (3D Data Processing, Visualization and Transmission, 2010)
25 Zhang, K., Lu, J., Lafruit, G., Lauwereins, R., Gool, L.V.: 'Real-time accurate stereo with bitwise fast voting on Cuda'. Int. Conf. Computer Vision Workshops, 2009, pp. 540–547
26 Humenberger, M., Zinner, C., Weber, M., Kubinger, W., Vincze, M.: 'A fast stereo matching algorithm suitable for embedded real-time systems', *Comput. Vis. Image Underst.*, 2010, 114, (11), pp. 1180–1202
27 Gong, M., Yang, Y.: 'Near real-time reliable stereo matching using programmable graphics hardware'. IEEE Conf. Computer Vision and Pattern Recognition, 2005, pp. 924–931
28 Richardt, C., Orr, D., Davies, I., Criminisi, A., Dodgson, N.: 'Real-time spatiotemporal stereo matching using the dual-cross-bilateral grid'. European Conf. Computer Vision, 2010, vol. 6313, pp. 510–523
29 Ambrosch, K., Kubinger, W.: 'Accurate hardware-based stereo vision', *Comput. Vis. Image Underst.*, 2010, **114**, (11), pp. 1303–1316

doi: 10.1049/iet-cvi.2011.0077