

# Stereo Matching with Color-Weighted Correlation, Hierarchical Belief Propagation, and Occlusion Handling

Qingxiong Yang, *Student Member, IEEE*, Liang Wang, *Student Member, IEEE*,  
Ruiyang Yang, *Member, IEEE*, Henrik Stewénius, *Member, IEEE*, and David Nistér, *Member, IEEE*

**Abstract**—In this paper, we formulate a stereo matching algorithm with careful handling of disparity, discontinuity, and occlusion. The algorithm works with a global matching stereo model based on an energy-minimization framework. The global energy contains two terms, the data term and the smoothness term. The data term is first approximated by a color-weighted correlation, then refined in occluded and low-texture areas in a *repeated* application of a hierarchical loopy belief propagation algorithm. The experimental results are evaluated on the Middlebury data sets, showing that our algorithm is the *top* performer among all the algorithms listed there.

**Index Terms**—3D/stereo scene analysis, segmentation, Markov random fields.

## 1 INTRODUCTION

STEREO is one of the most extensively researched topics in computer vision. Stereo research has recently experienced somewhat of a new era, as a result of publicly available performance testing such as the Middlebury benchmark [17], which has allowed researchers to compare their algorithms against all the state-of-the-art algorithms.

The stereo algorithm presented in this paper springs from the popular energy minimization framework that is the basis for most high-performance algorithms such as graph cuts [4], [16] and belief propagation (BP) [20], [19]. In this framework, there is typically a data term and a smoothness term, where the data term consists of the matching error implied by the extracted disparity map and the smoothness term encodes the prior assumption that the world surfaces are piecewise smooth. However, the algorithm presented in this paper differs from the normal framework in that, in the final stages of the algorithm, the data term is updated based on the current understanding of which pixels in the reference image are occluded or unstable due to low texture. This is the main contribution of the paper.

Another contribution of the paper is the successful integration of several recent ideas, including color-weighted

correlation [27], hierarchical BP [9], left/right checking [7], color segmentation [6], plane fitting [21], and depth enhancement. We have evaluated our stereo algorithm using the Middlebury benchmark and showed that our algorithm is the top performer in both integer-based and subpixel accuracy. While quality is our primary goal, we also take speed into account during the design of our algorithm. In particular, with the comparable quality, the method suitable for parallel execution is preferred. For instance, we prefer hierarchical BP to Graph cuts because it is easier for parallel hardware implementation while achieving similar performance [4], [18], [22]; so is the left/right checking component for occlusion handling [7]. Note that color-weighted correlation [27] is a parallel algorithm too. Hence, except for mean-shift, our algorithm is well suited for parallel hardware acceleration, e.g., the GPU or the IBM's Cell Processor. In addition, a fast-converging BP approach is proposed, which removes the redundant computation involved in the message updating step. Unlike standard BP, the runtime of fast-converging BP is sublinear to the number of iterations.

The paper is organized as follows: Section 2 gives a high-level overview of the approach. In Section 3, we present the detailed equations for all the building blocks. Additionally, a depth-enhanced approach is presented in Section 4. Section 5 reports results showing that our algorithm is currently the strongest available on the Middlebury data sets. Section 6 concludes.

## 2 OVERVIEW OF THE APPROACH

The algorithm can be partitioned into three blocks, initialization (Fig. 1), pixel classification (Fig. 2), and iterative refinement (Fig. 3). In the initialization step (see Fig. 1), the correlation volume is first computed. A basic way to construct the correlation volume is to compute the absolute difference of the luminance of the corresponding

- Q. Yang is with the Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, 1520 Beckman Institute, 405 N. Mathews, Urbana, IL 61801. E-mail: quang6@uiuc.edu.
- L. Wang and R. Yang are with the Computer Science Department, University of Kentucky, One Quality Street, Suite 800, Lexington, KY 40507. E-mail: {lwang, ryang}@cs.uky.edu.
- H. Stewénius is with Google Switzerland, Brandschenkestrasse 110, 8002 Zürich, Switzerland. E-mail: hstewenius@gmail.com.
- D. Nistér is with Microsoft Corp., One Microsoft Way, Redmond, WA 98052. E-mail: dnister@microsoft.com.

Manuscript received 10 Jan. 2007; revised 24 Oct. 2007; accepted 12 Mar. 2008; published online 14 Apr. 2008.

Recommended for acceptance by K. Daniilidis.

For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0013-0107. Digital Object Identifier no. 10.1109/TPAMI.2008.99.

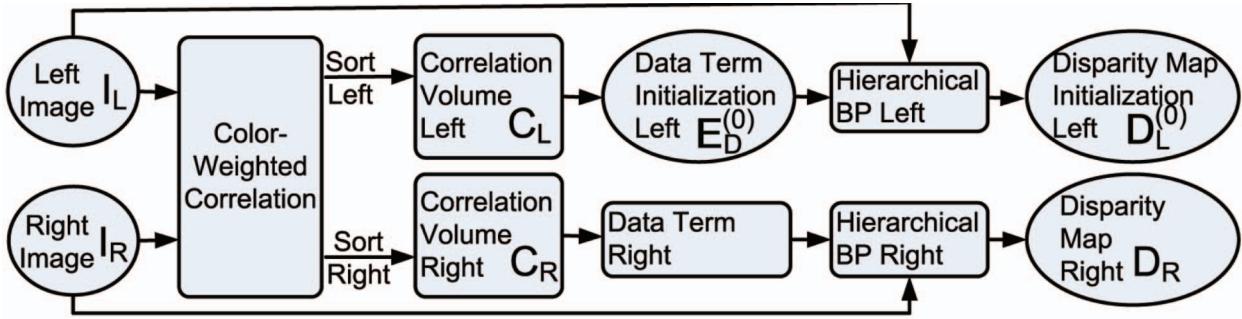


Fig. 1. The initialization module. Hierarchical BP is run with both the left and right images as the reference image. The initial data term is computed based on the color-weighted correlation and then fed to the hierarchical BP module for disparity map estimation, see the text for more details.

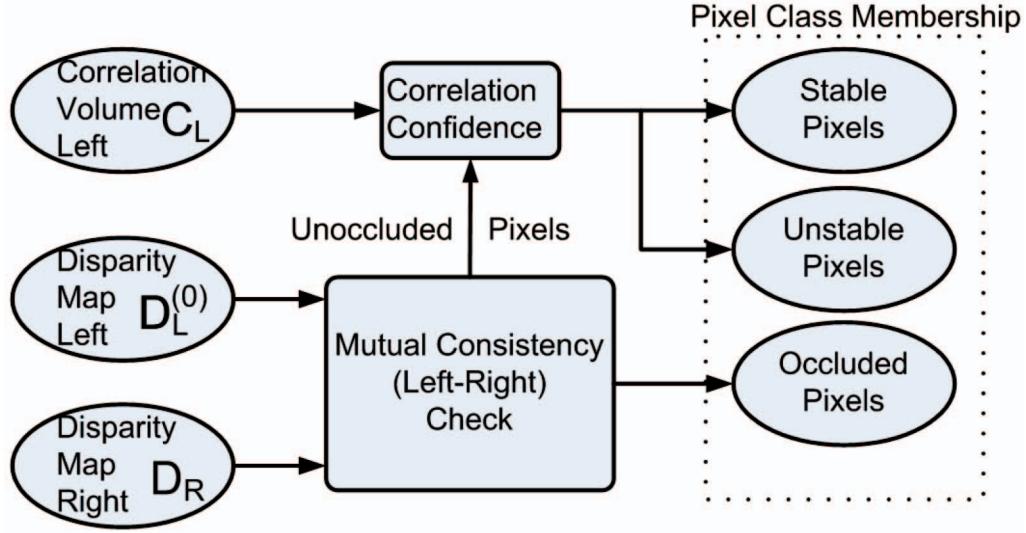


Fig. 2. The pixel classification module. Pixels are classified into occluded pixels, unstable pixels, and stable pixels. The occluded pixels are the ones that fail a mutual consistency check. The nonoccluded pixels are then further divided into stable and unstable pixels based on a confidence measure derived from the correlation volume.

pixels in the left and right image, but there are many other methods for correlation volume construction. For instance, Sun et al. [20] use Birchfield and Tomasi's pixel dissimilarity [1] to construct the correlation volume, and Felzenszwalb [9] suggests smoothing the image first before calculating the pixel difference. In this work, we are using color-weighted correlation to build the correlation volume, in a similar manner as was recently described by Yoon and Kweon [27]. The color-weighting makes the match scores less sensitive to occlusion boundaries by using the fact that occlusion boundaries most often cause color discontinuities as well. The initialization is applied twice with both the left and the right image as the reference image, respectively. This is done just to support a subsequent mutual consistency check (often called left-right check) that takes place in the pixel classification block. The initial data term  $E_D^{(0)}$  is computed from the color-weighted correlation volume  $C_L$ . As a global energy minimization approach, a hierarchical BP module is employed with the data term and the corresponding reference image as input for the disparity map estimation. The hierarchical BP is performed in a manner similar to Felzenszwalb and Huttenlocher [9], resulting in the initial left and right disparity maps  $D_L^{(0)}$  and  $D_R^{(0)}$ , respectively. The left disparity map  $D_L^{(0)}$  and the initial data term  $E_D^{(0)}$  is given

an iteration index  $i = 0$  here because it will be iteratively refined. In summary, the initialization step generates the following output: the initial left and right disparity maps  $D_L^{(0)}$ ,  $D_R^{(0)}$ , and the initial data term  $E_D^{(0)}$ .

In the pixel classification module (see Fig. 2), pixels are assigned one of three possible labels: *occluded*, *stable*, or *unstable*. The occluded pixels are the ones that fail the mutual consistency check that is performed using  $D_L^{(0)}$  and  $D_R^{(0)}$ . The pixels that pass the mutual consistency check are then labeled stable or unstable based on a confidence measure derived from the left correlation volume, which measures if the peak in the correlation score is distinctive enough that the local disparity can be considered stable. The output from the pixel classification module is the pixel class membership.

In the iterative refinement module (see Fig. 3), the initial left disparity map  $D_L^{(0)}$ , the left image  $I_L$ , the pixel class membership, and the initial data term  $E_D^{(0)}$  are all used as input. The goal here is to propagate information from the stable pixels to the unstable and the occluded pixels. This is done using color segmentation and plane fitting in a way inspired by Tao and Sawhney [21]. In our work, we use color segments in  $I_L$  extracted by mean shift [6]. In each color segment, the disparity values for the stable pixels are used in a plane fitting procedure. Note that the disparity

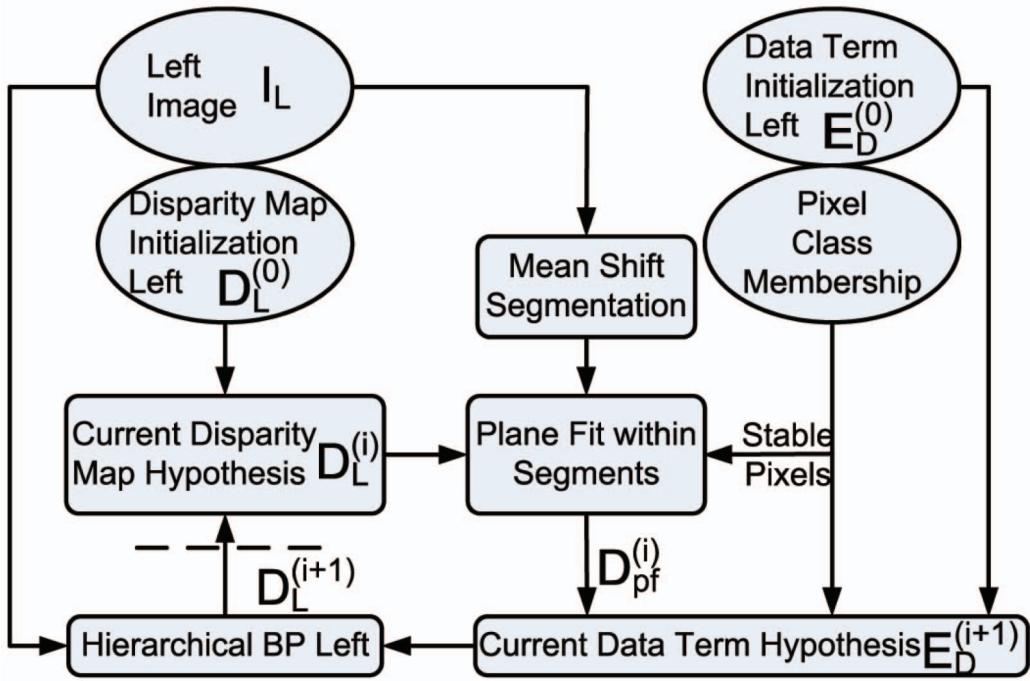


Fig. 3. The iterative refinement block, where the goal is to propagate information from the stable pixels to the unstable and the occluded pixels. Mean shift color segmentation is used to derive segments. Within each segment, the plane fitting is then applied to the stable pixels using the disparity values from the current disparity map hypothesis. The result  $D_{pf}^{(i)}$  from the plane fitting is then used together with the correlation volume and the pixel class membership to produce a new approximation  $E_D^{(i+1)}$  of the data term. The data term is used with the reference image in another round of hierarchical BP. This gives a new disparity map hypothesis  $D_L^{(i+1)}$ , which is fed back into the process.

values used here are taken from the current hypothesis  $D_L^{(i)}$  for the left disparity map. This disparity map is first initialized with the left disparity map  $D_L^{(0)}$  given by the initialization module. The result of plane fitting within color segments is then used together with the pixel class membership and the initial data term  $E_D^{(0)}$  to give the current data term hypothesis  $E_D^{(i+1)}$ , which is then fed to the hierarchical BP module. Effectively, the plane-fitted depth map is used as a regularization for the new disparity estimation. The hierarchical BP yields the updated disparity map hypothesis  $D_L^{(i+1)}$ , which is iteratively fed back into the plane fitting procedure. Note that the right disparity map  $D_R$  is not updated in our implementation for efficiency and simplicity, and our experimental results show that, as a simplified method, it is still the top performer on the Middlebury data sets.

### 3 DETAILED DESCRIPTION

In this section, we give a more detailed description of the building blocks outlined above. The order of description follows the above outline through Figs. 1, 2, and 3. A detailed explanation of the hierarchical BP algorithm is provided afterwards. To further improve the speed of standard BP algorithms, a fast-converging BP approach is developed.

#### 3.1 Initialization

The main building blocks of the initialization module, shown in Fig. 1, are color-weighted correlation, initial data term, and hierarchical BP.

The objective of the color-weighted cost aggregation is to initialize a reliable correlation volume. To obtain more

accurate results on both smooth and discontinuous regions, an appropriate window should be selected adaptively for each pixel during the cost aggregation step. That is, the window should be large enough to cover sufficient area in untextured regions, while small enough to avoid crossing depth discontinuities. Many methods [14], [3], [23], [24] have been proposed to solve this ambiguity problem.

In our implementation, we use an amended version of the color-weighted approach proposed recently by Yoon and Kweon [27]. In this method, instead of finding an optimal support window, adaptive support-weights are assigned to pixels in some large window with side-length  $\alpha_{cw}$  based both on the color proximity and the spatial proximity to the pixel under consideration (the central pixel of the support window).

In Yoon and Kweon's work, the similarity between two pixels within the support window is measured in the CIELab color space. However, we simply measure it in the RGB color space and, instead of using a raw pixel difference, we use the Birchfield and Tomasi's pixel dissimilarity [1] to improve the robustness against the image sampling noise. RGB color space is employed because according to our experiments, the performance of the proposed method is robust to both color spaces. The computation of our correlation volume is depicted as follows, which is mainly a review of the method presented in [27].

Assume the color difference  $\Delta_{xy}$  between pixel  $x$  and  $y$  (in the same image) is expressed as

$$\Delta_{xy} = \left( \sum_{c \in \{r,g,b\}} |I_c(x) - I_c(y)| \right) / 3, \quad (1)$$

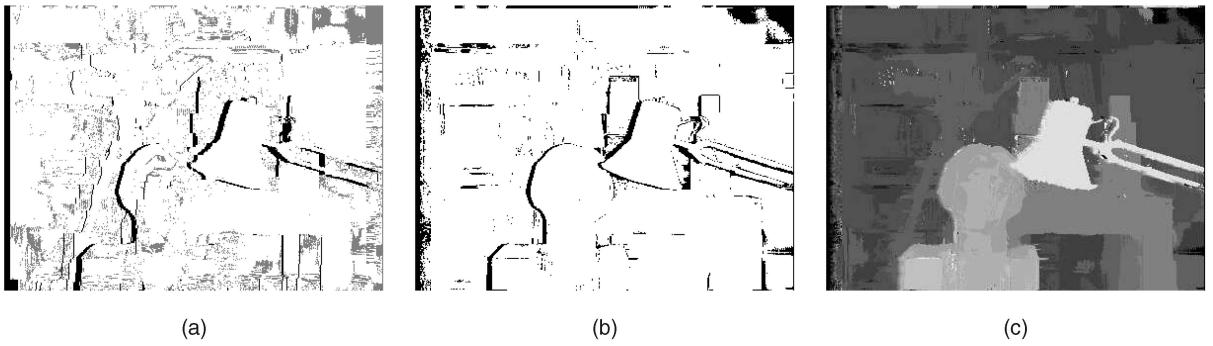


Fig. 4. (a) The result after applying pixel classification on the Tsukuba data set. The black pixels are occluded pixels, the gray pixels are unstable pixels, the others are stable pixels. (b) The bad pixels (black pixels) detected by comparing the disparity map with the ground truth. (c) The corresponding disparity map calculated by applying the winner-takes-all approach to the color-weighted correlation volume  $C_L$ . (a) Pixel class membership. (b) Bad pixels. (c) Disparity map.

where  $I_c$  is the intensity of the color channel  $c$ . The weight of pixel  $x$  in the support window of  $y$  (or vice versa) is then determined using both the color and spatial differences as

$$w_{xy} = e^{-(\beta_{cw}^{-1}\Delta_{xy} + \gamma_{cw}^{-1}\|x-y\|_2)}, \quad (2)$$

where  $\beta_{cw} = 10$  and  $\gamma_{cw} = 21$ . The values of these and subsequent parameters were determined empirically; see Section 5.1 for a discussion of the parameter settings and their sensitivity. The subscripted 2 at the end of (2) is to symbolize the L2 norm.

The correlation volume is then an aggregation with the soft windows defined by the weights as

$$C_{L,x_L}(d_x) = \frac{\sum_{(y_L,y_R) \in W_{x_L} \times W_{x_R}} w_{x_L y_L} w_{x_R y_R} d(y_L, y_R)}{\sum_{(y_L,y_R) \in W_{x_L} \times W_{x_R}} w_{x_L y_L} w_{x_R y_R}},$$

where  $W_x$  is the support window around  $x$  and  $d(y_L, y_R)$  represents the Birchfield and Tomasi's pixel dissimilarity,  $x_L$  and  $y_L$  are pixels in the left image  $I_L$ ,  $x_R = x_L - d_x$  and  $y_R = y_L - d_x$  are the corresponding pixels in the right image  $I_R$ .  $d_x$  is the disparity value of pixel  $x_L$  in the left image.

The initial data term is a truncated linear transform of the correlation volume:

$$E_{D,x_L}^{(0)}(d_x) = \lambda_{bp} \min(C_{L,x_L}(d_x), \eta_{bp}), \quad (3)$$

where  $\lambda_{bp} = 0.2$  is a constant, which is set experimentally, and  $\eta_{bp}$  is set to be twice the average of the correlation volume to exclude the outliers.  $E_D^{(0)}$  is then fed to the hierarchical loopy BP [9] module for disparity hypothesis estimation.

### 3.2 Pixel Classification

The main building blocks of the pixel classification step (see Fig. 2) are the mutual consistency check and the correlation confidence measure.

The mutual consistency check requires that the disparity value from the left and right disparity maps are consistent, i.e.,

$$D_L(x_L) = D_R(x_L - D_L(x_L)), \quad (4)$$

for a particular pixel  $x_L$  in the left image. If this relation does not hold, the pixel is declared occluded. Otherwise, the pixel is declared nonoccluded and passed on to the

correlation confidence measure. Notice that some of the erroneous disparity values due to the lack of texture may fail in this test, but this is rare, because hierarchical BP algorithm tend to propagate the disparity values from outside the low-textured areas into the centers for both the left and right disparity maps. Hence, most of the disparity values in the low-texture areas maybe incorrect in the initial disparity maps but will be consistent for both views. As is shown in Fig. 4, although the upper right corner lacks texture, it is not classified as an occluded area.

The correlation confidence is defined based on how distinctive the highest peak in a pixel's correlation profile is. Assuming that the cost for the best disparity value is  $C_L^1$  and the cost for the second best disparity value is  $C_L^2$ , the correlation confidence is then

$$\left| \frac{C_L^1 - C_L^2}{C_L^2} \right|. \quad (5)$$

If it is above a threshold  $\alpha_s$ , the pixel is declared stable, otherwise, unstable.  $\alpha_s = 0.04$  is set experimentally.

The experimental results of pixel classification on the Tsukuba data set are provided in Fig. 4. It shows that most of the bad pixels are classified into either occluded pixels or unstable pixels.

### 3.3 Iterative Refinement

The main building blocks of the iterative refinement, see Fig. 3, are the mean shift color segmentation, the plane fitting within segments, the data term formulation, and another hierarchical BP process identical to that in Section 3.1.

The mean shift color segmentation is performed as described in [6]; an example is shown in Fig. 5.

The plane fitting is performed in the disparity space and applied per segment. This is done robustly using RANSAC [10] on the disparity values of the stable pixels only. The error threshold for inlier selection is set to be 0.3 and the number of trials is predetermined to be 300. The output  $D_{pf}^{(i)}$  from this step is computed individually for each segment and depends on the ratio of stable pixels of this segment. If the ratio of stable pixels is above a parameter value  $\eta_s = 0.7$ , it means most of the current disparity values for the segment are approximated accurately, so we use  $D_L^{(i)}$  for the stable pixels and, for the remaining pixels (i.e., the unstable

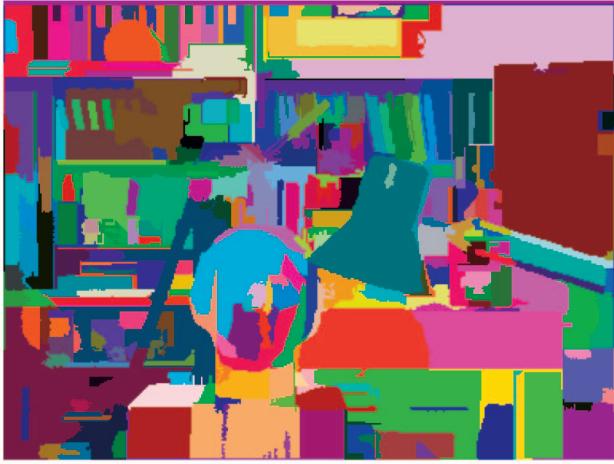


Fig. 5. Color segmentation on the Tsukuba data set.

and occluded pixels), we use the result of the plane fitting. If the ratio of the stable pixels is below  $\eta_s$ , we use the result of the plane fitting for all pixels.

The data term is formulated differently for the occluded, unstable, and stable pixels. The absolute difference,

$$a_i = \left| D_L^{(i+1)} - D_{pf}^{(i)} \right|, \quad (6)$$

between the new disparity map  $D_L^{(i+1)}$  and the plane-fitted disparity map  $D_{pf}^{(i)}$  is used to regularize the new estimation process. The difference is used to define the data term of the occluded, unstable, and stable pixels as

$$E_D^{(i+1)} = \begin{cases} \kappa_o a_i, & \text{if occluded,} \\ E_D^{(0)} + \kappa_u a_i, & \text{if unstable,} \\ E_D^{(0)} + \kappa_s a_i, & \text{if stable,} \end{cases}$$

respectively. The constants  $\kappa_o = 2.0$ ,  $\kappa_u = 0.5$ , and  $\kappa_s = 0.05$  reflect the fact that the unstable and occluded pixels need the most regularization. Hence, the whole iterative refinement step works as employing a local smoothness constraint to make sure that the unstable/occluded pixels will be on the same plane surface as the stable pixels, which have been clustered to the same segment as the unstable/occluded pixels.

### 3.4 Hierarchical Belief Propagation

The core energy minimization of our algorithm is carried out via the hierarchical BP algorithm. Here, we briefly review the max-product BP algorithm [25] we have adopted. The max-product BP algorithm works by passing messages around the graph defined by the four-connected image grid. Each message is a vector of the dimension given by the number of possible labels, and at each iteration, the new messages are computed as follows:

$$\begin{aligned} M_{\mathbf{X}, \mathbf{Y}}^t(d) = \arg \min_{d_X} & \left( E_{D, \mathbf{X}}(d_X) \right. \\ & \left. + \sum_{s \in N(\mathbf{X}), \mathbf{X} \neq \mathbf{Y}} M_{s, \mathbf{X}}^{t-1}(d_X) + h(d_X, d) \right), \end{aligned} \quad (7)$$

where  $M_{\mathbf{X}, \mathbf{Y}}^t$  is the message vector passed from pixel  $\mathbf{X}$  to one of its neighbors  $\mathbf{Y}$ ,  $E_{D, \mathbf{X}}$  is the data term of pixel  $\mathbf{X}$ , and

$h(d_X, d)$  is the jump cost.  $d$  is the label that minimizes the total energy for pixel  $\mathbf{X}$ , which contains the data term and the smoothness term:

$$\begin{aligned} E_{\mathbf{X}}(d) &= E_{D, \mathbf{X}}(d) + E_{S, \mathbf{X}}(d) \\ &= E_{D, \mathbf{X}}(d) + \sum_{\mathbf{Y} \in N(\mathbf{X})} M_{\mathbf{Y}, \mathbf{X}}(d). \end{aligned} \quad (8)$$

The common cost functions for the jump cost  $h(d_X, d)$  are based on the degree of difference between labels. In order to allow for discontinuities, the truncated linear model is commonly adopted:

$$h(d_X, d) = \min(\alpha_{bp}, \rho_{bp} |d_X - d|), \quad (9)$$

where  $\rho_{bp} = 1$  is the rate of increase in the cost and  $\alpha_{bp} = n_d/8$  controls when the cost stops increasing.  $n_d$  is the number of disparity levels. Equation (9) is defined under the assumption of piecewise-constant surfaces. However, the jump cost should decrease at depth edges. The color difference,

$$\delta_{\mathbf{X}, \mathbf{Y}} = \sum_{c \in \{r, g, b\}} |I_c(\mathbf{X}) - I_c(\mathbf{Y})|, \quad (10)$$

between neighboring pixels  $\mathbf{X}$  and  $\mathbf{Y}$  is used to decide the amount of the decrease of the cost since the color edges are likely to coincide with the depth edges. The difference  $\delta_{\mathbf{X}, \mathbf{Y}}$  is normalized to span the interval  $[0, 1]$ . The mean of  $\delta_{\mathbf{X}, \mathbf{Y}}$  over the whole frame  $\delta_{mean}$  is then subtracted out to yield the normalized difference  $\delta_{norm}$ . Defining the cost coefficient

$$\rho_s = 1 - \delta_{norm}, \quad (11)$$

the cost assigned to the pixel pair  $(\mathbf{X}, \mathbf{Y})$  is then

$$h(d_X, d) = \min(\alpha_{bp}, \rho_s \rho_{bp} |d_X - d|). \quad (12)$$

$h(d_X, d)$  depends on both  $\delta_{\mathbf{X}, \mathbf{Y}}$  and  $\delta_{mean}$ .  $h(d_X, d)$  steadily decreases with the increase of  $\delta_{\mathbf{X}, \mathbf{Y}}$ , because it is likely to be a color edge. A large  $\delta_{mean}$  means that the scene is heavy textured, and it is less likely to be a color edge even  $\delta_{\mathbf{X}, \mathbf{Y}}$  is large. Hence,  $h(d_X, d)$  steadily increases with the increase of  $\delta_{mean}$ .

The global energy is observed empirically to converge after a certain number of iterations. Finally, the label  $d$  that minimizes  $E_{\mathbf{X}}(d)$  individually at each pixel is selected.

A Standard loopy BP algorithm is too slow to be practically used to achieve nice results. Felzenszwalb and Huttenlocher [9] proposed a hierarchical algorithm that runs much faster than the previous algorithms while maintaining comparable accuracy. The main difference between hierarchical BP and standard BP is that hierarchical BP works in a coarse-to-fine manner. The basic steps are 1) initialize the messages at the coarsest level to all zeros, 2) apply loopy BP at the coarsest scale to iteratively refine the messages, and 3) use refined messages from the coarser level to initialize the messages for the next scale. In detail, assuming  $\mathbf{X}$  is one pixel in the coarsest scale and its corresponding pixels in the finer scale are  $\mathbf{X}'_i$ ,  $i \in [1, 4]$ , as is shown in Fig. 6, then

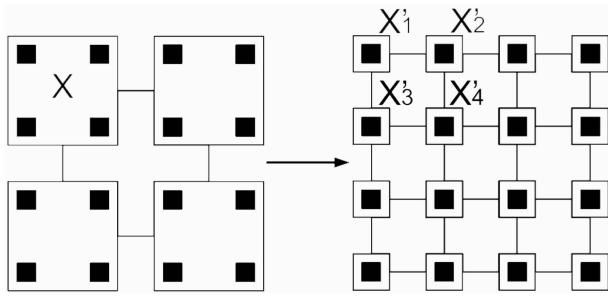


Fig. 6. Illustration of two levels in the coarse-to-fine method. Each node  $\mathbf{X}$  in the left figure corresponds to a block of four nodes  $\mathbf{X}'_i$  in the right figure.

$$E_{D,\mathbf{X}} = \sum_{i \in [1,4]} E_{D,\mathbf{X}'_i}, \quad (13)$$

$$M_{\mathbf{X}'_i, \mathbf{Y}_j} = M_{\mathbf{X}, \mathbf{Y}_j}, i, j \in [1, 4], \quad (14)$$

where  $\mathbf{Y}'_{i,j}$  are the four neighbors of pixel  $\mathbf{X}'_i$  and  $\mathbf{Y}_j$  are the corresponding four neighbors of pixel  $\mathbf{X}$ . For instance, assuming  $\mathbf{Y}'_{i,j}$  is the upper pixel of  $\mathbf{X}'_i$ , then  $\mathbf{Y}_j$  is also the upper pixel of  $\mathbf{X}$ . Hence, the number of the messages in the finer lever is four times larger than those in the coarser level. As a result, the messages will be hierarchically refined while the data term stays unchanged, since the data term for the coarser level is the sum of the corresponding four data terms of the finer level, as is shown in (14). Moreover, for simplicity,  $\rho_s$  is set to 1.0 for all the coarse levels, which means that the colors of the reference image are not used to define the jump cost. Finally, the refined messages and the data term are used to construct the total energy according to (8).

Two main parameters,  $s_{bp}$  and  $n_{bp}$ , define the behavior of this hierarchical BP algorithm,  $s_{bp}$  is the number of scales and  $n_{bp}$  is the number of iterations in each scale. In the paper, we experimentally set  $s_{bp} = 4$  and  $n_{bp} = 50$ ; a discussion of the parameter settings and their sensitivity is provided in Section 5.1.

### 3.5 Fast-Converging Belief Propagation

In a standard BP algorithm, in order to achieve high-quality stereo results, a large number of iterations is required to guarantee convergence. Therefore, the application of BP is mainly restricted to offline processing.

Several approximated BP algorithms have been developed recently. Chen [5] used K-L divergence to measure the similarity of the previous messages and decided whether to send a message or not by setting a threshold on it. Elidan et al. [8] maintained a priority queue of messages and updated the one with the highest residual. Ihler et al. [13] provided some theoretical analysis for the approximated sum-product BP algorithms. In this section, we propose a fast-converging BP based on the max-product BP employed in Section 3.4. In essence, by only updating the pixels that have not yet converged, our method effectively removes the redundant computation involved in standard BP.

In detail, according to (7), the new messages ( $M_{\mathbf{X}, \mathbf{Y}_j}^t, j \in [1, 4]$ ) delivered by a pixel  $\mathbf{X}$  to its neighbors  $\mathbf{Y}_j$  in iteration  $t$  are updated according to its own data term ( $E_{D,\mathbf{X}}$ ), the old messages ( $M_{\mathbf{Y}_j, \mathbf{X}}^{t-1}$ ) received from  $\mathbf{Y}_j$ , and the jump cost function  $h$ . Since  $E_{D,\mathbf{X}}$  and  $h$  stay unchanged for every

TABLE 1  
Comparison of the Performance of Standard BP and Fast-Converging BP of the Tsukuba Data Set

$\eta_z$	$\bar{\Delta E}$	$\bar{\mathcal{R}}$
1.0e-01	1.084e+02	3.120e-04
1.0e-04	0.361e-05	0.101e-10
1.0e-16	0	0

iteration,  $M_{\mathbf{X}, \mathbf{Y}_j}^t$  is updated according to  $M_{\mathbf{Y}_j, \mathbf{X}}^{t-1}$  only. For every pixel  $\mathbf{X}$ , a fast-converging BP algorithm will first check the similarity of the messages delivered from its four neighboring pixels at iteration  $t - 1$  and iteration  $t - 2$  and then update  $M_{\mathbf{X}, \mathbf{Y}_j}^t$  only if the L1-norm of their difference is larger than a threshold that is small enough:

$$\left| M_{\mathbf{Y}_j, \mathbf{X}}^{t-1} - M_{\mathbf{Y}_j, \mathbf{X}}^{t-2} \right|_1 \geq \eta_z, \quad (15)$$

where  $\eta_z$  is experimentally set to 0.1 in our paper. If the L1-norm value is smaller than  $\eta_z$ ,  $\mathbf{X}$  is declared converged. Note that a converged pixel in the current iteration  $t$  may be nonconverged in the next iteration  $t + 1$  since the messages of its four neighboring pixels may be updated during the current iteration, that is,  $M_{\mathbf{Y}_j, \mathbf{X}}^t$  may not be the same as  $M_{\mathbf{Y}_j, \mathbf{X}}^{t-1}$ .

We then demonstrate that the accuracy will be well-preserved with  $\eta_z \in [1.0e^{-16}, 1.0e^{-01}]$ . Let  $E_{sbp}(t)$  denote the energy of standard BP and  $E_{fbp}(t)$  denote the energy of fast-converging BP. Assume  $\Delta E(t) = |E_{sbp}(t) - E_{fbp}(t)|$  and  $\mathcal{R}(t) = \frac{\Delta E(t)}{E_{sbp}(t)}$  and let  $\bar{\Delta E}$  and  $\bar{\mathcal{R}}$  denote the average of  $\Delta E(t)$  and  $\mathcal{R}(t)$  over 50 iterations, respectively. Table 1 shows that  $\bar{\Delta E}$  and  $\bar{\mathcal{R}}$  decrease as the threshold  $\eta_z$  decreases and finally reach zero when  $\eta_z = 1.0e^{-16}$ . However, please note that smaller  $\eta_z$  implies more nonconverged pixels, as shown in Fig. 7a, and thereby increases the runtime. In our experiments, we set  $\eta_z = 0.1$ , which is observed to greatly improve the speed while well preserve the accuracy. Fig. 7b shows that after a number of iterations, most of the pixels on the graph converge. Fast-converging BP algorithm thus ignores these pixels, and the updating scheme is only applied to the nonconverged pixels, which greatly decreases the runtime of BP approaches with large number of iterations. Fig. 7c shows that unlike standard BP, the runtime of fast-converging BP is sublinear to the number of iterations.

## 4 DEPTH ENHANCEMENT

To reduce the discontinuities caused by the quantization in the depth hypothesis selection process, a subpixel estimation algorithm is proposed based on quadratic polynomial interpolation. In this paper, quadratic polynomial interpolation is used to approximate the cost function between three discrete depth candidates:  $d$ ,  $d_-$ , and  $d_+$ .  $d$  is the discrete depth with the minimal cost,  $d_- = d - 1$ , and  $d_+ = d + 1$ . The cost function is approximated as

$$f(x) = ax^2 + bx + c. \quad (16)$$

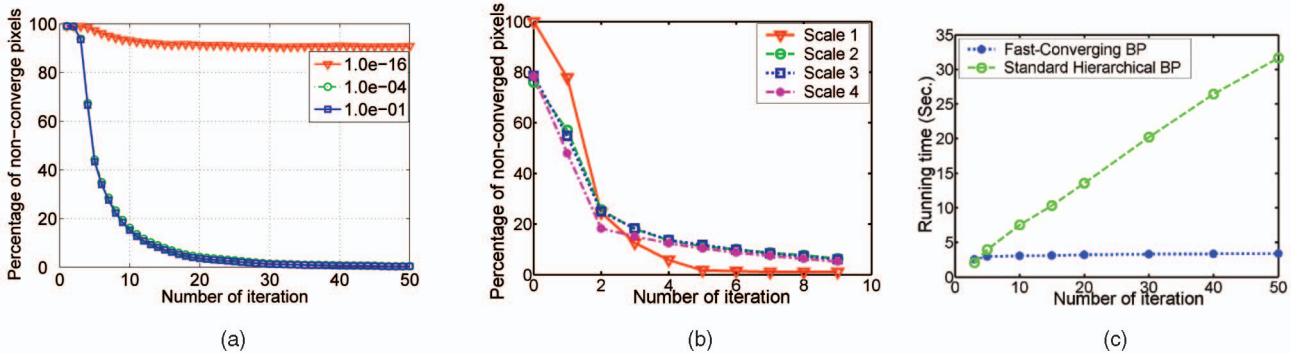


Fig. 7. (a) The percentage of the nonconverged pixels after every iteration for three  $\eta_z$  values:  $1.0e^{-16}$ ,  $1.0e^{-04}$ , and  $1.0e^{-01}$ . By setting  $\eta_z = 0.1$ , (b) provides the percentage of the nonconverged pixels after every iteration for different scales and (c) compares the runtime of fast-converging BP and standard hierarchical BP algorithms. Both algorithms are run on the Tsukuba data set with the same number of iterations and scales.

Hence, given  $d$ ,  $f(d)$ ,  $f(d_-)$ , and  $f(d_+)$ , the depth with the minimum of the quadric cost function  $f(x)$  can be computed:

$$x_{min} = d - \frac{f(d_+) - f(d_-)}{2(f(d_+) + f(d_-) - 2f(d))}. \quad (17)$$

At the end of our integer-based stereo pipeline, the disparity map  $D_L^{(5)}$  is produced, which means that the three depth candidates  $d$ ,  $d_-$ , and  $d_+$  are known immediately.  $d$  is extracted from the input depth map,  $d_- = d - 1$  and  $d_+ = d + 1$ . To perform depth enhancement with two views, three slices of the matching cost  $f(d)$ ,  $f(d_-)$ , and  $f(d_+)$  of the three depth candidates are selected from the color-weighted correlation volume, which is denoted by  $C_L$  in Figs. 1 and 2. The subpixel estimation is then carried out by using (17) to calculate the subpixel disparity map. Finally, we replace each value with the average of those values that are within one disparity over a  $9 \times 9$  window.

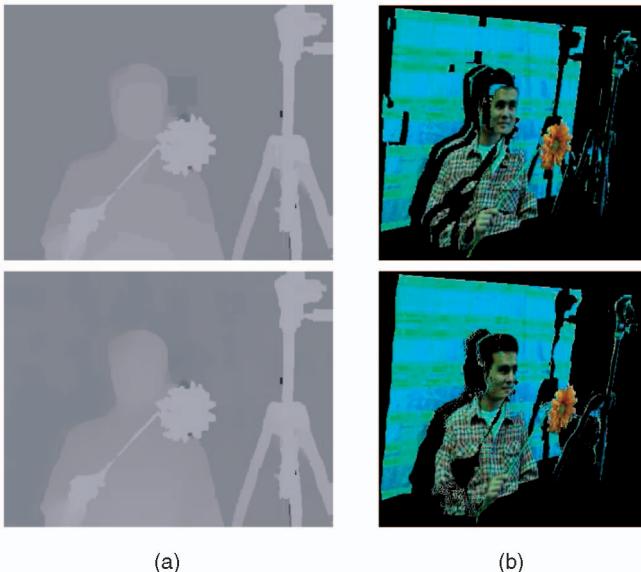


Fig. 8. (a) Disparity maps. (b) Synthesized views using (a). The first row shows the results without subpixel refinement and the second row shows the results with subpixel refinement. Notice that the quantization effect on the man's face and the background on the synthesized view before subpixel is removed after subpixel estimation.

Fig. 8 provides a visual comparison of the disparity maps and their synthesized views before and after subpixel estimation. Notice that the quantization effect on the man's face and the background on the synthesized view is removed after subpixel estimation.

## 5 EXPERIMENTS

### 5.1 Parameter Settings

In this section, we provide all of the parameter settings used in our algorithm. The same parameter settings were used throughout.

The parameters are shown in Table 2 and separated into four parts: three parameters ( $\alpha_{ms}, \beta_{ms}, \gamma_{ms}$ ) for the mean shift segmentation, three parameters ( $\alpha_{cw}, \beta_{cw}, \gamma_{cw}$ ) for the color-weighted filter, six parameters ( $\alpha_{bp}, \eta_{bp}, \rho_{bp}, \lambda_{bp}, s_{bp}, n_{bp}$ ) for the hierarchical BP, and six parameters ( $\kappa_s, \kappa_u, \kappa_o, \alpha_s, \eta_s, n_s$ ) for the iterative refinement. In the next few paragraphs, we explain these parameters and evaluate their effect on the stereo results.

For mean shift color segmentation,  $\alpha_{ms}$  is the spatial bandwidth,  $\beta_{ms}$  is the color bandwidth, and  $\gamma_{ms}$  is the minimum region size. They are set as the default values. Fig. 9 shows the performance of the proposed method according to the mean shift segmentation parameters. Figs. 9a and 9b set  $\gamma_{ms}$  to constants 20 and 50, respectively, and show the behavior of the proposed method for  $\alpha_{ms}$  and  $\beta_{ms}$ , both range from 4 to 10. The proposed method is very robust against  $\alpha_{ms}$  and  $\beta_{ms}$ , especially within the range of 6 and 10. Visually

TABLE 2  
Parameter Settings Used Throughout

Mean Shift Segmentation	$\alpha_{ms}$	$\beta_{ms}$	$\gamma_{ms}$			
	7	6	20			
Color-Weigh. Correlation	$\alpha_{cw}$	$\beta_{cw}$	$\gamma_{cw}$			
	33	10	21			
Hierarchical BP	$\lambda_{bp}$	$\eta_{bp}$	$\rho_{bp}$	$\alpha_{bp}$	$s_{bp}$	$n_{bp}$
	0.2	$2\bar{c}$	1.0	$n_d/8$	4	50
Iterative Refinement	$\kappa_s$	$\kappa_u$	$\kappa_o$	$\alpha_s$	$\eta_s$	$n_s$
	0.05	0.5	2	0.04	0.7	5

$n_d$  is the number of disparity levels.  $\bar{c}$  is the average of the values in the correlation volume.

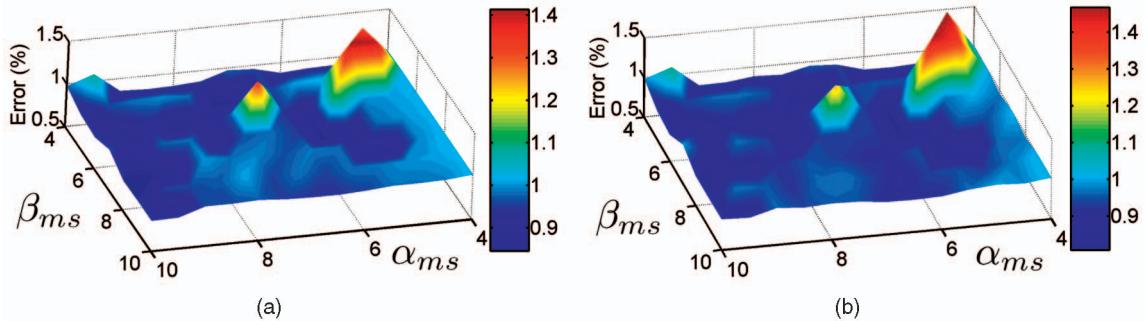


Fig. 9. Performance according to the mean shift parameters ( $\alpha_{ms}$  is spatial bandwidth,  $\beta_{ms}$  is color bandwidth, and  $\gamma_{ms}$  is the minimum region size) for Tsukuba data set. (a) The behavior of the algorithm according to the parameters  $\alpha_{ms}$  and  $\beta_{ms}$  by setting  $\gamma_{ms} = 20$  and, in (b),  $\gamma_{ms} = 50$ . Apparently, the performance is fairly constant with respect to the parameter  $\gamma_{ms}$ . (a)  $\gamma_{ms} = 20$ . (b)  $\gamma_{ms} = 50$ .

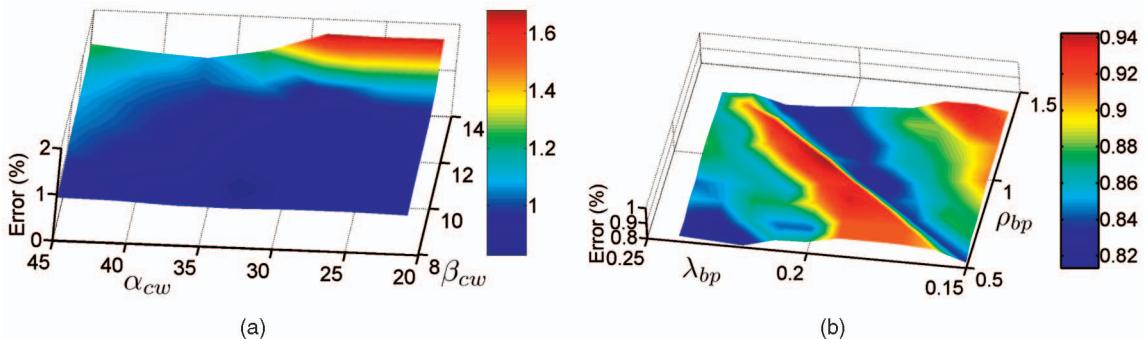


Fig. 10. Performance according to the color-weighted filter and the hierarchical BP parameters for the Tsukuba data set.  $\alpha_{cw}$  is the size of the support window,  $\beta_{cw}$  is defined in (2),  $\rho_{bp}$  is the rate of increase in the jump cost, and  $\lambda_{bp}$  is the scaling factor applied to the correlation volume after the truncation. (a) Color-weighted filter. (b) Hierarchical BP.

comparing Figs. 9a and 9b also shows that the method is robust against  $\gamma_{ms}$ . The runtime of the mean shift segmentation is about 20 seconds for the Tsukuba data set.

For color-weighted filter,  $\alpha_{cw}$  is the size of the support window, and  $\beta_{cw}$  and  $\gamma_{cw}$  are defined in (2). Fig. 10a shows the performance of the proposed method according to  $\alpha_{cw}$  and  $\beta_{cw}$  by keeping  $\gamma_{cw}$  constant. We can see that the proposed method is fairly robust against different sizes of a support window ( $\alpha_{cw}$ ). This is because the effect of outliers (i.e., pixels from different depths) does not increase in the proposed method even though the size of a support window increases. Our method also appears to be fairly robust against different values of  $\beta_{cw}$ , the error rate is almost constant for  $\beta_{cw}$  between 8 and 12. The runtime of the color-weighted filtering depends mostly on the parameter  $\alpha_{cw}$ . When  $\alpha_{cw} = 33$ , the runtime is about 30 seconds for the Tsukuba data set.

For hierarchical BP,  $\alpha_{bp}$  and  $\eta_{bp}$  are truncations of the jump cost and correlation volume, respectively. The parameter  $\rho_{bp}$  is the rate of increase in the jump cost and  $\lambda_{bp}$  is a scaling factor applied to the correlation volume after the truncation. The parameter  $s_{bp}$  is the number of scales, and  $n_{bp}$  is the number of iterations, as defined in Section 3.1. Parameters  $\rho_{bp}$ ,  $\lambda_{bp}$ ,  $s_{bp}$ , and  $n_{bp}$  are set experimentally. According to our experiments, we found that  $\rho_{bp}$  and  $\lambda_{bp}$  are the most sensitive ones and their performances are studied in Fig. 10b. Our method is very robust when  $\rho_{bp} \in [0.5, 1.5]$  and  $\lambda_{bp} \in [0.15, 0.25]$ . The runtime of the hierarchical BP is about 2 seconds for the Tsukuba data set.

Parameters  $\kappa_s$ ,  $\kappa_u$ , and  $\kappa_o$  for the iterative refinement are defined in Section 3.3. Among them,  $\kappa_s$  is less important

because it is the weight for the stable pixels. The performance of our method according to  $\kappa_u$  and  $\kappa_o$  are provided in Fig. 11a, which shows that our method is pretty robust over nontrivial ranges.  $\alpha_s$  is the threshold on correlation confidence defined in Section 3.2. Parameter  $\eta_s$  is related to the plane fitting process, as defined in Section 3.3. Fig. 11b evaluates the performance of our method according to  $\alpha_s$  and  $\eta_s$ , which shows that the performance does not depend much on  $\eta_s$  unless it is very small. The smaller  $\eta_s$  is, the fewer number of the color segments will the iterative refinement apply to. The performance according to  $\alpha_s$  is robust within the range of 0.02 and 0.12. However, the larger the  $\alpha_s$ , the fewer the stable pixels there are in each segment, which makes the RANSAC plane fitting less robust; also, the smaller the  $\alpha_s$ , the more unstable pixels will be classified to stable pixels, which also decreases the robustness of the RANSAC plane fitting. The final parameter  $n_s$  is the number of iterations for the iterative refinement process, Fig. 13 shows that it always converge after five iterations. The runtime of the refinement is about 10 seconds per iteration for the Tsukuba data set.

## 5.2 Experimental Results

We evaluate our algorithm on the Middlebury data sets and we show in Tables 3 and 4 that our algorithm on the average outperforms all the other algorithms listed on the Middlebury homepage with both error threshold 1 and error threshold 0.5. The result on each data set is computed by measuring the percentage of the pixels with an incorrect disparity estimate. This measure is computed for three subsets of the image.

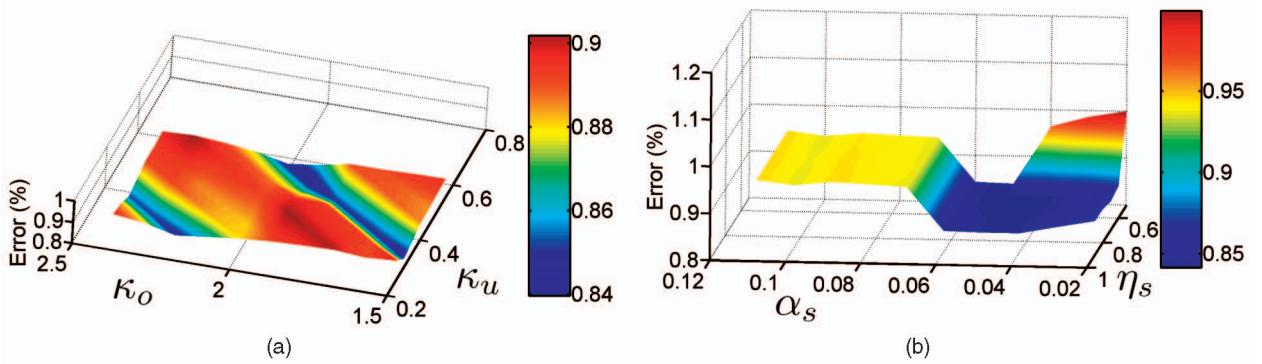


Fig. 11. Performance according to the iterative refinement parameters for the Tsukuba data set.  $\kappa_u$  is the weight for unstable pixels,  $\kappa_o$  is the weight for occluded pixels,  $\alpha_s$  is the threshold on correlation confidence defined in Section 3.2, and  $\eta_s$  is the threshold on the ratio of the stable pixels to all pixels.

TABLE 3  
Comparison of the Results on the Middlebury Data Sets with Error Threshold 1

Algorithm	Avg. Rank	Tsukuba			Venus			Teddy			Cones		
		nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc
<b>Our Algorithm</b>	<b>2.1</b>	<b>0.88<sub>1</sub></b>	<b>1.29<sub>1</sub></b>	<b>4.76<sub>1</sub></b>	0.13 <sub>2</sub>	0.45 <sub>3</sub>	1.87 <sub>2</sub>	<b>3.53<sub>1</sub></b>	8.30 <sub>3</sub>	<b>9.63<sub>1</sub></b>	2.90 <sub>3</sub>	8.78 <sub>5</sub>	7.79 <sub>2</sub>
AdaptingBP [15]	2.2	1.11 <sub>4</sub>	1.37 <sub>3</sub>	5.79 <sub>4</sub>	<b>0.10<sub>1</sub></b>	<b>0.21<sub>1</sub></b>	<b>1.44<sub>1</sub></b>	4.22 <sub>3</sub>	7.06 <sub>2</sub>	11.8 <sub>3</sub>	<b>2.48<sub>1</sub></b>	7.92 <sub>2</sub>	<b>7.32<sub>1</sub></b>
DoubleBP [26]	3.4	0.88 <sub>2</sub>	1.29 <sub>2</sub>	4.76 <sub>2</sub>	0.14 <sub>3</sub>	0.6 <sub>7</sub>	2.00 <sub>4</sub>	3.55 <sub>2</sub>	8.71 <sub>4</sub>	9.70 <sub>2</sub>	2.90 <sub>4</sub>	9.24 <sub>6</sub>	7.80 <sub>3</sub>
SymBP+occ [20]	6.3	0.97 <sub>3</sub>	1.75 <sub>6</sub>	5.09 <sub>3</sub>	0.16 <sub>4</sub>	0.33 <sub>2</sub>	2.19 <sub>5</sub>	6.47 <sub>7</sub>	10.7 <sub>5</sub>	17.0 <sub>8</sub>	4.79 <sub>13</sub>	10.7 <sub>10</sub>	10.9 <sub>9</sub>
Segm+visib [2]	6.4	1.30 <sub>7</sub>	1.57 <sub>4</sub>	6.92 <sub>9</sub>	0.79 <sub>10</sub>	1.06 <sub>8</sub>	6.76 <sub>11</sub>	5.00 <sub>4</sub>	<b>6.54<sub>1</sub></b>	12.3 <sub>4</sub>	3.72 <sub>7</sub>	8.62 <sub>4</sub>	10.2 <sub>8</sub>
C-SemiGlob [12]	7.3	2.61 <sub>18</sub>	3.29 <sub>13</sub>	9.89 <sub>15</sub>	0.25 <sub>6</sub>	0.57 <sub>4</sub>	3.24 <sub>7</sub>	5.14 <sub>5</sub>	11.8 <sub>6</sub>	13.0 <sub>5</sub>	2.77 <sub>2</sub>	8.35 <sub>3</sub>	8.20 <sub>4</sub>

The numbers in the last 12 columns are the percentage of the pixels with incorrect disparities on the different subsets of the images. Our algorithm is ranked in first place on average.

TABLE 4  
Comparison of Results on the Middlebury Data Sets with Error Threshold 0.5

Algorithm	Avg. Rank	Tsukuba			Venus			Teddy			Cones		
		nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc
<b>Our Algorithm</b>	<b>3.0</b>	8.78 <sub>5</sub>	9.45 <sub>3</sub>	14.9 <sub>2</sub>	<b>0.72<sub>1</sub></b>	<b>1.12<sub>1</sub></b>	<b>5.24<sub>1</sub></b>	10.1 <sub>2</sub>	16.4 <sub>2</sub>	<b>21.3<sub>1</sub></b>	8.49 <sub>7</sub>	14.7 <sub>7</sub>	16.5 <sub>4</sub>
C-SemiGlob [12]	4.5	13.9 <sub>10</sub>	14.7 <sub>10</sub>	18.9 <sub>13</sub>	3.30 <sub>3</sub>	3.82 <sub>2</sub>	10.9 <sub>5</sub>	<b>9.82<sub>1</sub></b>	17.4 <sub>4</sub>	22.8 <sub>2</sub>	5.37 <sub>2</sub>	<b>11.7<sub>1</sub></b>	<b>12.8<sub>1</sub></b>
SemiGlob [11]	6.0	13.4 <sub>9</sub>	14.3 <sub>9</sub>	20.3 <sub>15</sub>	4.55 <sub>5</sub>	5.38 <sub>6</sub>	15.7 <sub>9</sub>	11.0 <sub>4</sub>	18.5 <sub>5</sub>	26.1 <sub>5</sub>	<b>4.93<sub>1</sub></b>	12.5 <sub>2</sub>	13.5 <sub>2</sub>
AdaptingBP[15]	6.3	19.1 <sub>15</sub>	19.3 <sub>13</sub>	17.4 <sub>6</sub>	4.84 <sub>7</sub>	5.08 <sub>5</sub>	7.84 <sub>2</sub>	12.8 <sub>6</sub>	16.7 <sub>3</sub>	26.3 <sub>6</sub>	7.02 <sub>4</sub>	13.2 <sub>5</sub>	14.0 <sub>3</sub>
Segm+visib[2]	6.8	12.7 <sub>7</sub>	12.9 <sub>6</sub>	15.8 <sub>5</sub>	10.4 <sub>14</sub>	11.0 <sub>14</sub>	19.5 <sub>13</sub>	11.0 <sub>3</sub>	<b>13.2<sub>1</sub></b>	23.7 <sub>3</sub>	8.12 <sub>6</sub>	13.1 <sub>4</sub>	17.3 <sub>6</sub>
DoubleBP[26]	8.8	18.7 <sub>13</sub>	19.1 <sub>12</sub>	15.8 <sub>4</sub>	7.85 <sub>11</sub>	8.38 <sub>10</sub>	11.6 <sub>6</sub>	14.3 <sub>7</sub>	19.9 <sub>6</sub>	24.3 <sub>4</sub>	11.9 <sub>11</sub>	18.1 <sub>10</sub>	19.9 <sub>11</sub>

The numbers in the last 12 columns are the percentage of pixels with incorrect disparities on different subsets of the images. The average ranks show that our algorithm is state-of-the-art. The top performer for the Tsukuba data set is not shown in the table because its average rank is much lower than the ones provided.

- The subset of the nonoccluded pixels, denoted as “nonocc.”
- The subset of the pixels near the occluded areas, denoted as “disc.”
- The subset of the pixels being either nonoccluded or half-occluded, denoted as “all.”

The ranks show that our algorithm works pretty well in the nonoccluded and discontinuous areas. In most data sets, we take first or second place, which demonstrates that our algorithm is robust to different inputs. Also note that our algorithm has a better performance in the occluded areas than in [26]. This is because the correlation volume computed from the pixels in the border occlusions is not aggregated. Note that only partial disparity range can be evaluated since the matching pixels may be outside the

image. This trivial scheme better predicts the disparity values around the border occlusions while preserving the accuracy of the other areas.

In Fig. 12, the results after different intermediate stages are shown. This provides a visual explanation of how the different stages in the pipeline improve the results. For comparison, we also give the ground truth. The scores for the intermediate results are given in Table 5 along with  $D_L^{(5)}$  SPECIAL. It is the same as  $D_L^{(5)}$  except that we do not use the colors of the reference image to define the jump cost, which has a strong impact on the Teddy and Cones data sets.

In Figs. 13 and 14, we show how an increased number of iterations in estimating the data term  $E_D$  improves the result. The result with zero iteration is  $D_L^{(0)}$ , which is the initial disparity map. Based on this, we chose to use five iterations in our method.

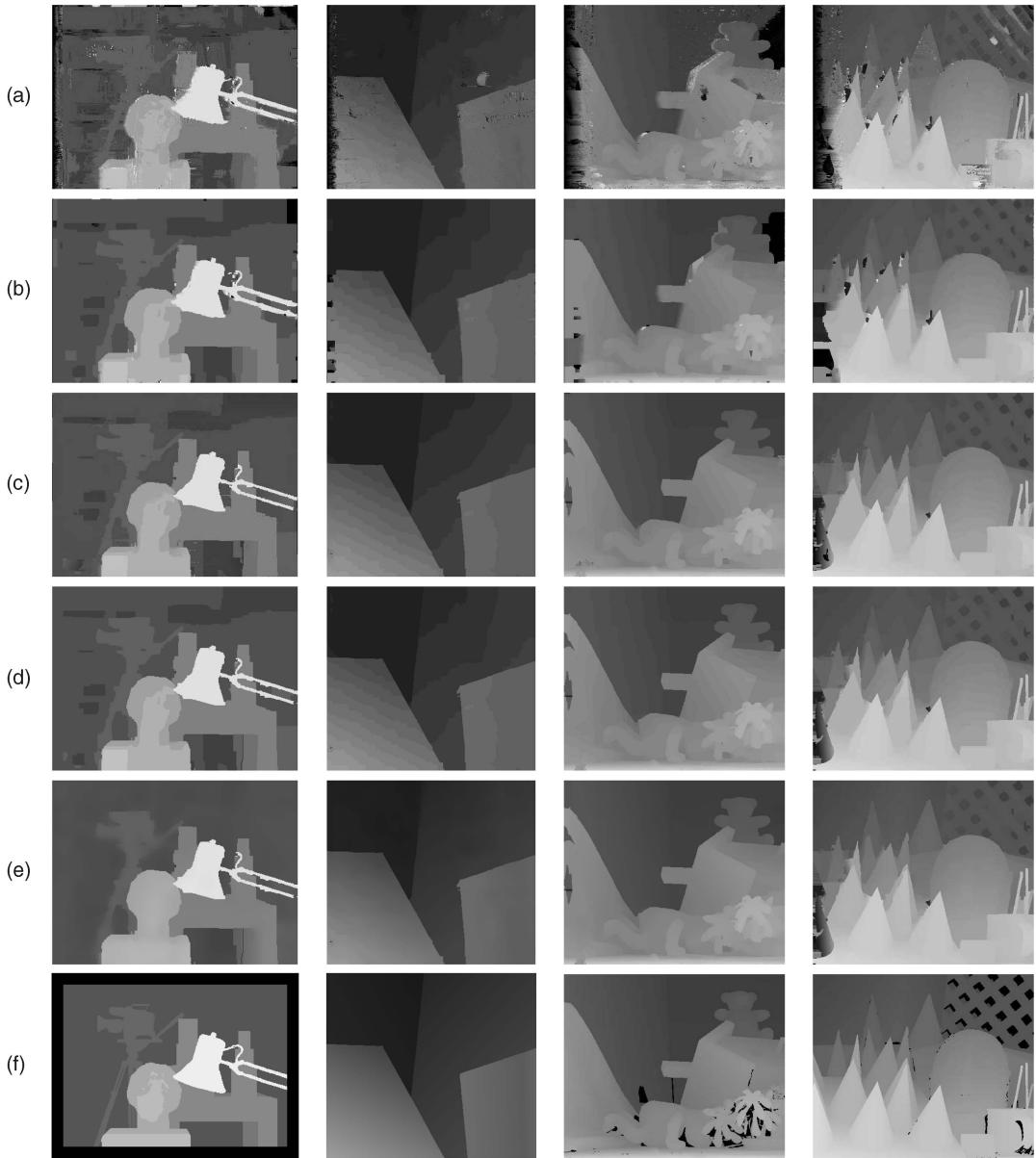


Fig. 12. Intermediate results from our algorithm for the four different standard test sets compared to the ground truth. (a) The disparity maps after applying the winner-takes-all approach to the color-weighted correlation volume  $C_L$ . (b) The output of the initial hierarchical BP. This result is denoted by  $D_L^{(0)}$  in Figs. 1, 2, and 3. (c) The results after fitting planes to the regions from the color segmentation. These are denoted by  $D_{pf}^{(i)}$  in Fig. 3. (d) The integer-based disparity maps from our stereo pipeline. These results are denoted by  $D_L^{(i+1)}$  in Fig. 3. (e) The disparity maps after depth enhancement. (f) The ground truth for visual comparison.

TABLE 5  
The First Four Rows in the Table Correspond to the First Four Rows in Fig. 12

Algorithm	Tsukuba			Venus			Teddy			Cones		
	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc
$D_{cw}$	2.70	4.74	7.37	3.59	5.21	12.9	14.6	23.4	24.0	12.5	22.3	18.9
$D_L^{(0)}$	1.21	3.28	5.95	0.68	1.96	8.03	7.83	15.5	15.5	4.25	12.7	10.4
$D_{pf}^{(4)}$	2.60	2.98	7.31	0.13	0.47	1.85	3.82	8.52	10.1	2.87	8.41	7.91
$D_L^{(5)}$	0.88	1.29	4.76	0.13	0.45	1.87	3.53	8.30	9.63	2.90	8.78	7.79
$D_L^{(5)}$ SPECIAL	0.88	1.30	4.77	0.14	0.48	1.95	3.71	8.87	10.3	3.07	8.81	8.17

The last row is the same as the fourth row ( $D_L^{(5)}$ ) except that we do not use the colors of the reference image to define the jump cost.

We also test our algorithm with some outdoor scenes, which are probably more challenging. Some experimental

results are provided in Fig. 15. As no ground truth is available, we used the disparity map to synthesize a novel

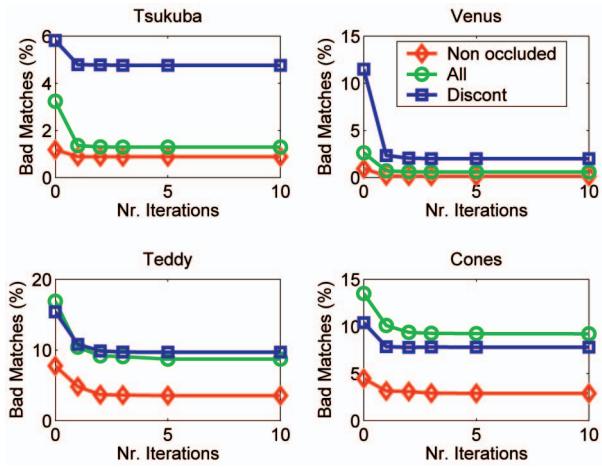


Fig. 13. The iterated computations of  $E_D^{(i+1)}$  improve the result. In most cases, one iteration is enough for convergence. After five iterations, the algorithm has always converged.

right image and compare with the real one. The experimental results show that our algorithm works well with outdoor scenes, especially for color images.

To achieve subpixel accuracy, we proposed a depth-enhanced approach for postprocessing purpose. The experimental results on the Middlebury data sets are provided on Fig. 12e. A set of synthesized views using the disparity maps in Figs. 12d and 12e are shown in Fig. 16, providing a visual comparison of the algorithms with and without depth enhancement. The improvement is obvious. The results shown in Fig. 16a are quantized to discrete number of planes. After subpixel estimation, the quantization effect is removed, as it is shown in Fig. 16b. The average ranks are provided in Table 4, showing that our algorithm is currently state-of-the-art with subpixel accuracy.

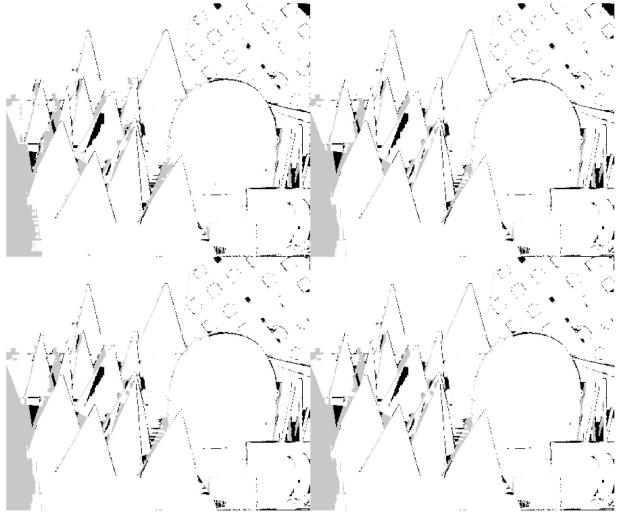


Fig. 14. Pixels with incorrect disparity for the "Cones" data set. In the first row, the results after one and two iterations are shown and in the second row, the results after three and five iterations are shown. The gray and black pixels are bad pixels with absolute disparity error larger than 1 and the gray pixel are half-occluded.

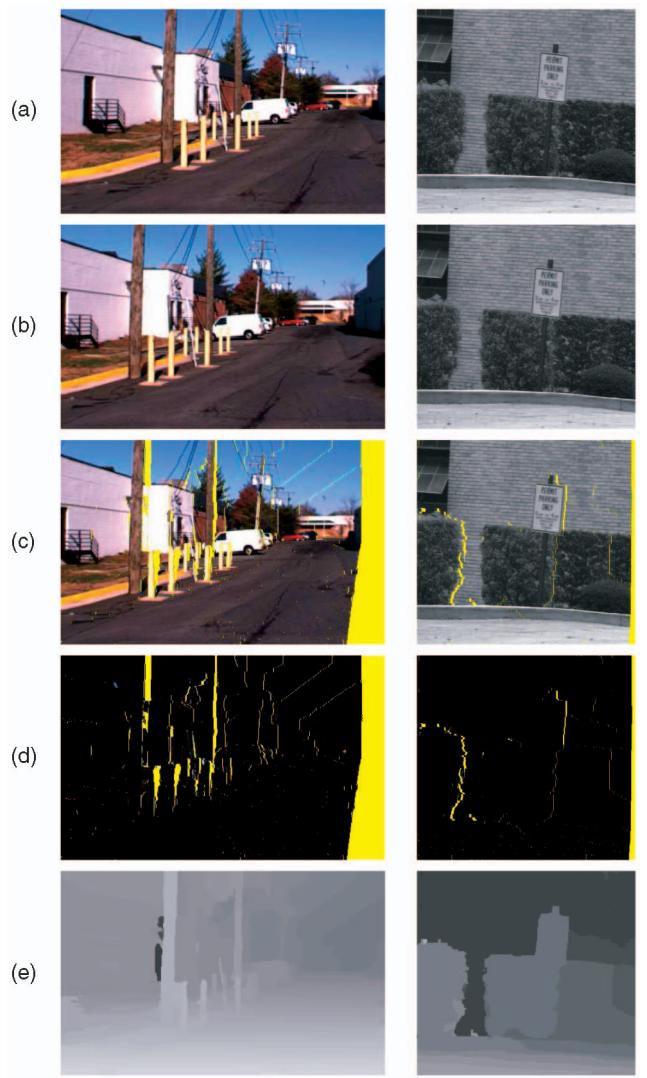


Fig. 15. (a) and (b) The left and right camera images. (c) The synthesized right image. (d) The absolute difference of (b) and (c). (e) Corresponding disparity maps.

## 6 CONCLUSIONS

In this paper, a stereo model based on energy minimization, color segmentation, plane fitting, repeated application of hierarchical BP, and depth enhancement was proposed. Typically, one application of the hierarchical BP brings the error down close to its final value so that the algorithm could perhaps be used as a two-step approach, where occlusions and untextured areas are first detected and then filled in from neighboring areas. In addition, a fast converging BP approach is proposed, which preserves the same accuracy as the standard BP. The runtime of fast-converging BP is sublinear to the number of iterations.

Our algorithm is currently outperforming the other algorithms on the Middlebury data sets on average, but there is space left for improvement. For instance, in our algorithm, we only refined the disparity map for the reference image, but [19] suggests that, by generating a good disparity map for the right image, the occlusion constraints can be extracted more accurately. Our algorithm performs well when the scene is mainly composed of planar

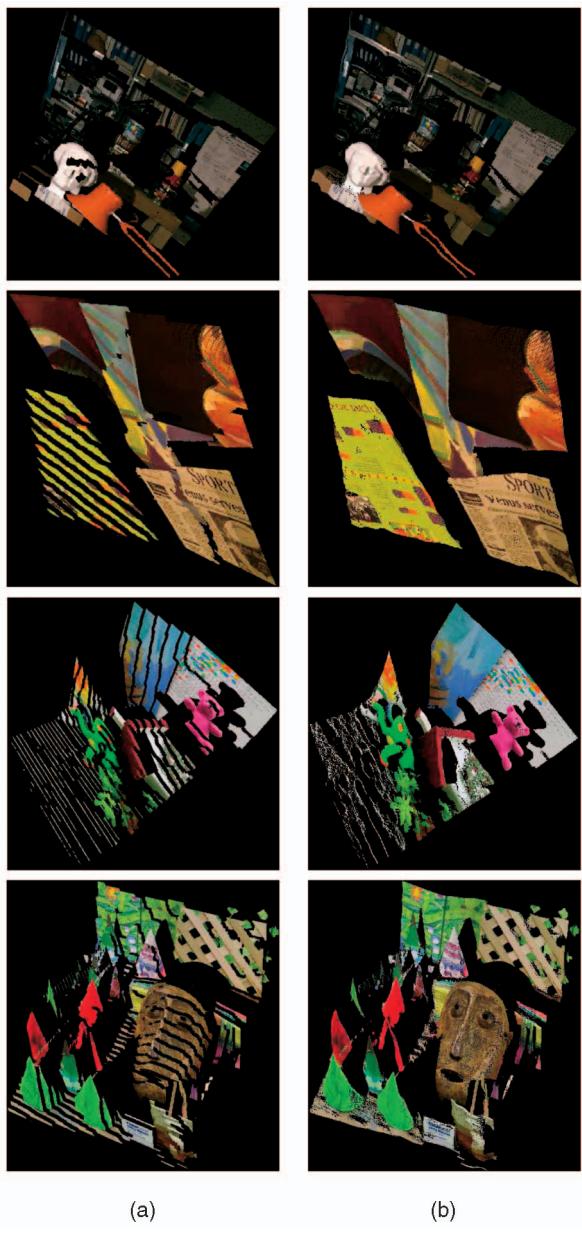
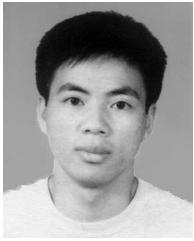


Fig. 16. (a) Synthesized views without depth enhancement. (b) Synthesized views after depth enhancement. The results shown in (a) are quantized to discrete number of planes. (b) After subpixel estimation, the quantization effect is removed.

surfaces because the depth information of the unstable areas is propagated from the stable pixels around the neighborhood by fitting a 3D plane. Thus, if the scene is mainly composed of smooth curved (e.g., quadric) surfaces, the performance of the propose algorithm may drop.

## REFERENCES

- [1] S. Birchfield and C. Tomasi, "A Pixel Dissimilarity Measure That Is Insensitive to Image Sampling," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, pp. 401-406, 1998.
- [2] M. Bleyer and M. Gelaut, "A Layered Stereo Algorithm Using Image Segmentation and Global Visibility Constraints," *Proc. IEEE Int'l Conf. Image Processing*, pp. 2997-3000, 2004.
- [3] Y. Boykov, O. Veksler, and R. Zabih, "A Variable Window Approach to Early Vision," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, pp. 1283-1294, 1998.
- [4] Y. Boykov, O. Veksler, and R. Zabih, "Fast Approximate Energy Minimization via Graph Cuts," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 23, pp. 1222-1239, 2001.
- [5] L. Chen, M. Cetin, and A.S. Willsky, "Distributed Data Association for Multi-Target Tracking in Sensor Networks," *Proc. Int'l Conf. Information Fusion*, 2005.
- [6] D. Comaniciu and P. Meer, "Mean Shift: A Robust Approach Toward Feature Space Analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, pp. 603-619, 2002.
- [7] G. Egnal and R. Wildes, "Detecting Binocular Half-Occlusions: Empirical Comparisons of Five Approaches," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, pp. 1127-1133, 2002.
- [8] G. Elidan, I. McGraw, and D. Koller, "Residual Belief Propagation: Informed Scheduling for Asynchronous Message Passing," *Proc. 22nd Conf. Uncertainty in Artificial Intelligence*, 2006.
- [9] P.F. Felzenszwalb and D.P. Huttenlocher, "Efficient Belief Propagation for Early Vision," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 261-268, 2004.
- [10] M.A. Fischler and R.C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Comm. ACM*, vol. 24, pp. 381-395, 1981.
- [11] H. Hirschmüller, "Accurate and Efficient Stereo Processing by Semi-Global Matching and Mutual Information," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, vol. 2, pp. 807-814, 2005.
- [12] H. Hirschmüller, "Stereo Vision in Structured Environments by Consistent Semi-Global Matching," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, vol. 2, pp. 2386-2393, 2006.
- [13] A.T. Ihler, J.W. Fisher, and A.S. Willsky, "Loopy Belief Propagation: Convergence and Effects of Message Errors," *J. Machine Learning Research*, vol. 6, pp. 905-936, 2005.
- [14] T. Kanade and M. Okutomi, "A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiments," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 16, pp. 920-932, 1994.
- [15] A. Klaus, M. Sormann, and K.F. Karner, "Segment-Based Stereo Matching Using Belief Propagation and a Self-Adapting Dissimilarity Measure," *Proc. Int'l Conf. Pattern Recognition*, vol. 3, pp. 15-18, 2006.
- [16] V. Kolmogorov and R. Zabih, "Computing Visual Correspondence with Occlusions Using Graph Cuts," *Proc. IEEE Int'l Conf. Computer Vision*, vol. 1, pp. 508-515, 2001.
- [17] D. Scharstein and R. Szeliski, *Middlebury Stereo Vision Research Page*, <http://vision.middlebury.edu/stereo/eval/>, 2008.
- [18] D. Scharstein and R. Szeliski, "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms," *Int'l Computer Vision*, vol. 47, pp. 7-42, 2003.
- [19] J. Sun, Y. Li, S.B. Kang, and H.-Y. Shum, "Symmetric Stereo Matching for Occlusion Handling," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, vol. 2, pp. 399-406, 2005.
- [20] J. Sun, N.-N. Zheng, and H.-Y. Shum, "Stereo Matching Using Belief Propagation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, pp. 787-800, 2003.
- [21] H. Tao and H. Sawhney, "Global Matching Criterion and Color Segmentation Based Stereo," *Proc. IEEE Workshop Applications of Computer Vision*, pp. 246-253, 2000.
- [22] M. Tappen and W. Freeman, "Comparison of Graph Cuts with Belief Propagation for Stereo," *Proc. IEEE Int'l Conf. Computer Vision*, vol. 1, pp. 508-515, 2003.
- [23] O. Veksler, "Stereo Correspondence with Compact Windows via Minimum Ratio Cycle," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, pp. 1654-1660, 2002.
- [24] O. Veksler, "Fast Variable Window for Stereo Correspondence Using Integral Image," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 556-561, 2003.
- [25] Y. Weiss and W. Freeman, "On the Optimality of Solutions of the Max-Product Belief Propagation Algorithm in Arbitrary Graphs," *IEEE Trans. Information Theory*, vol. 2, pp. 732-735, 2001.
- [26] Q. Yang, L. Wang, R. Yang, H. Stewénius, and D. Nistér, "Stereo Matching with Color-Weighted Correlation, Hierarchical Belief Propagation and Occlusion Handling," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition*, vol. 2, pp. 2347-2354, 2006.
- [27] K.-J. Yoon and I.-S. Kweon, "Adaptive Support-Weight Approach for Correspondence Search," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 28, pp. 650-656, 2006.

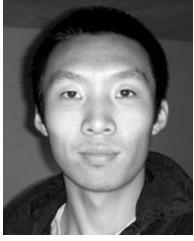


**Qingxiong Yang** received the BE degree from the University of Science and Technology of China (USTC) in 2004 and the MSc degree in computer science from the University of Kentucky in 2007. He is currently a PhD student in the Department of Electrical and Computer Engineering at the University of Illinois at Urbana-Champaign. His research interests reside in computer vision and graphics. He is a student member of the IEEE.



member of the IEEE.

**Henrik Stewénus** received the MSc degree in engineering physics in 2001, the Licentiate degree in mathematics in 2003, and the PhD degree in applied mathematics in 2005, with the thesis "Gröbner Basis Methods for Minimal Problems in Computer Vision," from Lund University, Sweden. He then spent two years working on geometry and scalable recognition at the University of Kentucky. He is currently working at Google Switzerland, Zurich. He is a member of the IEEE.



**Liang Wang** received the BS degree from the School of Computer Science, Beijing University of Aeronautics and Astronautics, in 2004. He is currently a PhD student in the Computer Science Department at the University of Kentucky. His research interests lie in computer vision, especially in 3D reconstruction and stereo matching. He is a student member of the IEEE.



**David Nistér** received the MSc degree in computer science and engineering and the Licentiate of engineering degree from the Chalmers University of Technology, Gothenburg, Sweden, in 1997 and 1998, respectively, and the PhD degree in computer vision, numerical analysis, and computing science from the Royal Institute of Technology (KTH), Stockholm, Sweden, in 2001, with the thesis "Automatic Dense Reconstruction from Uncalibrated Video Sequences." He is currently with Microsoft Corp. He was previously an assistant professor in the Computer Science Department and the Center for Visualization and Virtual Environments at the University of Kentucky and a researcher in the Vision Technologies Laboratory at Sarnoff Corp., Princeton, New Jersey. He is a member of the IEEE.



**Ruigang Yang** received the MS degree in computer science from Columbia University in 1998 and the PhD degree in computer science from the University of North Carolina, Chapel Hill, in 2003. He is an assistant professor in the Computer Science Department at the University of Kentucky. His research interests include computer graphics, computer vision, and multimedia. He is a recipient of the US National Science Foundation CAREER award in 2004 and a member of the IEEE, the IEEE Computer Society, and the ACM.

▷ For more information on this or any other computing topic, please visit our Digital Library at [www.computer.org/publications/dlib](http://www.computer.org/publications/dlib).