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

Qingxiong Yang, *Student Member, IEEE,* Liang Wang, *Student Member, IEEE,* Ruigang 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.**

I. 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 [19], [20]. 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 belief propagation (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 sub-pixel 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, method suitable for parallel execution are 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 running time of fast-converging BP is sub-linear to the number of iterations.

The paper is organized as follows: Section II gives a high-level overview of the approach. In

Section III we present the detailed equations for all the building blocks. Additionally, a depth-enhanced approach is presented in Section IV. Section V reports results showing that our algorithm is currently the strongest available on the Middlebury data set. Section VI concludes.

II. Overview of the Approach

The algorithm can be partitioned into three blocks, initialization (Figure 1), pixel classification (Figure 2) and iterative refinement (Figure 3). In the initialization step (see Figure 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 pixels in the left and right image, but there are many other methods for correlation volume construction. For instance, Sun et al. [19] 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 belief propagation is performed in a manner similar to Felzenszwalb [9], resulting in the initial left and right disparity maps $D_L^{(0)}$ and D_R, 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, and the initial data term $E_D^{(0)}$.

In the pixel classification module (see Figure 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. 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 Figure 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 [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 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 belief propagation 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 set.

III. 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 Figures 1,2 and 3. A detailed explanation of the hierarchical belief propagation algorithm is provided afterwards. To further improve the speed

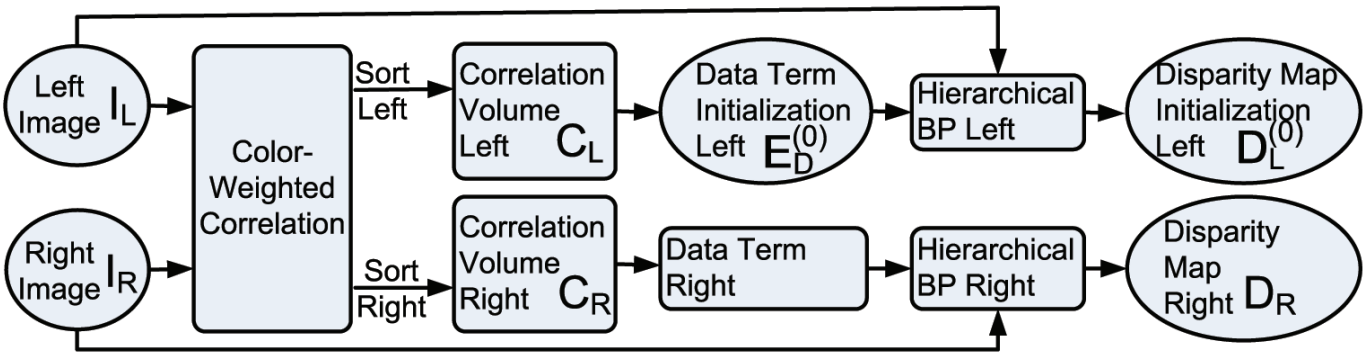


Fig. 1. The initialization module. Hierarchical BP is run with both the left and right images as the reference image. The initial data term is computed based on the color-weighted correlation, and then fed to the hierarchical BP module for disparity map estimation, see the text for more details.

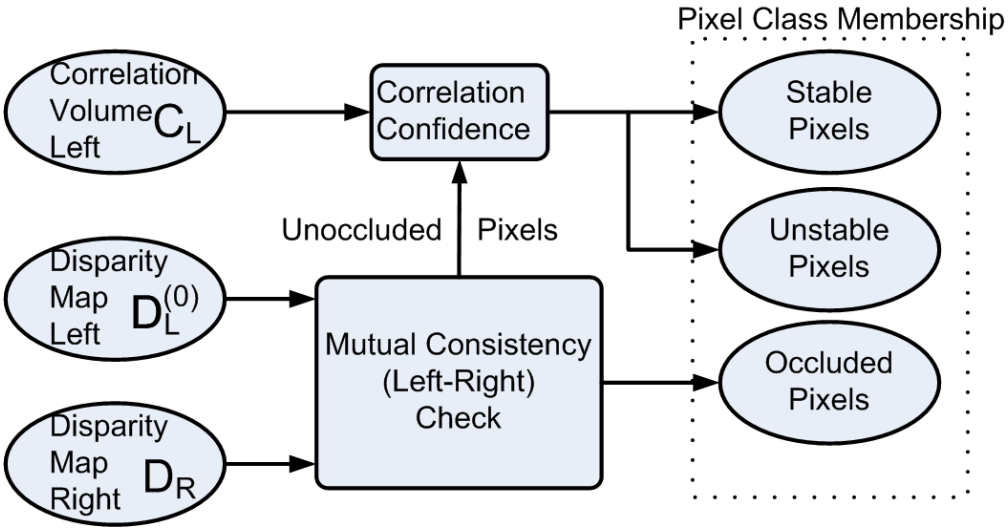


Fig. 2. The pixel classification module. Pixels are classified into occluded pixels, unstable pixels and stable pixels. The occluded pixels are the ones that fail a mutual consistency check. The non-occluded pixels are then further divided into stable and unstable pixels based on a confidence measure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

of standard belief propagation algorithms, a fast-converging belief propagation approach is developed.

A. Initialization

The main building blocks of the initialization module, shown in Figure 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 α_{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

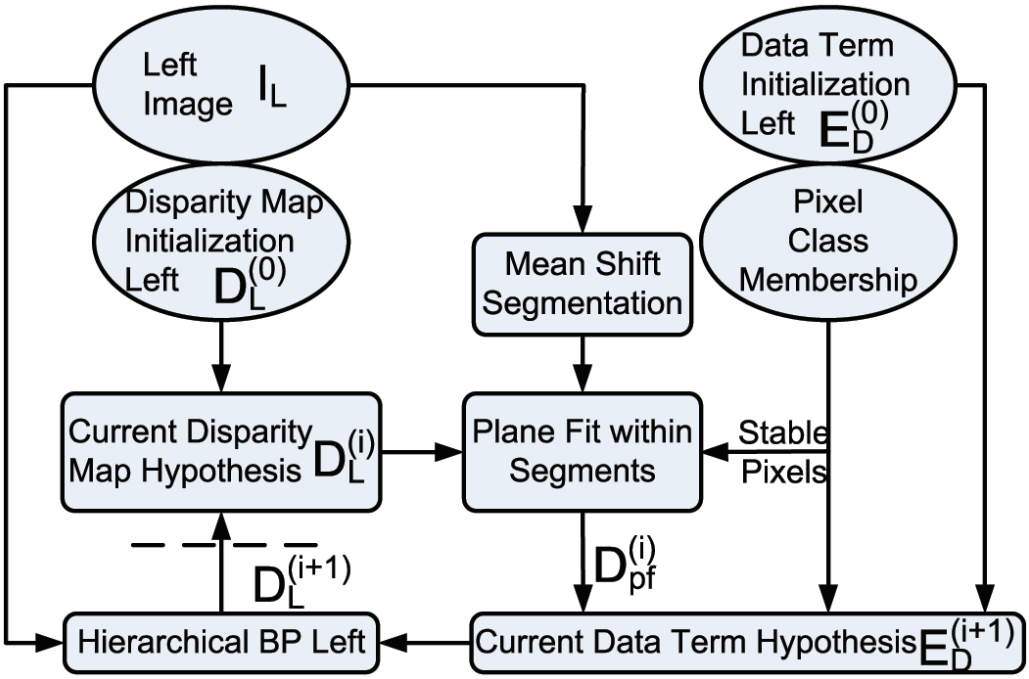


Fig. 3. The iterative refinement block, where the goal is to propagate information from the stable pixels to the unstable and the occluded pixels. Mean shift color segmentation is used to derive segments. Within each segment the plane fitting is then applied to the stable pixels, using the disparity 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 belief propagation. This gives a new disparity map hypothesis $D_L^{(i+1)}$, which is fed back into the process.

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

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

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

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

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

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

$$C_{L,\mathbf{x}_L}(d_{\mathbf{x}}) = \frac{\sum_{(\mathbf{y}_L,\mathbf{y}_R)\in W_{\mathbf{x}_L}\times W_{\mathbf{x}_R}} w_{\mathbf{x}_L\mathbf{y}_L} w_{\mathbf{x}_R\mathbf{y}_R} d(\mathbf{y}_L,\mathbf{y}_R)}{\sum_{(\mathbf{y}_L,\mathbf{y}_R)\in W_{\mathbf{x}_L}\times W_{\mathbf{x}_R}} w_{\mathbf{x}_L\mathbf{y}_L} w_{\mathbf{x}_R\mathbf{y}_R}},$$

where $W_{\mathbf{x}}$ is the support window around $\mathbf{x}$ and $d(\mathbf{y}_L, \mathbf{y}_R)$ represents the Birchfield and Tomasi's pixel dissimilarity, $\mathbf{x}_L$ and $\mathbf{y}_L$ are pixels in the left image $\mathbf{I}_L$, $\mathbf{x}_R = \mathbf{x}_L - d_{\mathbf{x}}$ and $\mathbf{y}_R = \mathbf{y}_L - d_{\mathbf{x}}$ are the corresponding pixels in the right image $\mathbf{I}_R$. $d_{\mathbf{x}}$ is the disparity value of pixel $\mathbf{x}_L$ in the left image.

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

$$E_{D,\mathbf{x}_L}^{(0)}(d_{\mathbf{x}}) = \lambda_{bp} \min(C_{L,\mathbf{x}_L}(d_{\mathbf{x}}), \eta_{bp}), \qquad (3)$$

where $\lambda_{bp} = 0.2$ is a constant which is set experimentally, and η_{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 belief propagation [9] module for disparity hypothesis estimation.

B. Pixel Classification

The main building blocks of the pixel classification step (see Figure 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.

$$D_L(\mathbf{x}_L) = D_R(\mathbf{x}_L - D_L(\mathbf{x}_L)) \tag{4}$$

for a particular pixel $\mathbf{x}_L$ in the left image. If this relation does not hold, the pixel is declared as occluded. Otherwise, the pixel is declared non-occluded 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 it is shown in Figure 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

$$|\frac{C_L^1 - C_L^2}{C_L^2}|. \tag{5}$$

If it is above a threshold α_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 Figure 4. It shows that most of the bad pixels are classified into either occluded pixels or unstable pixels.

C. Iterative Refinement

The main building blocks of the iterative refinement, see Figure 3, are the mean shift color segmentation, the plane fitting within segments, the data term formulation, and another hierarchical belief propagation process identical to that in Section III-A.

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

The mean shift color segmentation is performed as described in [6]; an example is shown in Figure 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 and occluded pixels), we use the result of the plane fitting. If the ratio of the stable pixels is below η_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

$$a_i = |D_L^{(i+1)} - D_{pf}^{(i)}| \tag{6}$$

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

(a) Pixel class membership. (b) Bad pixels. (c) Disparity map.

Fig. 4. The left figure is 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. The middle figure shows the bad pixels (black pixels) detected by comparing the disparity map with the ground truth. The right figure is the corresponding disparity map calculated by applying the winner-takes-all approach to the color-weighted correlation volume C_L.

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 constrain 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.

D. 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 dimension given by the number of possible labels, and at each iteration, the new messages are computed as follows:

$$M_{\mathbf{X},\mathbf{Y}}^t(d) = \arg\min_{d_{\mathbf{X}}}(E_{D,\mathbf{X}}(d_{\mathbf{X}}) + \sum_{s\in N(\mathbf{X}),\mathbf{X}\neq\mathbf{Y}} M_{s,\mathbf{X}}^{t-1}(d_{\mathbf{X}}) + h(d_{\mathbf{X}}, d)), \quad (7)$$

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 $h(d_{\mathbf{X}}, d)$ 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 $h(d_{\mathbf{X}}, d)$ are based on the degree of difference between labels. In order to allow for discontinuities, the truncated linear model is commonly adopted:

$$h(d_{\mathbf{X}}, d) = \mathbf{min}(\alpha_{\mathbf{bp}}, \rho_{\mathbf{bp}} \mid \mathbf{d_X} - \mathbf{d} \mid), \quad (9)$$

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 δ_{mean} is then subtracted out to yield the normalized difference δ_{norm}. Define the cost coefficient

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

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

$$h(d_{\mathbf{X}}, d) = \mathbf{min}(\alpha_{\mathbf{bp}}, \rho_{\mathbf{s}}\rho_{\mathbf{bp}} \mid \mathbf{d_X} - \mathbf{d} \mid). \quad (12)$$

$h(d_{\mathbf{X}}, d)$ depends on both $\delta_{\mathbf{X},\mathbf{Y}}$ and δ_{mean}. $h(d_{\mathbf{X}}, d)$ steadily decreases with the increase of $\delta_{\mathbf{X},\mathbf{Y}}$, because it is likely to be a color edge. A large δ_{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, $h(d_{\mathbf{X}}, d)$ steadily increases with the increase of δ_{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.

Standard loopy belief propagation algorithm is too slow to be practically used to achieve nice results. Felzenszwalb [9] proposed a hierarchical algorithm which 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: (a) initialize the messages at the coarsest level to all zeros, (b) apply loopy BP at the coarsest scale to iteratively refine the messages. (c) use refined messages from the coarser level to initialize the messages for the next scale. In detail, assume $\mathbf{X}$ is one pixel in the coarsest scale, and its corresponding pixels in the finer scale are $\mathbf{X}'_i$, $i \in [1, 4]$ as it is shown in Figure 6, then

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

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

where $Y'_{i,j}$ are the four neighbors of pixel $\mathbf{X}'_i$, and $\mathbf{Y}_j$ are the corresponding four neighbors of pixel $\mathbf{X}$. For instance, assume $Y'_{i,j}$ 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 it is shown in Equation 14. And for simplicity, ρ_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 Equation 8.

Two main parameters s_{bp} and n_{bp} define the behavior of this hierarchical belief propagation 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 are provided in Section V-A.

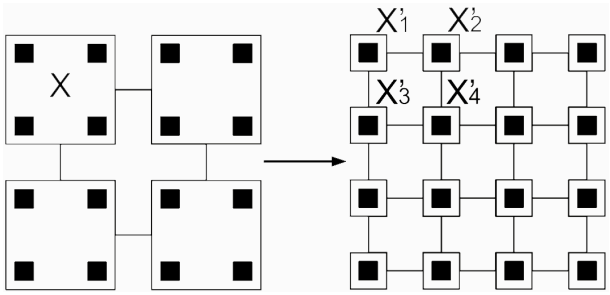


Fig. 6. Illustration of two levels in the coarse-to-fine method. Each node $\mathbf{X}$ in left figure corresponds to a block of four nodes $\mathbf{X}'_i$ in the right figure.

E. Fast-Converging Belief Propagation

In 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 off-line 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 [8] maintained a priority queue of messages and updated the one with the highest residual. Ihler [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 III-D. 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 Equation 7, the new messages ($M^t_{\mathbf{X},\mathbf{Y}_j}$, $j \in [1, 4]$) 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 ($M^{t-1}_{\mathbf{Y}_j,\mathbf{X}}$) received from $\mathbf{Y}_j$, and the jump cost function h. Since $E_{D,\mathbf{X}}$ and h stay unchanged for every iteration, $M^t_{\mathbf{X},\mathbf{Y}_j}$ is updated according to $M^{t-1}_{\mathbf{Y}_j,\mathbf{X}}$ only. For every pixel $\mathbf{X}$, 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

$M^t_{\mathbf{X},\mathbf{Y}_j}$ only if the L1-norm of their difference is larger than a threshold that is small enough:

$$|M^{t-1}_{\mathbf{Y}_j,\mathbf{X}} - M^{t-2}_{\mathbf{Y}_j,\mathbf{X}}|_1 \geq \eta_z \tag{15}$$

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

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 $\mathscr{R}(t) = \frac{\Delta E(t)}{E_{sbp}(t)}$, and let $\widetilde{\Delta E}$ and $\widetilde{\mathscr{R}}$ denote the average of $\Delta E(t)$ and $\mathscr{R}(t)$ over 50 iterations respectively. Table I shows that $\widetilde{\Delta E}$ and $\widetilde{\mathscr{R}}$ decrease as the threshold η_z decreases, and finally reach zero when $\eta_z = 1.0e^{-16}$. But please note that smaller η_z implies more non-converged pixels as shown in Figure 7 (a), and thereby increases the running time. In our experiments, we set $\eta_z = 0.1$, which is observed to greatly improve the speed while well preserve the accuracy. Figure 7 (b) 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 non-converged pixels, which greatly decreases the running time of BP approaches with large number of iterations. Figure 7 (c) shows that unlike standard BP, the running time of fast-converging BP is sub-linear to the number of iterations.

η_z	$\widetilde{\Delta E}$	$\widetilde{\mathscr{R}}$
1.0e-01	1.084e+02	3.120e-04
1.0e-04	0.361e-05	0.101e-10
1.0e-16	0	0

TABLE I

COMPARISON OF THE PERFORMANCE OF STANDARD BP AND FAST-CONVERGING BP OF THE TSUKUBA DATA SET.

IV. DEPTH ENHANCEMENT

To reduce the discontinuities caused by the quantization in the depth hypothesis selection process, a sub-pixel 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:

$$f(x) = ax^2 + bx + c. \tag{16}$$

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:

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

At the end of our integer-based stereo pipeline, the disparity map $D^{(5)}_L$ is produced, which means that the three depth candidates d, d_-, 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 $f(d)$, $f(d_-)$, $f(d_+)$ of the three depth candidates are selected from the color-weighted correlation volume, which is denoted by C_L in Figures 1 and 2. The sub-pixel estimation is then carried out by using Equation (17) to calculate the sub-pixel disparity map. Finally, we replace each value with the average of those values that are within one disparity over a 9×9 window.

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

V. EXPERIMENTS

A. Parameter Settings

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

The parameters are shown in Table II and separated into four parts: three parameters (α_{ms}, β_{ms}, γ_{ms}) for the mean shift segmentation, three parameters (α_{cw}, β_{cw}, γ_{cw}) for the color-weighted filter, six parameters (α_{bp}, η_{bp}, ρ_{bp}, λ_{bp}, s_{bp}, n_{bp}) for the hierarchical belief propagation, and six parameters (κ_s, κ_u, κ_o, α_s, η_s, n_s) for the iterative refinement.

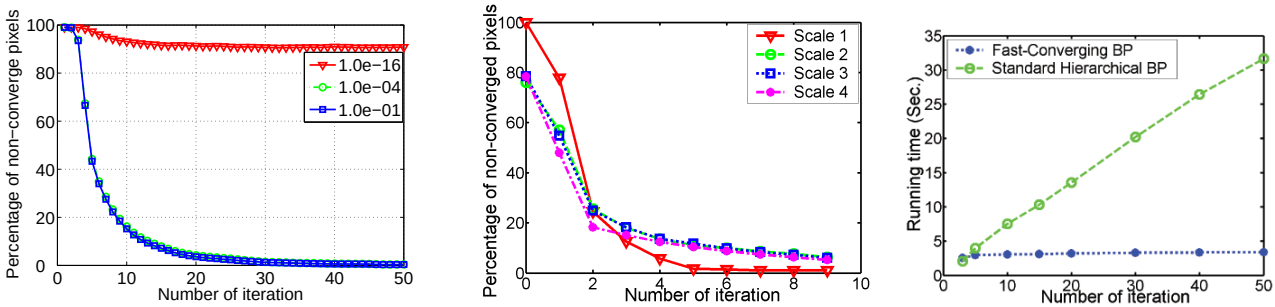


Fig. 7. (a) shows the percentage of the non-converged pixels after every iteration for three η_z values: $1.0e^{-01}$, $1.0e^{-04}$ and $1.0e^{-16}$. By setting $\eta_z = 0.1$, (b) provides the percentage of the non-converged pixels after every iteration for different scales, and (c) compares the running time 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.

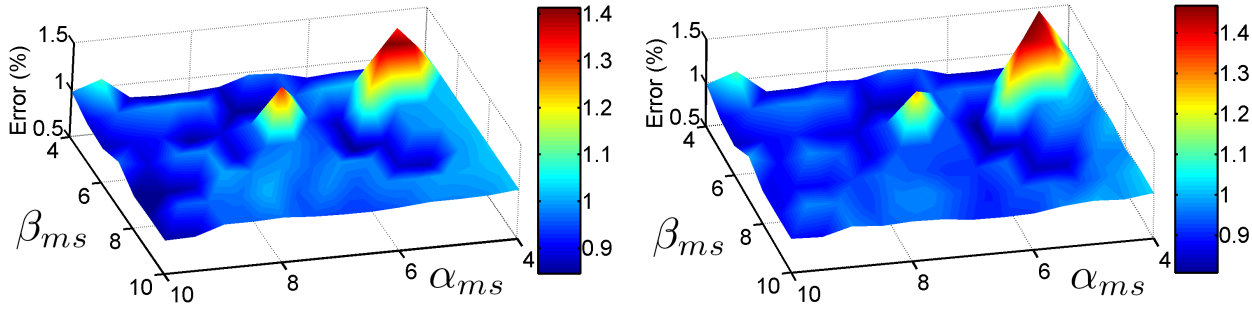


Fig. 9. Performance according to the mean shift parameters (α_{ms} is spatial bandwidth, β_{ms} is color bandwidth, and γ_{ms} is the minimum region size) for Tsukuba data set. (a) is the behavior of the algorithm according to the parameters α_{ms} and β_{ms} by setting $\gamma_{ms} = 20$, and in (b),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

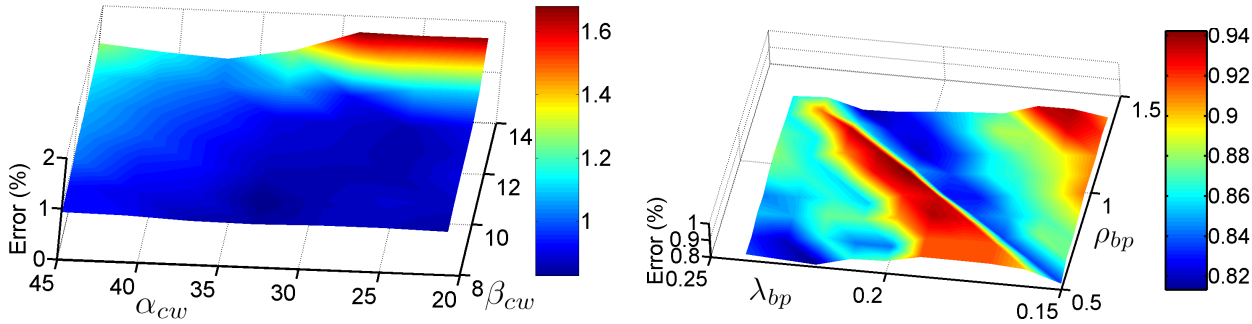


Fig. 10. Performance according to the color-weighted filter and the hierarchical BP parameters for Tsukuba data set. α_{cw} is the size of the support window, β_{cw} is defined in Equation (2), ρ_{bp} is the rate of increase in the jump cost and λ_{bp} is the scaling factor applied to the correlation volume after the truncation.

In the next few paragraphs we explain these parameters and evaluate their effect on the stereo results.

For mean shift color segmentation, α_{ms} is the spatial bandwidth, β_{ms} is the color bandwidth, and γ_{ms} is the minimum region size. They are set as the default values. Figure 9 shows the performance of the proposed method according to the mean shift segmentation parameters. Figure 9 (a) and (b) set γ_{ms} to constants 20 and 50 respectively, and show the behavior of the proposed method for α_{ms} and β_{ms}, both range from 4 to 10. The proposed method is very robust against α_{ms} and β_{ms}, especially within the range of 6 and 10. Visually comparison of Figure 9 (a) and (b) also shows that the method is robust

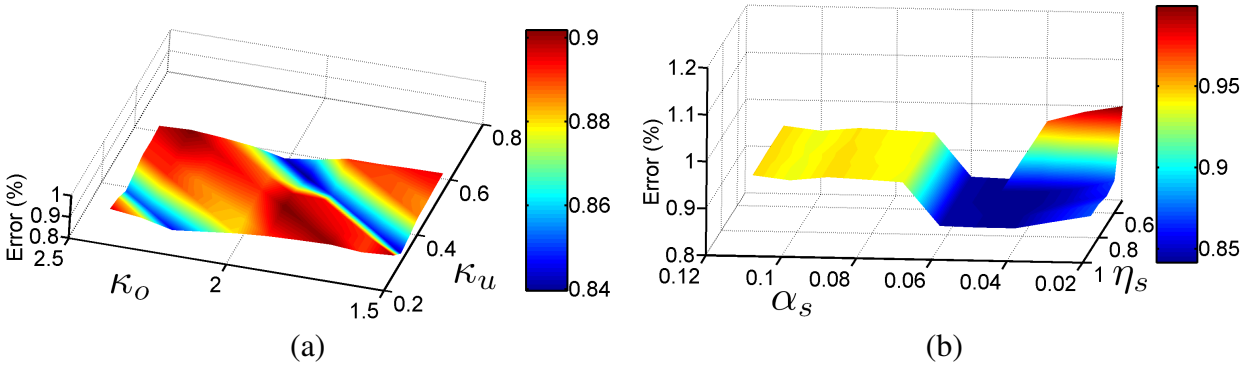


Fig. 11. Performance according to the iterative refinement parameters for Tsukuba data set. κ_u is the weight for unstable pixels, κ_o is the weight for occluded pixels, α_s is the threshold on correlation confidence defined in Section III-B and η_s is the threshold on the ratio of the stable pixels to all pixels.

(a) Disparity maps. (b) Synthesized views.

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

Mean Shift	α_{ms}	β_{ms}	γ_{ms}			
Segmentation	7	6	20			
Color-Weigh.	α_{cw}	β_{cw}	γ_{cw}			
Correlation	33	10	21			
Hierarchical	λ_{bp}	η_{bp}	ρ_{bp}	α_{bp}	s_{bp}	n_{bp}
BP	0.2	$2\bar{c}$	1.0	$n_d/8$	4	50
Iterative	κ_s	κ_u	κ_o	α_s	η_s	n_s
Refinement	0.05	0.5	2	0.04	0.7	5

TABLE II

PARAMETER SETTINGS USED THROUGHOUT. n_d IS THE NUMBER OF DISPARITY LEVELS. $\bar{c}$ IS THE AVERAGE OF THE VALUES IN THE CORRELATION VOLUME.

against γ_{ms}. The running time of the mean shift segmentation is about 20 seconds for the Tsukuba data set.

For color-weighted filter, α_{cw} is the size of the support window and β_{cw} and γ_{cw} are defined in Equation (2). Figure 10 (a) shows the performance of the proposed method according to α_{cw} and β_{cw} by keeping γ_{cw} constant. We can see that the proposed method is fairly robust against different sizes of a support window (α_{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 β_{cw}, the error rate is almost constant for β_{cw} between 8 and 12. The running time of the color-weighted filtering depends mostly on the parameter α_{cw}. When $\alpha_{cw} = 33$, the running time is about 30 seconds for the Tsukuba data set.

For hierarchical BP, α_{bp} and η_{bp} are truncations of the jump cost and correlation volume, respectively. The parameter ρ_{bp} is the rate of increase in the jump cost and λ_{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 III-A. Parameters ρ_{bp}, λ_{bp}, s_{bp} and n_{bp} are set experimentally. According to our experiments, we found that ρ_{bp} and λ_{bp} are the most sensitive ones, and their performances are studied in Figure 10 (b). Our method is very robust when $\rho_{bp} \in [0.5, 1.5]$ and $\lambda_{bp} \in [0.15, 0.25]$. The running time of the hierarchical BP is about 2 seconds for the Tsukuba data set.

Parameters κ_s, κ_u and κ_o for the iterative refinement are defined in Section III-C. Among them, κ_s is less important because it is the weight for the stable pixels. The performance of our method according to κ_u and κ_o are provided in Figure 11 (a) which shows that our method is pretty robust over non-trivial ranges. α_s is the threshold on correlation confidence defined in Section III-B. Parameter η_s is related to the plane fitting process, as defined in Section III-C. Figure 11 (b) evaluates the performance of our method according to α_s and η_s, which shows the performance does not depend much on η_s unless it is very small. The smaller η_s is, the fewer number of the color segments will the iterative refinement apply to. The performance according to α_s is robust within the range of 0.02 and 0.12. However, the larger the α_s, the fewer the stable pixels are in each segment, which makes the RANSAC plane fitting less robust; also, the smaller the α_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, Figure 13 shows that it always converge after five iterations. The running time of the refinement is about 10 seconds per iteration for the Tsukuba data set.

B. Experimental results

We evaluate our algorithm on the Middlebury data set and we show in Table III and IV that our algorithm on 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:

- The subset of the non-occluded pixels, denoted "nonoccl".
- The subset of the pixels near the occluded areas, denoted "disc".
- The subset of the pixels being either non-occluded or half-occluded, denoted "all".

The ranks show that our algorithm works pretty well in the non-occluded 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 [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 preserves the accuracy of the other areas.

In Figure 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 V 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 Figure 13 and Figure 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.

We also test our algorithm with some outdoor scenes, which are probably more challenging. Some experimental results are provided in Figure 15. As no ground truth is available, we used the disparity map to synthesize novel 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 sub-pixel accuracy, we proposed a depth-enhanced approach for post-processing purpose. The experimental results on the Middlebury data sets are provided on the fifth row in Figure 12. A set of synthesized views using the disparity maps on the fourth and the fifth row in Figure 12 are shown in Figure 16, providing a visual comparison of the algorithms with and without depth enhancement. The improvement is obvious. The results shown in column (a) are quantized to discrete number of planes. After sub-pixel estimation, the quantization effect is removed, as it is shown in column (b). The average ranks are provided in Table IV, showing that our algorithm is currently state-of-the-art with sub-pixel accuracy.

VI. Conclusions

In this paper, a stereo model based on energy minimization, color segmentation, plane fitting, repeated

Algorithm	Avg. Rank	Tsukuba nonocc	Tsukuba all	Tsukuba disc	Venus nonocc	Venus all	Venus disc	Teddy nonocc	Teddy all	Teddy disc	Cones nonocc	Cones all	Cones disc
Our Algorithm	**2.1**	**0.88**$_1$	**1.29**$_1$	**4.76**$_1$	0.13$_2$	0.45$_3$	1.87$_2$	**3.53**$_1$	8.30$_3$	**9.63**$_1$	2.90$_3$	8.78$_5$	7.79$_2$
AdaptingBP [15]	2.2	1.11$_4$	1.37$_3$	5.79$_4$	**0.10**$_1$	**0.21**$_1$	**1.44**$_1$	4.22$_3$	7.06$_2$	11.8$_3$	**2.48**$_1$	7.92$_2$	**7.32**$_1$
DoubleBP [26]	3.4	0.88$_2$	1.29$_2$	4.76$_2$	0.14$_3$	0.6 $_7$	2.00$_4$	3.55$_2$	8.71$_4$	9.70$_2$	2.90$_4$	9.24$_6$	7.80$_3$
SymBP+occ [20]	6.3	0.97$_3$	1.75$_6$	5.09$_3$	0.16$_4$	0.33$_2$	2.19$_5$	6.47$_7$	10.7$_5$	17.0$_8$	4.79$_{13}$	10.7$_{10}$	10.9$_9$
Segm+visib [2]	6.4	1.30$_7$	1.57$_4$	6.92$_9$	0.79$_{10}$	1.06$_8$	6.76$_{11}$	5.00$_4$	**6.54**$_1$	12.3$_4$	3.72$_7$	8.62$_4$	10.2$_8$
C-SegmGlob [12]	7.3	2.61$_{18}$	3.29$_{13}$	9.89 $_{15}$	0.25$_6$	0.57$_4$	3.24$_7$	5.14$_5$	11.8$_6$	13.0$_5$	2.77$_2$	8.35$_3$	8.20$_4$

TABLE III

COMPARISON OF THE RESULTS ON THE MIDDLEBURY DATA SET WITH ERROR THRESHOLD 1. THE NUMBERS IN THE LAST TWELVE COLUMNS ARE THE PERCENTAGE OF THE PIXELS WITH INCORRECT DISPARITIES ON THE DIFFERENT SUBSETS OF THE IMAGES. OUR ALGORITHM IS RANKED AT THE FIRST PLACE ON AVERAGE.

Algorithm	Avg. Rank	Tsukuba nonocc	Tsukuba all	Tsukuba disc	Venus nonocc	Venus all	Venus disc	Teddy nonocc	Teddy all	Teddy disc	Cones nonocc	Cones all	Cones disc
Our Algorithm	**3.0**	8.78$_5$	9.45$_3$	14.9$_2$	**0.72**$_1$	**1.12**$_1$	**5.24**$_1$	10.1$_2$	16.4$_2$	**21.3**$_1$	8.49$_7$	14.7$_7$	16.5$_4$
C-SemiGlob [12]	4.5	13.9$_{10}$	14.7$_{10}$	18.9$_{13}$	3.30$_3$	3.82$_2$	10.9$_5$	**9.82**$_1$	17.4$_4$	22.8$_2$	5.37$_2$	**11.7**$_1$	**12.8**$_1$
SemiGlob [11]	6.0	13.4$_9$	14.3$_9$	20.3$_{15}$	4.55$_5$	5.38$_6$	15.7$_9$	11.0$_4$	18.5$_5$	26.1$_5$	**4.93**$_1$	12.5$_2$	13.5$_2$
AdaptingBP[15]	6.3	19.1$_{15}$	19.3$_{13}$	17.4$_6$	4.84$_7$	5.08$_5$	7.84$_2$	12.8$_6$	16.7$_3$	26.3$_6$	7.02$_4$	13.2$_5$	14.0$_3$
Segm+visib[2]	6.8	12.7$_7$	12.9$_6$	15.8$_5$	10.4$_{14}$	11.0$_{14}$	19.5$_{13}$	11.0$_3$	**13.2**$_1$	23.7$_3$	8.12$_6$	13.1$_4$	17.3$_6$
DoubleBP[26]	8.8	18.7$_{13}$	19.1$_{12}$	15.8$_4$	7.85$_{11}$	8.38$_{10}$	11.6$_6$	14.3$_7$	19.9$_6$	24.3$_4$	11.9$_{11}$	18.1$_{10}$	19.9$_{11}$

TABLE IV

COMPARISON OF RESULTS ON THE MIDDLEBURY DATA SET WITH ERROR THRESHOLD 0.5. THE NUMBERS IN THE LAST TWELVE COLUMNS ARE THE PERCENTAGE OF PIXELS WITH INCORRECT DISPARITIES ON DIFFERENT SUBSETS OF THE IMAGES. THE AVERAGE RANKS SHOW THAT OUR ALGORITHM IS THE 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.

Algorithm	Tsukuba nonocc	Tsukuba all	Tsukuba disc	Venus nonocc	Venus all	Venus disc	Teddy nonocc	Teddy all	Teddy disc	Cones nonocc	Cones all	Cones 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

TABLE V

THE FIRST FOUR ROWS IN THE TABLE CORRESPOND TO THE FIRST FOUR ROWS IN FIGURE 12. THE LAST ROW IS THE SAME THE FOURTH ROW ($D_L^{(5)}$) EXCEPT THAT WE DO NOT USE THE COLORS OF THE REFERENCE IMAGE TO DEFINE THE JUMP COST.

application of hierarchical belief propagation, and depth enhancement was proposed. Typically, one application of the hierarchical belief propagation 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 belief propagation approach is proposed, which preserves the same accuracy as the standard BP. The running time of fast-converging BP is sub-linear to the number of iterations.

Our algorithm is currently outperforming the other algorithms on the Middlebury data set 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 [20] suggests that by generating a good disparity map

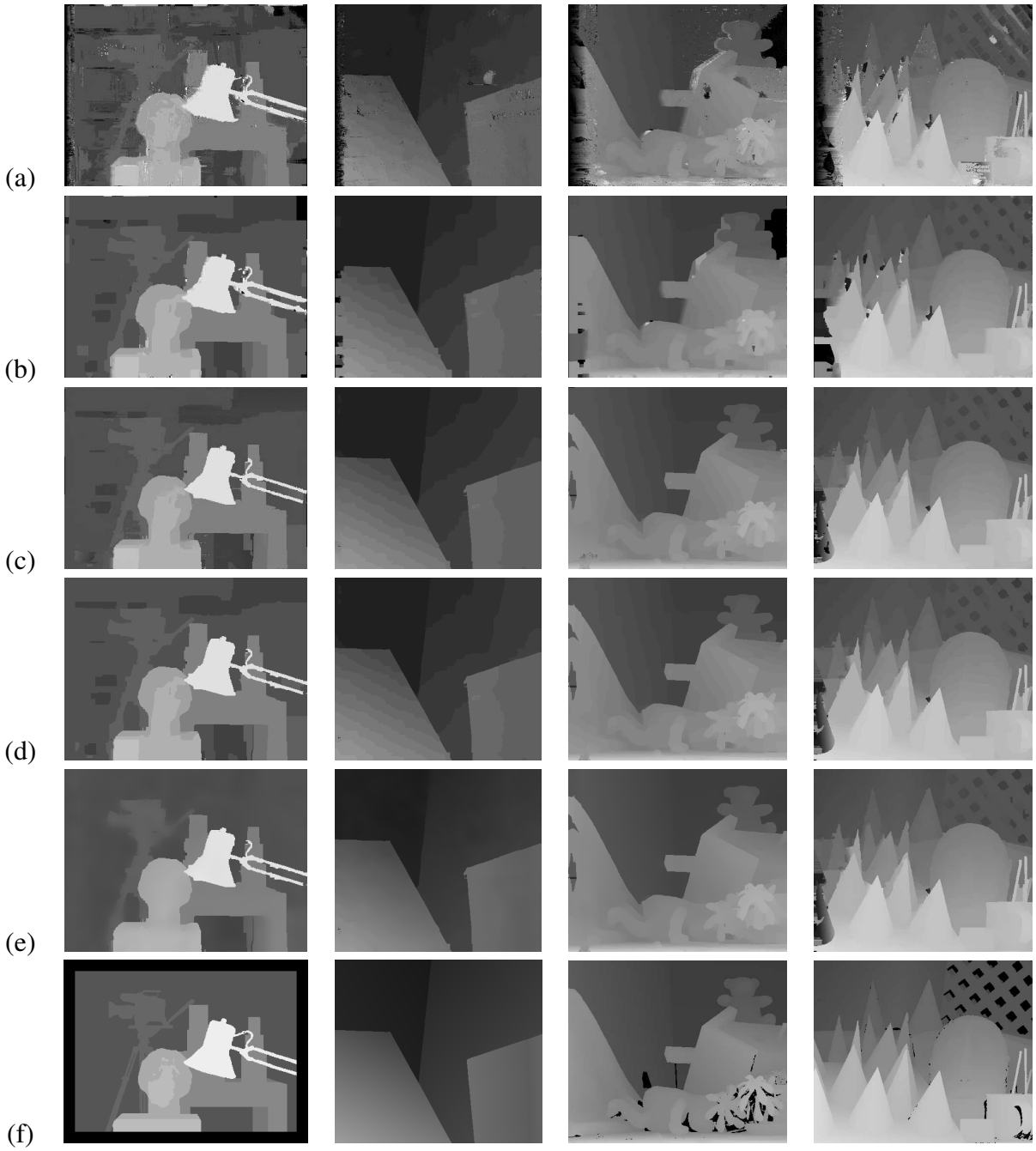


Fig. 12. Intermediate results from our algorithm for the four different standard test sets compared to the ground truth. (a) provides the disparity maps after applying the winner-takes-all approach to the color-weighted correlation volume C_L. (b) are the output of the initial hierarchical BP. This result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

for the right image, the occlusion constraints can be extracted more accurately. Our algorithm performs well when the scene is mainly composed of planar surfaces, because the depth information of the unstable areas are 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 Transactions on Pattern Analysis*

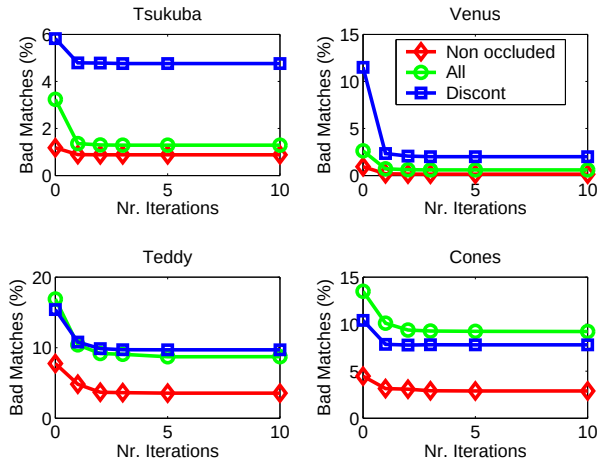


Fig. 13. The iterated computations of $E_D^{(i+1)}$ improves the result. In most cases one iteration is enough for convergence. After five iterations, the algorithm has always converged.

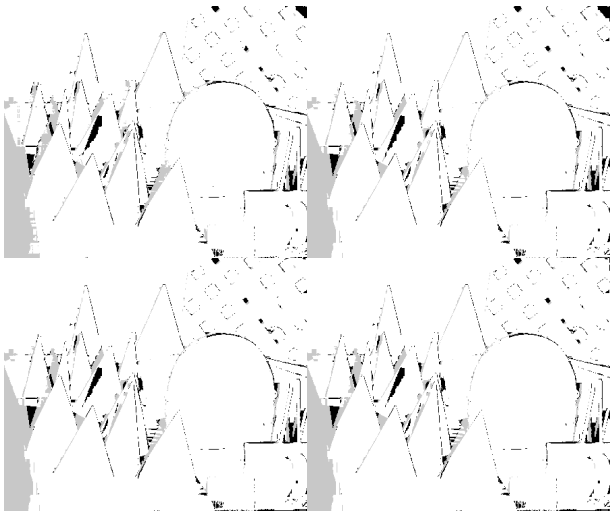


Fig. 14. Pixels with incorrect disparity for the "Cones" data set. On the first row the results after 1 and 2 iterations are shown and on the second row the results after 3 and 5 iterations are shown. The gray and black pixels are bad pixels with absolute disparity error larger than 1, and the gray pixel are half-occluded.

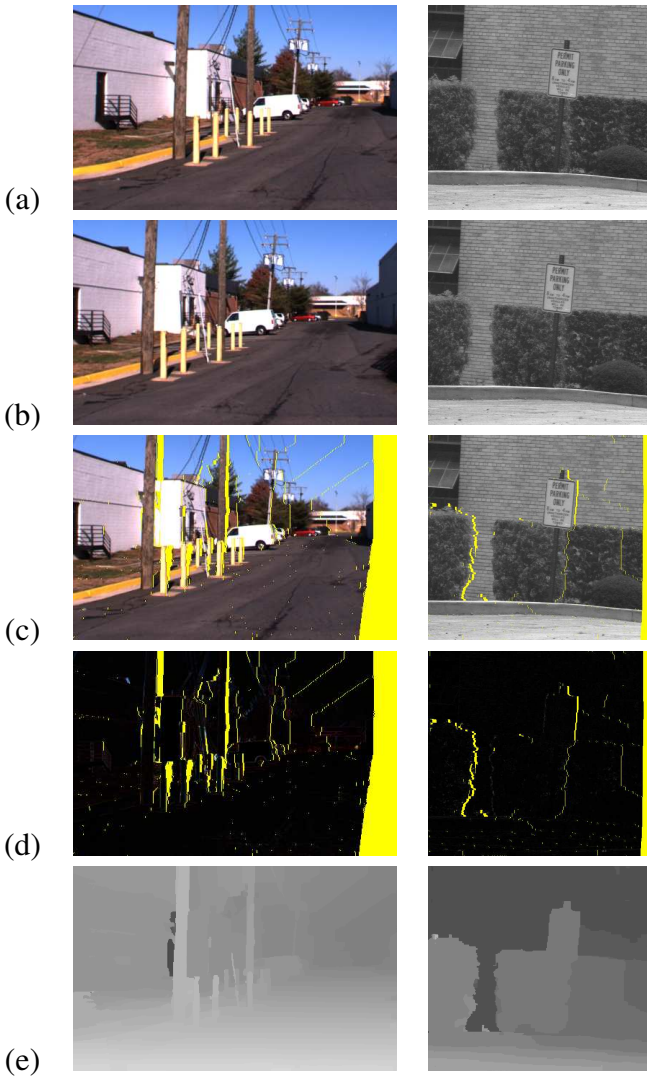


Fig. 15. (a) and (b) are the left and right camera images, (c) are the synthesized right image, (d) are the absolution difference of (b) and (c), (e) are corresponding disparity maps.

and Machine Intelligence, Vol. 20, No. 4, April 1998.

[2] M. Bleyer and M. Gelautz, A Layered Stereo Algorithm Using Image Segmentation and Global Visibility Constraints, *IEEE International Conference on Image Processing*, pp. 2997-3000, 2004.

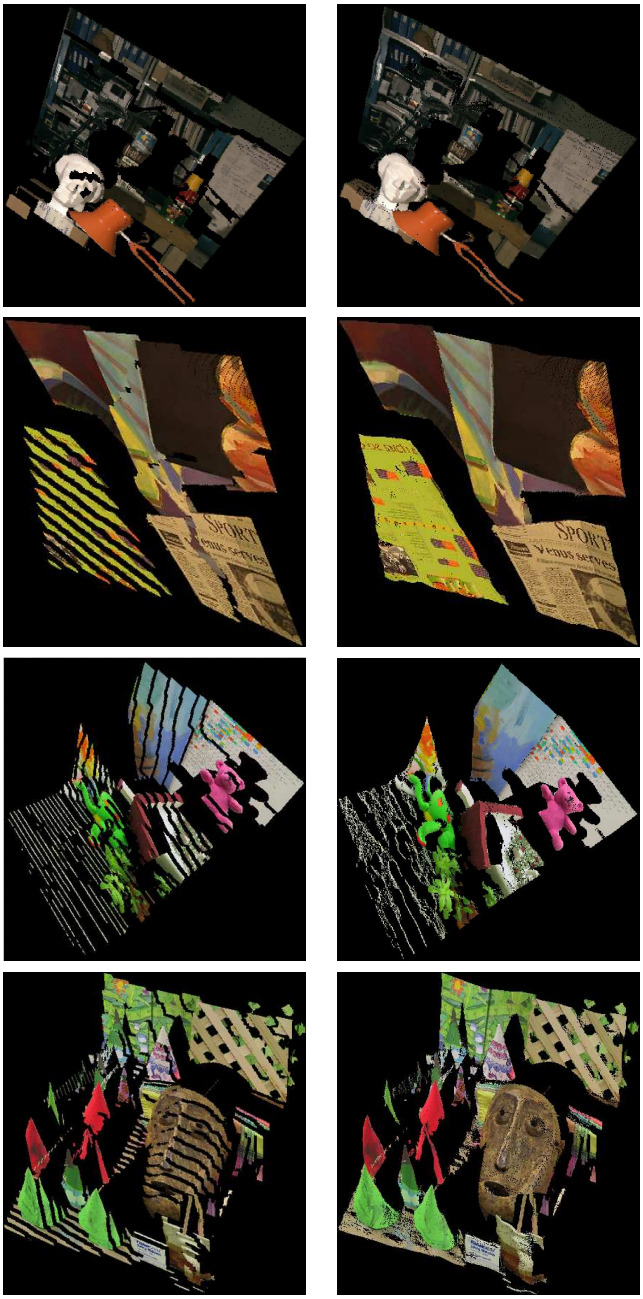
[3] Y. Boykov, O. Veksler and R. Zabih, A Variable Window Approach to Early Vision, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 20, No. 12, 1998.

[4] Y. Boykov, O. Veksler and R. Zabih, Fast Approximate Energy Minimization via Graph Cuts, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 23, No. 11, 2001.

[5] L. Chen, M. Cetin, and A. S. Willsky, Distributed Data Association for Multi-Target Tracking in Sensor Networks, *International Conference on Information Fusion*, 2005.

[6] D. Comaniciu and P. Meer, Mean shift: A Robust Approach Toward Feature Space Analysis, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 5, May 2002.

[7] G. Egnal and R. Wildes, Detecting Binocular Half-Occlusions: Empirical Comparisons of Five Approaches, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1127-



(a) Before. (b) After.

Fig. 16. (a) Synthesized views without depth enhancement. (b) Synthesized views after depth enhancement. The results shown in column (a) are quantized to discrete number of planes. Column (b) shows that after sub-pixel estimation, the quantization effect is removed.

1133, 2002.

[8] G. Elidan, I. McGraw, and D. Koller, Residual Belief Propagation: Informed Scheduling for Asynchronous Message Passing, *Proceedings of the Twenty-second Conference on Uncertainty in AI (UAI)*, 2006.

[9] P. F. Felzenszwalb and D. P. Huttenlocher, Efficient Belief Propagation for Early Vision, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. I: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, *Communications of the ACM*, Vol. 24, pp. 381-395 1981.

[11] H. Hirschmüeller, Accurate and Efficient Stereo Processing by Semi-Global Matching and Mutual Information,*IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol II:807-814, 2005.

[12] H. Hirschmüller, Stereo Vision in Structured Environments by Consistent Semi-Global Matching, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. II:2386-2393, 2006.

[13] A. T. Ihler, J. W. Fisher, A. S. Willsky, Loopy Belief Propagation: Convergence and Effects of Message Errors, *Journal of Machine Learning Research*, Vol. 6:905-936, 2005.

[14] T. Kanade and M. Okutomi, A Stereo Matching Algorithm with an Adaptive Window: Theory and Experiments, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 16, No. 9, September 1994.

[15] A. Klaus, M. Sormann and K. F. Karner, Segment-Based Stereo Matching Using Belief Propagation and a Self-Adapting Dissimilarity Measure, *International Conference of Pattern Recognition* , Vol. III:15-18, 2006.

[16] V. Kolmogorov and R. Zabih, Computing Visual Correspondence with Occlusions using Graph Cuts, *IEEE International Conference on Computer Vision*, Vol. I:508-515 2001.

[17] D. Scharstein and R. Szeliski, Middlebury Stereo Vision Research Page, $http://vision.middlebury.edu/stereo/eval/$

[18] D. Scharstein and R. Szeliski, A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms, *International Journal of Computer Vision*, 47:7-42, 2002.

[19] J. Sun, N.-N. Zheng and H.-Y. Shum, Stereo Matching Using Belief Propagation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25, No. 7, July 2003.

[20] J. Sun, Y. Li, S. B. Kang and H.-Y. Shum,

Symmetric Stereo Matching for Occlusion Handling, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. II:399-406, 2005.

[21] H. Tao and H. Sawhney, Global Matching Criterion and Color Segmentation Based Stereo, *IEEE Workshop on Applications of Computer Vision*, pp. 246-253, 2000.

[22] M. Tappen and W. Freeman, Comparison of Graph Cuts with Belief Propagation for Stereo, *IEEE International Conference on Computer Vision*, Vol. I:508-515 2003.

[23] O. Veksler, Stereo Correspondence with Compact Windows via Minimum Ratio Cycle, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 12, 2002.

[24] O. Veksler, Fast Variable Window for Stereo Correspondence using Integral Images, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. I:556-561, 2003.

[25] Y. Weiss and W.T. Freeman, On the Optimality of Solutions of the Max-Product Belief Propagation Algorithm in Arbitrary Graphs, *IEEE Transactions on Information Theory*, Vol. II: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, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. II:2347-2354, 2006.

[27] K.-J. Yoon and I.-S. Kweon, Adaptive Support-Weight Approach for Correspondence Search, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):650-656, 2006.

Qingxiong Yang received the B.E. degree from University of Science and Technology of China (USTC) in 2004 and MSc degree in Computer Science from University of Kentucky in 2007, and is currently a Ph.D. 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.

Liang Wang received his B.S. degree from the School of Computer Science, Beijing University of Aeronautics and Astronautics in 2004. He is currently a Ph.D 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.

Ruigang Yang is an Assistant Professor in the Computer Science Department at the University of Kentucky. He received his Ph.D. degree in Computer Science from University of North Carolina at Chapel Hill in 2003. Prior to coming to UNC-Chapel Hill, he earned a M.S. degree in Computer Science from Columbia University in 1998. Dr. Yang's research interests include computer graphics, computer vision, and multimedia. He is a recipient of U.S. NSF CAREER award in 2004, and a member of the IEEE Computer Society and ACM.

Henrik Stewénius 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', all from Lund University, Sweden. He is currently an assistant research professor at the Center for Visualization and Virtual Environments at the University of Kentucky.

His research interests are in geometry and scalable recognition.

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