Stereo Reconstruction with Mixed Pixels Using Adaptive Over-Segmentation

Yuichi Taguchi *
The University of Tokyo
Tokyo, Japan
yuichi@hc.ic.i.u-tokyo.ac.jp

Bennett Wilburn
Microsoft Research Asia
Beijing, China
bwilburn@microsoft.com

C. Lawrence Zitnick
Microsoft Research
Redmond, WA
larryz@microsoft.com

Abstract

We present an over-segmentation based, dense stereo algorithm that jointly estimates segmentation and depth. For mixed pixels on segment boundaries, the algorithm computes foreground opacity (alpha), as well as color and depth for the foreground and background. We model the scene as a collection of fronto-parallel planar segments in a reference view, and use a generative model for image formation that handles mixed pixels at segment boundaries. Our method iteratively updates the segmentation based on color, depth and shape constraints using MAP estimation. Given a segmentation, the depth estimates are updated using belief propagation. We show that our method is competitive with the state-of-the-art based on the new Middlebury stereo evaluation, and that it overcomes limitations of traditional segmentation based methods while properly handling mixed pixels. Z-keying results show the advantages of combining opacity and depth estimation.

1. Introduction

Dense stereo matching is challenging in the presence of occlusions and textureless image regions. Color segmentation based methods have been shown to effectively handle these cases [15, 9, 3, 8, 24]. These approaches assume that depth varies smoothly within regions of homogeneous color and that depth discontinuities coincide with color boundaries. This assumption helps resolve the depth ambiguity within textureless regions and allows for precise delineation of object boundaries corresponding to depth discontinuities.

A drawback of segmentation based stereo is that depth discontinuities may not lie along color boundaries. As a result, image segmentations based on color information may contain segments that span depth discontinuities. If the color segmentation is held fixed, errors will result in the final depth map [15, 24]. We present an approach that overcomes initial segmentation errors by jointly estimating depth and image segmentation.

Most dense stereo methods compute a single depth value for each pixel. For mixed pixels (pixels which span two objects at different depths), computing the depths of the foreground and background components of the pixel gives a more complete understanding of the scene structure. Moreover, one must compute the opacity (alpha) and foreground/background colors for mixed pixels in order to get high-quality results for Z-keying and view interpolation. Alpha estimation is usually done as a post-processing step [11, 5, 24, 7], given a presumed pixel-accurate depth map. We incorporate alpha estimation into the depth and segmentation computation to produce more accurate results. To be clear, our goal is not calculating matting and depth information for very fuzzy or hairy foreground objects. Instead, we focus on the mixed pixels that occur along the boundaries of nearly all objects in the scene, at all depths.

Our stereo reconstruction method jointly estimates image segmentation, depth, and matting/depth information for mixed pixels. We use an over-segmentation approach to represent a scene as a collection of fronto-parallel planar segments. The segments are characterized by their depth, 2D shape, and color. These parameters are jointly estimated by alternating the update of segment shapes and depths. To update the segment shapes, we use a generative model that accounts for mixed pixels at the segment boundary as well as the depth and shape probabilities. To update the segment depths, we define a pairwise Markov random field for the segments, and minimize its energy using belief propagation. The algorithm explicitly handles occlusions by checking the visibility of pixels based on the previous estimates of segment depths.

The rest of this paper is organized as follows. In the next section, we review the prior art and identify our contributions. Section 3 describes our scene representation and stereo image model. Section 4 explains how we infer the scene structure. In section 5, we validate our methods using the new Middlebury stereo evaluation. Our method is currently ranked fourth best, and performs well on image pairs that confound most segment-based approaches. A Z-keying example shows the ability of the algorithm to extract alpha values across the entire range of depths in the scene. Finally, we close with a discussion of the strengths and weaknesses of our method, and some comments on the treatment of mixed pixels in the Middlebury stereo evaluation.

[*] This work was done while the first author was visiting Microsoft Research Asia.

2. Related Work

This section describes prior work related to segmentation-based stereo and alpha matting. For a comprehensive review of dense two-frame stereo methods, we refer the reader to Scharstein and Szeliski's taxonomy and evaluation [12]. Here, we review stereo methods that use planar scene representations. Wang and Adelson [17] decompose images into multiple layers for motion analysis. They iteratively update the layers using affine motion analysis and clustering. They cluster based on flow, which can be inaccurate near occlusion boundaries. Baker *et al.* [1] used a layered scene representation with alpha for stereo reconstruction. They represent a stereo scene as a collection of planes with per-pixel depth offsets. They refine estimates of the plane equation and depth offset for each layer using an algorithm that accounts for occlusion and mixed pixels, but the initialization of the scene layers is not automatic. Tao *et al.* [15] present a method using color over-segmentation and a piecewise planar scene representation that inspired many other researchers [9, 3, 8, 24]. These methods perform well for reasons discussed earlier, but they all segment the input images in a pre-processing step and cannot recover from segmentation errors. Deng *et al.* [6] partially overcome this vulnerability by subdividing segments from one image using segment boundaries from the other, creating what they call ”patches”. Their method improves the stereo estimation, but because it updates the patches, not the segmentations, it is still vulnerable to initial segmentation errors.

With the exception of Baker *et al.* [1], the work mentioned above does not account for the partial opacity of mixed pixels on object boundaries. Much of the work in this area has concentrated on digital matting (extracting a foreground object from an image or video). For example, Chuang *et al.* [5] and Ruzon and Tomasi [11] propose matting methods that use user-defined trimaps. These trimaps specify three regions in the image: background, foreground, and the undefined area in which the algorithm must compute the foreground opacity and color. These are matting, not stereo, methods, so they do not compute depth. They require a user-defined trimap, and they assume the image has clearly separable foreground and background components.

Some researchers have proposed stereo or optical flow methods that explicitly account for alpha and are fully automatic. Zitnick *et al.* [24] propose a video view interpolation method that computes depth using segmentation-based stereo, uses the depth map to automatically create a trimap, and then computes opacity and foreground/background color information using Bayesian matting [5]. Hasinoff *et al.* [7] also propose a multi-view stereo method that refines a depth map by modeling occlusion boundaries as 3D curves. Both methods first compute a depth map with a single depth value per pixel, then refine the depth and compute matting information. As such, they are vulnerable to errors in the depth map computation, although Hasinoff *et al.* can overcome small errors.

Zitnick *et al.* [22] present an optical flow method that computes a consistent segmentation of two or more images in a sequence and also accounts for mixed pixels. Their method produces good optical flow results and has the advantage of updating the image segmentations, but it is not directly applicable to stereo because it does not handle occlusions or account for stereo constraints. Finally, Xiong and Jia propose a method for stereo matching on objects with fractional boundaries [19]. Their method uses stereo image pairs to produce very impressive matting results, but they present no quantitative evaluation of the accuracy of their depth maps. They also formulate alpha estimation as a matting problem, separating the entire scene into one background layer and one foreground layer. With this assumption of two layers, they can handle objects with very large fractional boundaries (i.e. very fuzzy or hairy items), which our method does not. However, they are limited to two depth layers, so the method is not suitable for general scenes, which may have objects evenly distributed across many depths (for example, the Cones data set in the Middlebury stereo evaluation [12]).

3. Scene Representation and Stereo Image Model

Figure 1 shows an overview of our algorithm. The input is two stereo images that are rectified or calibrated with respect to each other. One of these images is considered the reference view, and we represent the scene as a collection of fronto-parallel planar segments in that view's coordinate system. We use an over-segmentation approach and assume that all pixels in each segment have the same depth. Slanted planes are therefore approximated by a set of small segments. The key to our algorithm is alternately updating the shape and depth of these segments. We use a generative model of an image to update the segment shapes based on maximum a posteriori (MAP) estimation. We model stereo constraints as a pairwise Markov random field (MRF) and update the segment depths using belief propagation. The following subsections introduce these models, and section 4 describes our inference methods in detail.

3.1. A Generative Model of an Image for Updating Segment Shapes

We model an image as a set of potentially overlapping segments. Our generative model is inspired by Zitnick *et al.* [22]. In contrast to their work, however, we model stereo constraints. Moreover, we generate only one set of segments for the scene, instead of a segmentation of each input image.

Pixels in the reference image are mapped to segments by segment indices. To handle mixed pixels that commonly occur near segment boundaries, each pixel i is assigned to

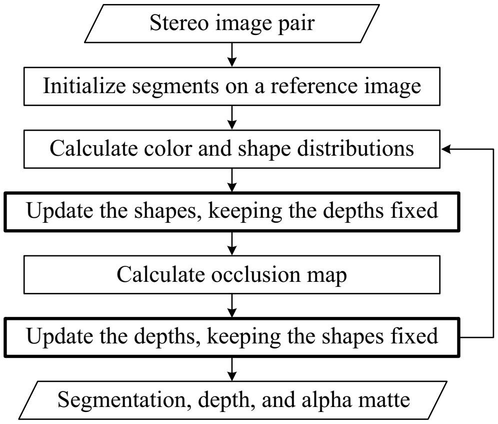

Stereo image pair
Initialize segments on a reference image
Calculate color and shape distributions
Update the shapes, keeping the depths fixed
Calculate occlusion map
Update the depths, keeping the shapes fixed
Segmentation, depth, and alpha matte

Figure 1. Overview of our algorithm.

two segment indices, s_i^f and s_i^b, representing the foreground and background components, respectively. Pixels that do not lie near segment boundaries are captured by the case $s_i^f = s_i^b$.

Each segment is modeled by its depth, color and shape. We assume each segment has a constant depth and its color is modeled by a Gaussian. A segment's spatial distribution is modeled using both a Gaussian and the set of pixels currently assigned as foreground to the segment. Thus, a segment s is described by the parameters

$$\boldsymbol{\Phi}_s = (d_s, \boldsymbol{\mu}_s, \boldsymbol{\Sigma}_s, \boldsymbol{\eta}_s, \boldsymbol{\Delta}_s, S_s), \tag{1}$$

where $(\boldsymbol{\mu}_s, \boldsymbol{\Sigma}_s)$ and $(\boldsymbol{\eta}_s, \boldsymbol{\Delta}_s)$ are mean and covariance matrix of the Gaussian distribution for the segment's color and shape, respectively, and d_s is the depth of the segment. S_s is a set of pixels over which segment s is believed to exist as foreground.

We express our generative model in a Bayesian framework and solve for the parameters using MAP estimation. Given the observed color $\boldsymbol{c}_i$ and position $\boldsymbol{x}_i$ of a pixel i, as well as the segment parameters $\boldsymbol{\Phi}$, we factorize the generative model as follows:

$$\begin{aligned} & p(\boldsymbol{c}_i, \boldsymbol{x}_i, \boldsymbol{c}_i^f, \boldsymbol{c}_i^b, \alpha_i, s_i^f, s_i^b | \boldsymbol{\Phi}) \propto \\ & \quad p(\boldsymbol{c}_i | \boldsymbol{c}_i^f, \boldsymbol{c}_i^b, \alpha_i)\, p(\boldsymbol{c}_i^f | s_i^f, \boldsymbol{\Phi})\, p(\boldsymbol{c}_i^b | s_i^b, \boldsymbol{\Phi})\, p(\alpha_i) \\ & \quad p(\boldsymbol{x}_i | s_i^f, \boldsymbol{\Phi})\, p(\boldsymbol{x}_i | s_i^b, \boldsymbol{\Phi})\, p(s_i^f)\, p(s_i^b)\,. \end{aligned} \tag{2}$$

We model the first factor of this equation by

$$p(\boldsymbol{c}_i | \boldsymbol{c}_i^f, \boldsymbol{c}_i^b, \alpha_i) = \mathcal{N}(\boldsymbol{c}_i; \alpha_i \boldsymbol{c}_i^f + (1 - \alpha_i)\boldsymbol{c}_i^b, \psi), \tag{3}$$

where $\mathcal{N}(\boldsymbol{x}; \boldsymbol{\mu}, \boldsymbol{\Sigma})$ is the normal distribution with mean $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$. This equation assumes the observed color of pixel i is generated by a noisy alpha-blending of the segment colors.

Given the segment indices s_i^f and s_i^b, the conditional distributions of the two hidden pixel colors $\boldsymbol{c}_i^f$ and $\boldsymbol{c}_i^b$ are computed using the segments' color models as

$$p(\boldsymbol{c}_i^f | s_i^f, \boldsymbol{\Phi}) = \mathcal{N}(\boldsymbol{c}_i^f; \boldsymbol{\mu}_{s_i^f}, \boldsymbol{\Sigma}_{s_i^f}), \tag{4}$$

and similarly for $\boldsymbol{c}_i^b$. The prior $p(\alpha_i)$ on α_i is set to be uniform and may be omitted.

The spatial likelihoods for a pixel i given segment indices s_i^f and s_i^b are split as

$$\begin{aligned} & p(\boldsymbol{x}_i | s_i^f, \boldsymbol{\Phi}) = \\ & \quad p(\boldsymbol{x}_i | s_i^f, \boldsymbol{\eta}_{s_i^f}, \boldsymbol{\Delta}_{s_i^f})\, p(\boldsymbol{x}_i | s_i^f, S_{s_i^f})\, p(\boldsymbol{x}_i | s_i^f, d_{s_i^f}) \end{aligned} \tag{5}$$

$$p(\boldsymbol{x}_i | s_i^b, \boldsymbol{\Phi}) = p(\boldsymbol{x}_i | s_i^b, \boldsymbol{\eta}_{s_i^b}, \boldsymbol{\Delta}_{s_i^b})\, p(\boldsymbol{x}_i | s_i^b, S_{s_i^b}). \tag{6}$$

The first factor $p(\boldsymbol{x}_i | s_i^f, \boldsymbol{\eta}_{s_i^f}, \boldsymbol{\Delta}_{s_i^f})$ is equal to the normal distribution $\mathcal{N}(\boldsymbol{x}_i; \boldsymbol{\eta}_{s_i^f}, \boldsymbol{\Delta}_{s_i^f})$, and similarly for $p(\boldsymbol{x}_i | s_i^b, \boldsymbol{\eta}_{s_i^b}, \boldsymbol{\Delta}_{s_i^b})$. The second factors enforce the constraint that segments should be locally coherent. This is accomplished by favoring segment assignments with strong local support. Specifically, we define them to be proportional to the number of pixels within a small neighborhood ε_i of $\boldsymbol{x}_i$:

$$p(\boldsymbol{x}_i | s_i^f, S_{s_i^f}) \propto \sum_{j \in \varepsilon_i} h(j, S_{s_i^f}) \tag{7}$$

$$p(\boldsymbol{x}_i | s_i^b, S_{s_i^b}) \propto \sum_{j \in \varepsilon_i} h(j, S_{s_i^b}). \tag{8}$$

The value of $h(j, S_s)$ is one if pixel j is a member of S_s and zero otherwise. Note that the background segment index is only influenced by the current assignment of the foreground segment index in the neighborhood ε_i. This constraint limits the extent of mixed pixels near the segment boundaries.

The final factor $p(\boldsymbol{x}_i | s_i^f, d_{s_i^f})$, only affected by the foreground segment index, ensures that the stereo matching cost for a pixel assigned to depth $d_{s_i^f}$ is small:

$$p(\boldsymbol{x}_i | s_i^f, d_{s_i^f}) \propto \exp(-C(i, d_{s_i^f})). \tag{9}$$

Here, $C(i, d_{s_i^f})$ is the matching cost described in detail in the next subsection. This formulation ensures that the pixel should belong to a segment whose estimated depth $d_{s_i^f}$ is likely given the depth probability distribution of the pixel.

We assume the segment priors are uniform. As a result $p(s_i^f)$ and $p(s_i^b) = \frac{1}{M}$, where M is the number of segments. Since $p(s_i^f)$ and $p(s_i^b)$ are uniform, they may be omitted when computing the MAP estimate.

3.2. Stereo Constraints for Updating Segment Depths

We model stereo and smoothness constraints using a pairwise MRF of segments, as shown in Fig. 2. Each node corresponds to a segment s and shares edges with neighboring segments t for $(s, t) \in N$, where N is the set of all adjacent segments. Here we define segments using only the foreground segment indices; i.e. the background segment indices are ignored in the depth update step. The state of each node is its corresponding segment's depth, so the number of states for each node is equal to the number of

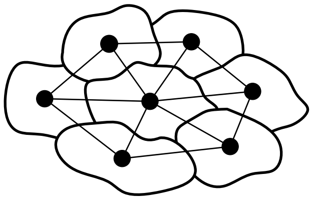


Figure 2. Stereo and smoothness constraints on the segment depths are modeled using a pairwise MRF, in which each segment corresponds to a node, and the nodes of neighboring segments are joined by edges.

depth levels. Zitnick *et al.* [23] use this formulation with a fixed segmentation. Our algorithm, by contrast, updates the topology of the MRF in each iteration using the updated segment shapes.

We measure the quality of a depth assignment for the MRF using the following energy function:

$$E = \sum_s D_s(d_s) + \sum_{(s,t)\in N} V(d_s, d_t). \qquad (10)$$

Here, $D_s(d_s)$ is the cost (commonly called the "data cost") of assigning a depth d_s to a segment s. This cost accounts for the matching of visible pixels between stereo views and a penalty for occluded regions. The term $V(d_s, d_t)$ is a cost (the "discontinuity cost") that penalizes depth assignments for neighboring segments s and t that violate a smoothness assumption.

3.2.1 Data Cost

We define the matching cost of a visible pixel i to be

$$C(i, d_s) = \rho_d(F(\boldsymbol{x}_i, d_s)). \qquad (11)$$

The function F measures the intensity similarity between the pixel in the reference image whose coordinate is $\boldsymbol{x}_i$ and the pixel projected to the other image with a depth value d_s. For this function, we use Birchfield and Tomasi's pixel dissimilarity measure [2], which is insensitive to image sampling. The function ρ_d is an error function that is robust to outliers due to noise, occlusions, specularities, and so on. We use a truncated L1 norm [14, 13]

$$\rho_d(x) = -\ln((1 - e_d)\exp(-|x|/\sigma_d) + e_d), \qquad (12)$$

where the parameters σ_d and e_d control the shape of the function.

Even with robust similarity measures, it is important to explicitly identify occluded pixels in the reference view so the algorithm does not match occluded regions in one view with pixels in the other. We incorporate an occlusion penalty in the data cost. We use a formulation that is similar to ones used in other segmentation-based stereo works [3, 18]. Before calculating the data cost for each iteration, we create an occlusion map by warping all of the pixels in the reference view to the non-reference view using currently estimated segment depths. The warped pixel depths (in the non-reference view coordinate system) are stored at the projected pixel coordinates. If more than one pixel from the reference view project to the same image coordinates in the non-reference views, they are sorted in depth order.

When calculating the data cost for each pixel, we project the pixel into the non-reference view and check its depth against the occlusion map. We distinguish the following three visibility cases: (a) the projected pixel is visible and occludes no other pixels; (b) the projected pixel is occluded by another pixel; and (c) the projected pixel is visible, but occludes another pixel. For each pixel, the data cost $\bar{C}(i, d_s)$ for each visibility case is given by

$$\bar{C}(i, d_s) = \begin{cases} C(i, d_s) & : \text{case (a)} \\ \lambda_{occ} & : \text{case (b)} \\ C(i, d_s) + \lambda_{occ} - C(j, d'_s) & : \text{case (c)} \end{cases} . \qquad (13)$$

For case (a), the pixel data cost is simply the matching cost. For case (b), an occluded pixel, the data cost is λ_{occ}, a positive constant that slightly penalizes occluded pixels. For case (c), the data cost favors low matching costs for the projected pixel, penalizes occlusions, and discourages occluding other pixels with low matching costs. $C(j, d'_s)$ is the matching cost of the occluded pixel j with (previously estimated) depth d'_s.

The data term of each segment is the sum of the matching costs of the pixels in the segment:

$$D_s(d_s) = \sum_{i\in s} \bar{C}(i, d_s). \qquad (14)$$

3.2.2 Discontinuity Cost

Like many other stereo methods, ours assumes that depth varies smoothly almost everywhere, except at object boundaries. We also assume that neighboring segments with similar colors are likely to have similar depths. Moreover, the larger the shared boundary between two segments, the stronger the discontinuity penalty should be. We express this discontinuity cost using a truncated L2 norm of depth difference of neighboring segments:

$$V(d_s, d_t) = \lambda_{disc}\, b_{st}\, \min((d_s - d_t)^2, T_{st}). \qquad (15)$$

The parameter λ_{disc} is a positive constant, b_{st} is the number of pixels on the boundary between segments s and t, and T_{st} is the truncation point for the L2 norm function. For each neighboring pair of segments, the truncation point is set such that pairs with large color differences have a small impact on the discontinuity cost, and pairs with small differences have a large impact. We use

$$T_{st} = \max(T_{max}\exp(-||\boldsymbol{\mu}_s - \boldsymbol{\mu}_t||^2/2\sigma_c^2),\, T_{min}), \qquad (16)$$

where $\boldsymbol{\mu}_s$ and $\boldsymbol{\mu}_t$ are the mean colors of segments s and t. The parameter σ_c controls the influence of the segments' color difference. T_{min} is chosen to be a small value to ensure that each segment has at least some influence on its neighboring segments.

4. Inference Procedure

This section describes the details of our inference process using the models introduced in the previous section. For the initial segmentation, we use a mean-shift segmentation method [4] with default parameters. The resulting large segments are partitioned into a grid of 8×8 pixels, because our method is based on over-segmentation. The initial depth of each segment is estimated using max-product belief propagation on our stereo MRF model. Because we have no occlusion information for this initial step, the message update order can strongly impact the inference. We use a synchronous update schedule [16], in which the messages are only updated at the end of each iteration, to ensure that the message update order does not affect the inference.

Next, we alternate between updating segment shapes and segment depths. To update segment shapes, we must find parameters which maximize the probability in Eq. (2) for each pixel i. To do this, we first choose candidate segment indices $(\hat{s}_i^f, \hat{s}_i^b)$ for the pixel based on the current estimate of segment assignments S and the constraints in Eqs. (7) and (8). The candidate segment indices can be any pair of two segments found within the neighborhood ε_i of pixel i. For each index pair $(\hat{s}_i^f, \hat{s}_i^b)$, we approximate alpha using the estimated segment colors $\boldsymbol{\mu}_s$ as

$$\hat{\alpha}_i = \frac{(\boldsymbol{c}_i - \boldsymbol{\mu}_{\hat{s}_i^b}) \cdot (\boldsymbol{\mu}_{\hat{s}_i^f} - \boldsymbol{\mu}_{\hat{s}_i^b})}{||\boldsymbol{\mu}_{\hat{s}_i^f} - \boldsymbol{\mu}_{\hat{s}_i^b}||^2}. \tag{17}$$

We use another approximation that the background color is same as the background segment color, i.e. $\hat{\boldsymbol{c}}_i^b = \boldsymbol{\mu}_{\hat{s}_i^b}$. Given $\hat{\boldsymbol{c}}_i^b$ and $\hat{\alpha}_i$, we can compute the foreground color using the alpha matting equation. Finally, we choose the parameters that maximize the probability of Eq. (2). For each iteration, to roughly estimate $\boldsymbol{\psi}$, the color noise covariance matrix, we compute and average the noise covariance matrices of all of the segments. If at any point a segment becomes too small (assigned to fewer than 12 pixels), it is discarded from the segmentation map, and the pixels within that segment are merged into the neighboring segments.

After each shape update step, we recalculate the occlusion map with the newly estimated segment depths. We then use belief propagation to update the segment depths. In contrast to the initial step, we now have occlusion information (based on previously estimated depths), so we can use an accelerated update schedule in which updated messages are immediately used to calculate the messages of neighboring segments. This scheme makes the inference fast, even with many segments. In our experiments, we only need two message propagation steps in each depth update step. We store the messages at the end of each depth update step, and use them for the initial messages in the next iteration. Whenever a new edge appears due to segment shape updates, the message for that edge is initialized to zero.

Table 1. Parameters used for the Middlebury evaluation.

ε_i	λ_{occ}	λ_{disc}	e_d	σ_d	σ_c	T_{max}	T_{min}
5×5	1.1	0.1	0.01	4.0	12.0	64.0	0.9

5. Experiments

In this section, we evaluate our method with the following experiments. First, we show the accuracy of our stereo algorithm using the new (second version) Middlebury stereo evaluation [12]. Next, we present the robustness of our adaptive segmentation method. Our method recovers from initial segmentation errors by updating segment shapes, and performs well for the Map image pair, which is known to be difficult for fixed segmentation methods. Finally, we show a Z-keying example to demonstrate the quality of our alpha matting results.

5.1. Stereo Reconstruction Accuracy

The error metric for the Middlebury stereo evaluation is the percentage of "bad pixels" (pixels for which the absolute disparity error is greater than 1 pixel) in the following three regions: non-occluded regions $(R_{\bar{O}})$, all regions except for unknown pixels (R_A), and regions near depth discontinuities (R_D). We used the same parameters, shown in Table 1, for all stereo pairs. We discretized the disparity space with an interval of 0.5 pixels, and performed 20 iterations of shape and depth updates. The running times were about 90 seconds for the Tsukuba data set (384×288 pixels, 31 depth levels) and 20 minutes for the Cones data set (450×375 pixels, 119 depth levels) on a 3.2 GHz PC.

Since the foreground and background segment indices for mixed pixels at segment boundaries differ, we have two different depth values for those pixels. The Middlebury evaluation, however, requires a single-valued depth map (one with one depth value per pixel). We use a threshold α_{th} to select a depth value for mixed pixels: for pixel i with $s_i^f \neq s_i^b$, if $\alpha_i \geq \alpha_{th}$ then select $d_{s_i^f}$, otherwise use $d_{s_i^b}$.

Table 2 summarizes the results of our method with two fixed thresholds (0.0 and 0.5) for all data sets, compared with the other state-of-the-art methods. Figure 3 shows depth maps and their "bad pixels" (shown as black for non-occluded regions and gray for occluded regions) using different alpha thresholds, for an inset from the Tsukuba image. The table and figure show that for the evaluation, the best threshold differs for different data sets. In particular, our results suggest that the Tsukuba depth map is biased toward foreground depth values for mixed pixels. This is confirmed by the insets in Fig. 3; the left Tsukuba input image more closely resembles our $(\alpha_{th} = 0.5)$ depth map than the $(\alpha_{th} = 0.0)$ one. Using a fixed alpha threshold value of 0.5 for all stereo pairs, the average rank of our algorithm is the fourth best in the Middlebury evaluation.

The final depth maps, segmentation maps, and alpha mattes for the Middlebury image pairs are shown in Fig. 4.

Table 2. Results on the new Middlebury stereo evaluation [12], comparing the percentage of "bad pixels" in non-occluded regions ($R_{\bar{O}}$), all regions except for unknown pixels (R_A), and regions near depth discontinuities (R_D). The best result in each column is in bold print. Subscript numbers for our method are the relative ranks in each column. The average rank of our algorithm is currently fourth best on the evaluation.

Algorithm	Tsukuba			Venus			Teddy			Cones		
	$R_{\bar{O}}$	R_A	R_D	$R_{\bar{O}}$	R_A	R_D	$R_{\bar{O}}$	R_A	R_D	$R_{\bar{O}}$	R_A	R_D
AdaptingBP [9]	1.11	1.37	5.79	**0.10**	0.21	1.44	4.22	7.06	11.8	**2.48**	7.92	**7.32**
DoubleBP [20]	**0.88**	**1.29**	**4.76**	0.14	0.60	2.00	3.55	8.71	**9.70**	2.90	9.24	7.80
SubPixDoubleBP [21]	1.24	1.76	5.98	0.12	0.46	1.74	**3.45**	8.38	10.0	2.93	8.73	7.91
Ours ($\alpha_{th} = 0.0$)	1.52 $_{16}$	1.93 $_{13}$	4.77 $_{2}$	0.11 $_{2}$	0.22 $_{2}$	**1.07** $_{1}$	7.10 $_{11}$	11.3 $_{6}$	16.6 $_{10}$	3.75 $_{10}$	9.21 $_{8}$	9.28 $_{11}$
Ours ($\alpha_{th} = 0.5$)	1.69 $_{17}$	2.04 $_{16}$	5.64 $_{4}$	0.14 $_{3}$	**0.20** $_{1}$	1.47 $_{2}$	7.04 $_{11}$	11.1 $_{6}$	16.4 $_{9}$	3.60 $_{9}$	8.96 $_{8}$	8.84 $_{9}$
SymBP+occ [13]	0.97	1.75	5.09	0.16	0.33	2.19	6.47	10.7	17.0	4.79	10.7	10.9
SO+borders [10]	1.29	1.71	6.83	0.25	0.53	2.26	7.02	12.2	16.3	3.90	9.85	10.2
Segm+visib [3]	1.30	1.57	6.92	0.79	1.06	6.76	5.00	**6.54**	12.3	3.72	8.62	10.2

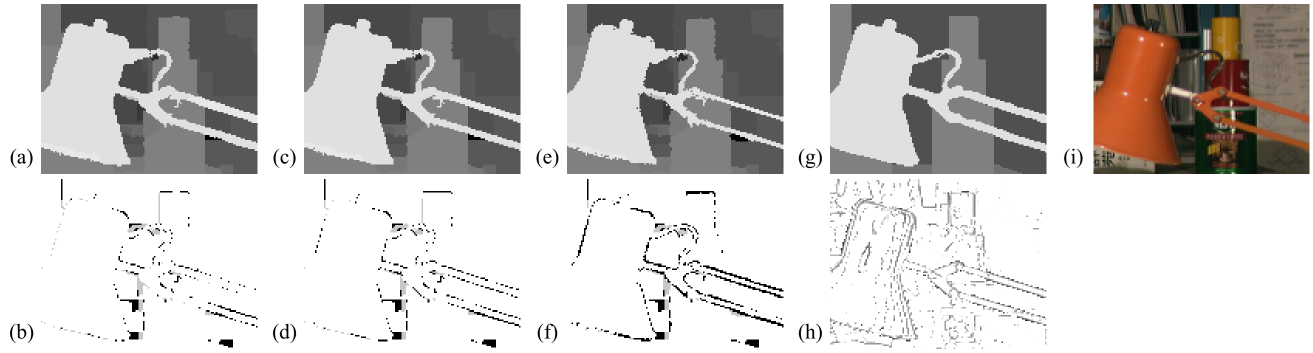


Figure 3. Depth maps and bad pixels for an inset of the Tsukuba data set, for different alpha thresholds: (a, b) $\alpha_{th} = 0.0$, (c, d) $\alpha_{th} = 0.5$, and (e, f) $\alpha_{th} = 1.0$. (g) Ground truth. (h) Estimated alpha matte. (i) Original left image. According to the evaluation ground truth data, best results are obtained with $\alpha_{th} = 0.0$. Although visually the ground truth depth map (g) matches our $\alpha_{th} = 0.0$ depth map (a), the actual left Tsukuba input image (i) seems to more closely resemble our $\alpha_{th} = 0.5$ depth map.

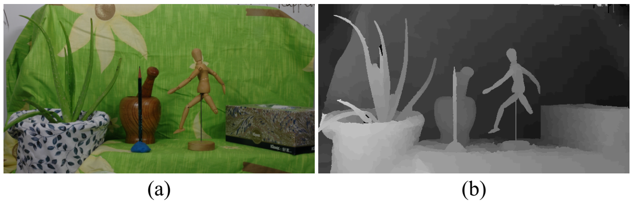


Figure 5. Result from Puppet image pair. (a) Original left image and (b) depth map ($\alpha_{th} = 0.5$).

Sharp object boundaries are recovered for all four data sets. Although slanted planes are approximated well with small segments of constant depth, our method fails for heavily slanted planes, such as the floor in the Teddy data set. Figure 5 depicts another result obtained using the Puppet image pair from [23], which is comparable to the results in [23].

5.2. Robustness of Adaptive Over-Segmentation

Figure 6 shows close-up views of the segmentation and depth maps at different iterations and demonstrates the robustness of our adaptive over-segmentation. The mean-shift segmentation method (with default parameters) [4] labels objects at different depths as one segment due to their similar colors (Fig. 6 (a)), causing errors for methods that use fixed segmentations. Our method, by contrast, recovers from these errors, producing better depth maps (Figs. 6 (b) and (c)).

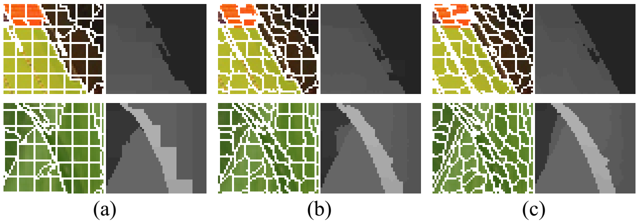


Figure 6. Close-up views (The right edge of the left plane in the Venus image (top) and the left leaf in the Puppet image (bottom)) of segmentation and depth maps at (a) initial step, (b) after 2 iterations, and (c) after 10 iterations. Our method recovers from initial segmentation errors, where objects at different depths are labeled as one segment, although there are still small regions that have wrong depth values.

Figure 7 shows stereo reconstruction results for the Map data set from the old Middlebury stereo evaluation. This data set is difficult for typical segmentation-based methods

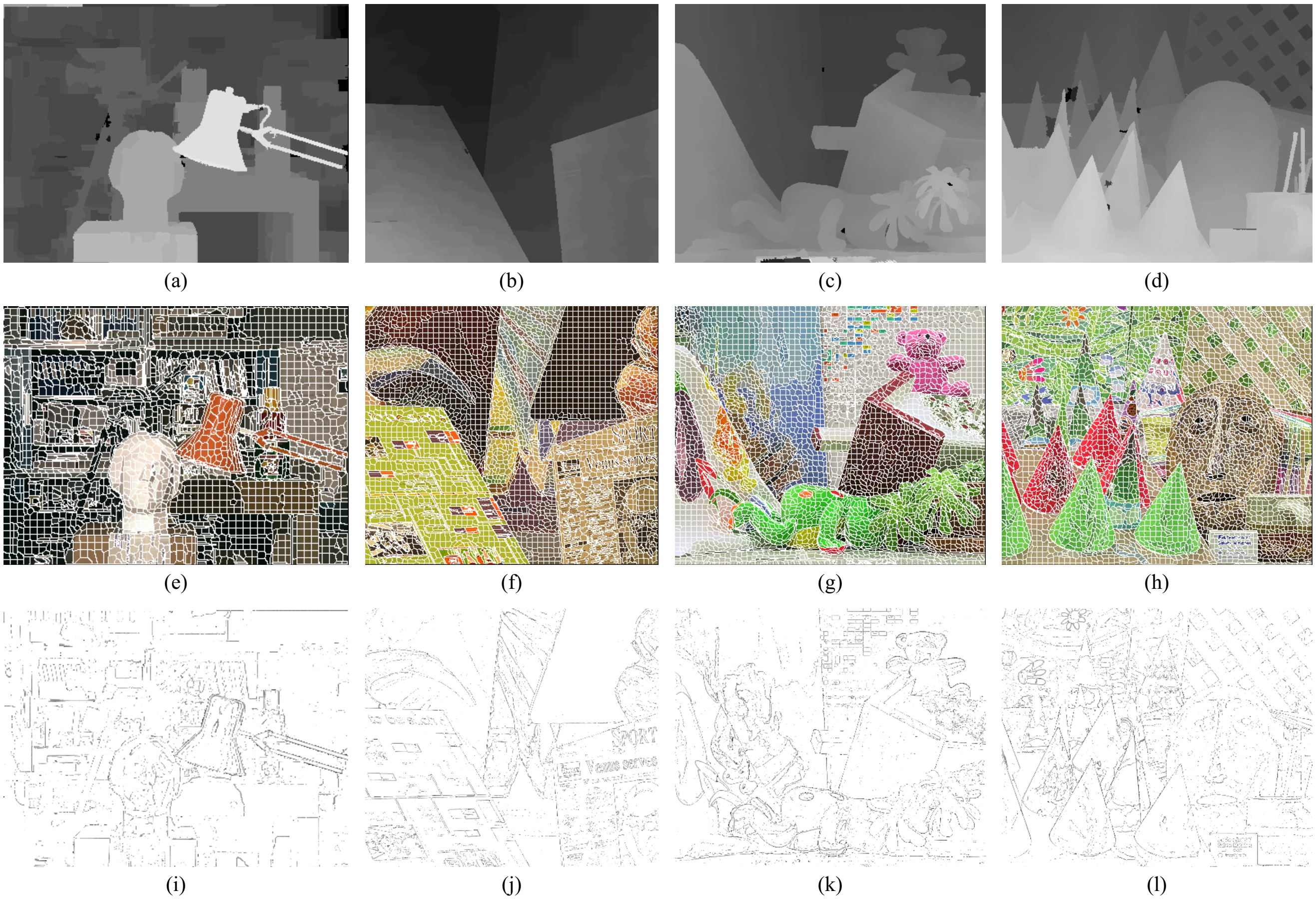


Figure 4. Output images. (a–d) Depth map ($\alpha_{th} = 0.5$), (e–h) segmentation map, and (i–l) alpha matte for each data set.

[8, 3, 18, 23], because color segmentation fails at object boundaries with similar foreground and background colors. For example, Fig. 7 (d) shows the results from Hong and Chen's method [8], a color-segmentation based algorithm that ranked third on the old Middlebury evaluation. Deng *et al.*'s patch-based approach [6] overcomes many of these errors, as shown in Fig. 7 (e). (Deng *et al.* fill occluded regions with neighboring depth values because these regions were not considered in the old Middlebury evaluation.) Our result is similar in quality to Deng *et al.*'s (Fig. 7 (f)). Currently, Sun *et al.* [13] obtain the best results for this image pair by using segmentation as a soft constraint (Fig. 7 (g)).

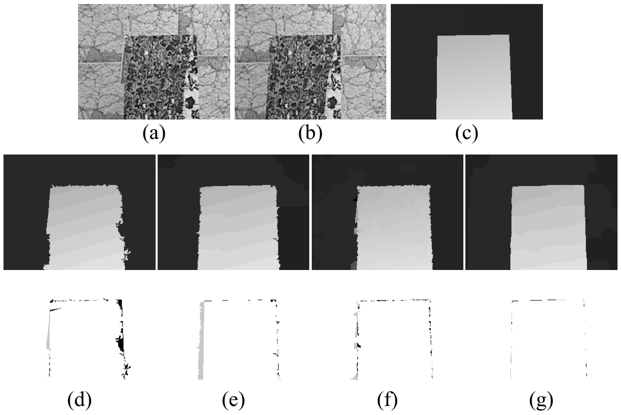
5.3. Z-Keying

Figure 8 shows a Z-keying result using estimated depth maps and alpha mattes for the Teddy and Cones image pairs. We extracted the teddy bear from the left Teddy image and composited it into the left Cones image. Because we use alpha mattes for both extraction and composition, there is no color bleeding on boundaries between the teddy bears and other objects (Figs. 8 (a) and (b)). By comparison, the matting results using a single depth map (calculated with $\alpha_{th} = 0.5$) and no alpha matte (Fig. 8 (c)) have artifacts.

6. Discussion and Conclusions

Our adaptive over-segmentation based stereo algorithm overcomes limitations of traditional segmentation based methods while properly handling mixed pixels on object boundaries. Our depth maps are not only accurate according to accepted standards (Middlebury) but in fact more complete, because we produce opacity information and foreground/background colors and depths for mixed pixels. In contrast to most matting methods, we produce this information along depth discontinuities throughout the scene, not only for foreground objects. Currently, the most significant limitation of our method is that it assumes a constant depth for all pixels in each segment, so it does not handle heavily slanted planes well. In future work, we could attempt to address this problem by using oriented planes or parametric surfaces instead of fronto-parallel segments.

To compare our stereo results with other researchers, we create single-valued depth maps to use with the Middlebury stereo evaluation. In doing so, we discovered that the Tsukuba ground truth depth map is biased toward the foreground depths of mixed pixels. Our performance on the Middlebury evaluation gives us good confidence in our


(a) (b) (c)

(d) (e) (f) (g)

Figure 7. Results for the Map image pair, known to be difficult for segmentation based stereo methods. (a) Left input image. (b) Right input image. (c) Ground truth depth map. (d–g) Computed depth maps (middle row) and their bad pixels (bottom row). (d) Hong and Chen [8]. (e) Deng *et al.* [6]. (f) Our method ($\alpha_{th} = 0.5$). (g) Sun *et al.* [13].

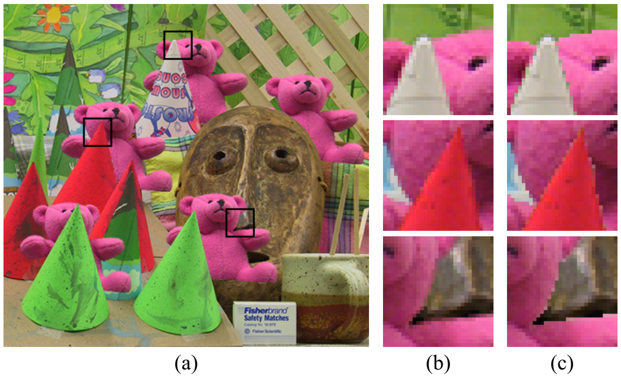

(a) (b) (c)

Figure 8. Z-keying example. (a) The teddy bear is extracted from the Teddy data set, and composited to the Cones data set. (b) Close-up views of rectangles in (a). (c) Close-up views of the result with a single depth map ($\alpha_{th} = 0.5$) and no alpha matte.

depth reconstruction, but it does not fully evaluate the quality of our matting results. Computing depth and matting information is clearly important for applications like view interpolation and Z-keying. In the future, we believe it would be useful to create a new stereo evaluation with ground truth opacities, and foreground/background colors and depths.

Acknowledgments: We would like to thank Sing Bing Kang for his helpful comments and Jian Sun for providing the Puppet image pair.

References

[1] S. Baker, R. Szeliski, and P. Anandan. A layered approach to stereo reconstruction. In *CVPR*, pages 434–441, 1998. 2

[2] S. Birchfield and C. Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Trans. PAMI*, 20(4):401–406, 1998. 4

[3] M. Bleyer and M. Gelautz. A layered stereo algorithm using image segmentation and global visibility constraints. In *ICIP*, pages 2997–3000, 2004. 1, 2, 4, 6, 7

[4] C. M. Christoudias, B. Georgescu, and P. Meer. Synergism in low level vision. In *ICPR*, volume 4, pages 150–155, 2002. 5, 6

[5] Y. Chuang, B. Curless, D. Salesin, and R. Szeliski. A bayesian approach to digital matting. In *CVPR*, volume II, pages 264–271, 2001. 1, 2

[6] Y. Deng, Q. Yang, X. Lin, and X. Tang. A symmetric patch-based correspondence model for occlusion handling. In *ICCV*, pages 1316–1322, 2005. 2, 7, 8

[7] S. W. Hasinoff, S. B. Kang, and R. Szeliski. Boundary matting for view synthesis. *CVIU*, 103(1):22–32, 2006. 1, 2

[8] L. Hong and G. Chen. Segment-based stereo matching using graph cuts. In *CVPR*, pages 74–81, 2004. 1, 2, 7, 8

[9] A. Klaus, M. Sormann, and K. Karner. Segment-based stereo matching using belief propagation and a self-adapting dissimilarity measure. In *ICPR*, pages 15–18, 2006. 1, 2, 6

[10] S. Mattoccia, F. Tombari, and L. D. Stefano. Stereo vision enabling precise border localization within a scanline optimization framework. In *ACCV*, pages 517–527, 2007. 6

[11] M. Ruzon and C. Tomasi. Alpha estimation in natural images. In *CVPR*, volume 1, pages 18–25, 2000. 1, 2

[12] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1):7–42, 2002. 2, 5, 6

[13] J. Sun, Y. Li, S. B. Kang, and H.-Y. Shum. Symmetric stereo matching for occlusion handling. In *CVPR*, volume 2, pages 399–406, 2005. 4, 6, 7, 8

[14] J. Sun, N.-N. Zheng, and H.-Y. Shum. Stereo matching using belief propagation. *IEEE Trans. PAMI*, 25(7):787–800, 2003. 4

[15] H. Tao, H. Sawhney, and R. Kumar. A global matching framework for stereo computation. In *ICCV*, pages 532–539, 2001. 1, 2

[16] M. F. Tappen and W. T. Freeman. Comparison of graph cuts with belief propagation for stereo, using identical MRF parameters. In *ICCV*, volume 2, pages 900–906, 2003. 5

[17] J. Wang and E. Adelson. Representing moving images with layers. *IEEE Trans. IP*, 3(5):625–638, 1994. 2

[18] Y. Wei and L. Quan. Region-based progressive stereo matching. In *CVPR*, volume 1, pages 106–113, 2004. 4, 7

[19] W. Xiong and J. Jia. Stereo matching on objects with fractional boundary. In *CVPR*, 2007. 2

[20] Q. Yang, L. Wang, R. Yang, H. Stewenius, and D. Nister. Stereo matching with color-weighted correlation, hierarchical belief propagation and occlusion handling. In *CVPR*, volume 2, pages 2347–2354, 2006. 6

[21] Q. Yang, R. Yang, J. Davis, and D. Nister. Spatial-depth super resolution for range images. In *CVPR*, 2007. 6

[22] C. L. Zitnick, N. Jojic, and S. B. Kang. Consistent segmentation for optical flow estimation. In *ICCV*, volume 2, pages 1308–1315, 2005. 2

[23] C. L. Zitnick and S. B. Kang. Stereo for image-based rendering using image over-segmentation. *IJCV*, 75(1):49–65, 2007. 4, 6, 7

[24] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski. High-quality video view interpolation using a layered representation. In *ACM SIGGRAPH*, pages 600–608, 2004. 1, 2