A Stereo Approach that Handles the Matting Problem via Image Warping

Michael Bleyer[1], Margrit Gelautz[1], Carsten Rother[2], Christoph Rhemann[1]
[1]Institute for Software Technology and Interactive Systems
Vienna University of Technology, Austria
[2]Microsoft Research Cambridge
Cambridge, UK

Abstract

We propose an algorithm that simultaneously extracts disparities and alpha matting information given a stereo image pair. Our method divides the reference image into a set of overlapping, partially transparent color segments. Each segment pixel is assigned an alpha value and color. The disparity inside the segment is modeled via a plane. The goodness of alphas, colors and disparity planes is measured by a new energy function. Its basic idea is to use the three parameters for generating artificial views representing the left and right images. If alphas, colors and disparity planes are correct, these artificial images should be very similar to the real ones. For generating the artificial right view, we warp all pixels of the left into the geometry of the right image using the disparity planes. We introduce the assumption of constant solidity in order to correctly model how pixels' alpha values are affected by the warping operation. Experimental results on the Middlebury set show that our algorithm gives good results in comparison to the state-of-the-art in stereo matching.

1. Introduction

Modern computer vision applications such as novel view generation or z-keying require high-quality disparity maps. For these applications, it is specifically important to produce precisely delineated disparity borders, which is traditionally difficult in stereo matching. Ideally, alpha values of border pixels should also be provided. Using the alpha information, one can blend transparent pixels against a novel background. This avoids border artifacts in the newly composed image that are extremely disturbing to the human eye.

Besides the requirements of these applications, synergies between the stereo and matting problems motivate a combined solution strategy. From a stereo perspective, mixed pixels are problematic, because they violate the color consistency assumption in regions close to disparity discontinuities. For example, consider the case where a red foreground pixel is mixed with a green background pixel in the left image. This red foreground pixel is mixed with a blue background in the right image. Due to the high dissimilarity of the mixed colors (yellow and magenta), it is likely that the disparity of the foreground pixel will be estimated wrongly. However, we can overcome this problem by decomposing the mixed color into its original color components and computing corresponding alpha values, i.e. solving the matting problem. From the matting perspective, disparity information is important, because it provides two instead of one mixed colors for computing an alpha value. This redundancy can be exploited to reduce matting ambiguities.

In this work, we propose an energy minimization approach for combined stereo and matting that exploits these synergies. Our algorithm partitions the left image into segments of homogeneous color. The segments are enlarged to include transparencies that typically occur close to segment borders. To model transparency, we assign each segment pixel an alpha value and a color. To model disparity, each segment is assigned to a disparity plane. We test the goodness of disparity planes and matting information via image warping. The disparity planes are thereby exploited to warp the overlapping segments into the geometry of the right view. The matting information is exploited to subsequently reconstruct the mixed color of each pixel in the warped view. To derive matching costs, this mixed color is compared against the pixel's real color taken from the right input image. By performing this reconstruction process of mixed colors, we can overcome the aforementioned color consistency problem. Our ideas are integrated in an energy function that we optimize by a combination of a greedy search procedure and Belief Propagation.

In the context of previous work, our method is related to various color segmentation-based stereo algorithms (e.g. [2, 5, 8, 17, 21]). We share their assumption that disparity inside a segment follows some particular disparity model (e.g. constant or planar). However, we relax their second assumption that the disparity borders coincide with segment borders, since we can alter a segment's shape by modifying its pixels' alpha values. There is a closer relationship to the methods of [2, 17] that both use image warping to generate an artificial right view. The major difference is that they cannot cope with alpha matte information.

Obviously, our method is also related to single-image matting approaches (e.g. [10, 12, 14, 18, 19]). In contrast

to these algorithms, we do not require user input and benefit from a second image. More specifically, our method is similar to [18] in the sense that the authors impose a simple pairwise smoothness prior on alpha and apply Belief Propagation for optimization. There is also some similarity to [11, 14] where the problem of composite colors that are the mixture of more than two pixels is addressed. Since such pixels occur when coping with more than two depth layers, our technique also handles this n-layer matting problem.

Although early work on combined stereo and matting dates back a decade [1, 15], there has been relatively little progress since then. The matting problem is sometimes solved in a post-processing step. For example, Zitnick et al. [21] first compute a disparity map and then apply a single-image matting technique on the detected disparity boundaries. However, this approach cannot succeed if the disparity discontinuities have been extracted erroneously. Hasinoff et al. [7] fit a 3D curve to precomputed disparity borders. The shape of this curve is then optimized to make it consistent with the stereo information, which enables recovery from small errors in the initial disparity map.

There are two papers that we consider closely related to our work. Xiong and Jia [20] propose a method that achieves disparity and matting results of high quality on complex fractional boundaries. The method exploits the problem synergies described above, but has the limitation that it needs to segment the image into a foreground and a background region. Matching and matting is then performed at the expanded border of these regions. However, in scenes containing many objects at different depths, such a two-layer segmentation hardly makes sense, which severely reduces the applicability of this approach. Taguchi et al. [16] propose a stereo method that works for multiple depth layers. They apply an adaptive color segmentation. A segment should thereby not be expanded to pixels that show high stereo matching costs. Pixel mattes are handled via a parametric model. However, the synergies between stereo and matting are not modeled. The authors neither use the matting information in the matching process nor take advantage of the stereo redundancy in the alpha computation. The problem of color inconsistencies is also ignored.

We make five contributions. First, we propose a formulation for combined stereo and matting that is applicable to scenes with an arbitrary number of depth layers. Despite this ability, our method still exploits the aforementioned problem synergies. Second, the idea of image warping [2, 17] is extended for handling transparent pixels. Third, we introduce the novel assumption of constant solidity that is used to convert alpha values between the stereo views and is more general than assumptions of previous approaches [20]. Fourth, our stereo formulation handles mixed colors that are composed of more than two pixels. Fifth, the combination of these ideas leads to a new energy function.

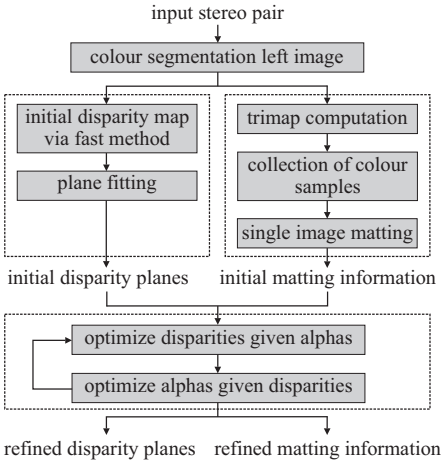


Figure 1. Overview of the algorithm.

2. Algorithm

Figure 1 shows a block diagram of the proposed algorithm. The algorithm takes left and right images as input. These images are assumed to be rectified so that correspondences lie on the same horizontal scanline. Our method provides a disparity map and matting information as output. We describe the individual steps in this section.

2.1. Initialization

The algorithm starts by dividing the left input image into non-overlapping segments of homogeneous color using mean-shift based color segmentation [4]. As is common in segmentation-based stereo, an oversegmentation is computed. The set of segments forms the starting point for generating initial disparity and matting solutions.

2.1.1 Initial disparity planes

We first compute a dense disparity map using the fast pixel-based dynamic programming method of [3]. The results of this method are of medium quality, but good enough to obtain reliable disparity models for the segments. We have chosen a planar disparity model that is able to cope with slanted surfaces. Although disparity planes may oversimplify the real surface shape, they work surprisingly well in practice, especially if used in conjunction with oversegmentation. The disparity d_p of a pixel p that belongs to segment S is computed using the segment's disparity model D_S:

$$d_p = D_S[a] \cdot x_p + D_S[b] \cdot y_p + D_S[c]. \quad (1)$$

Here, the x- and y-coordinates of p are denoted as x_p and y_p. The plane parameters a, b and c of D_S are derived by least-squared error fitting. We fit a disparity plane to all pixels of S from the initial disparity map. Since the least-squared error solution is sensitive to outliers, we perform the iterative plane fitting procedure described in [2].

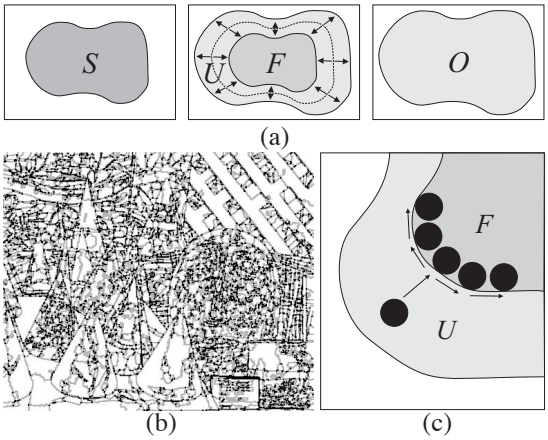


Figure 2. Initialization of matting information. (a) An overlapping segment O is computed by dilation of segment's S borderline. U and F represent the transparent and non-transparent regions of O, respectively. (b) Overlayed unknown regions. Gray values indicate that the unknown regions of 0 (white), 2 (gray) or more than 2 (black) segments overlap at this pixel. (A point of the unknown region that does not overlap with another segment's unknown region is deleted from its segment's unknown region.) (c) Collection of color samples from the non-transparent region.

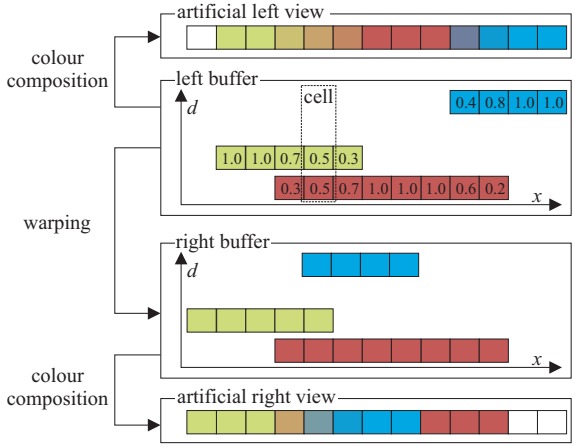


Figure 3. Basic idea. Our method estimates the pixels' alpha values and colors as well as disparity planes of segments. These parameters are optimized so that artificial left and right views become similar to the real left and right views. The axes represent x-coordinates (x) and disparities (d).

2.1.2 Initial matting information

To cope with mixed pixels that occur at segment boundaries, we generate overlapping, partially transparent segments. For each segment S, we extract its borderline to all neighboring segments. This borderline is then expanded using morphological dilation (figure 2a). The dilated borderline represents our unknown region U. We assume that transparent pixels only occur inside U. Note that this assumption is consistent with recent matting [12] and deblurring [9] approaches. These approaches suggest that in scenes with non-transparent objects, alpha is caused by convolving a camera's point spread function (PSF) with a hard (non-transparent) segmentation. The PSF models many effects like defocus blur and discretization artifacts. We use the same image formation process, which means that for non-transparent objects, alphas are *only* caused around the segment boundaries. We strongly believe that non-transparent objects are predominant for typical stereo applications.

In figure 2b, we overlay the unknown regions of all overlapping segments in the Cones test set [13]. One can see that pixels can lie in the unknown regions of three or more segments (black pixels in the figure). This makes sense, since a mixed pixel can be composed of colors coming from multiple (> 2) depth layers. Hence, as opposed to traditional matting algorithms that solve the two-layer matting problem with dedicated fore- and background, we handle the n-layer problem in our formulation.

As depicted in figure 1, our algorithm continues by collecting color samples from regions that are not affected by the matting problem [19]. We therefore compute the region $F = S \setminus U$ that is assumed to be the non-transparent part of the overlapping segment. For each pixel $u \in U$, we search its closest pixel in F (Euclidean distance). Starting from this pixel, we trace the boundary between U and F in both directions (figure 2c). The colors of visited pixels are stored as color samples for u. (In our implementation, we take 15 color samples.) If $F = \emptyset$, which can occur for very small segments, we just store a single color sample, i.e. the mean color of all pixels inside the corresponding non-overlapping segment S. We also store a color sample for each pixel f of the non-transparent region F. This color sample is the color of f in the left image.

Our method proceeds by computing initial matting information. To be more precise, by matting information, we mean an alpha value α_p and a color c_p for each pixel p of the overlapping segments. To avoid unnecessary overlap with section 2.2, we present only a rough outline of the initialization algorithm. Let S be a non-overlapping segment and O its corresponding overlapping segment derived by dilation. In the starting configuration, α_p of each pixel $p \in O$ is set to 1 if $p \in S$ and to 0 if $p \notin S$. The color c_p is set randomly to one of the color samples that have been collected for p. We then minimize an energy function that is the sum of equations (6), (7) and (9), using the optimization strategy of section 2.2.3.

2.2. Combined stereo and matting

2.2.1 Image warping with transparent pixels

Figure 3 illustrates the principle of our combined stereo and matting approach. We first generate a buffer representing the left image that stores all pixels of the overlapping segments. In the buffer, each pixel p has an alpha value α_p and a color c_p, which is represented by the numbers and differ-

ent colors in the figure. Multiple pixels can reside on the same image coordinate. In this context, we use the expression *cell* to refer to an array of pixels that share the same image coordinates. Given alpha values and colors, we can compute the mixed color of each cell. This leads to an *artificial left view*. If alpha vales and colors are correct, this artificial view should be very similar to the *real left view*.

In addition to the left buffer, we also generate a buffer representing the right image. The entries in the right buffer are generated by warping the pixels of the left buffer into the geometry of the right view. The disparity d_p of a pixel p is computed using equation (1) with the plane parameters taken from the overlapping segment to which p belongs. The new x-coordinate of p in the right buffer is then computed by $x_p - d_p$. In the warping process, we assume that the pixels' colors remain constant. However, it is invalid to assume that alpha values are constant, which is discussed in the following paragraphs. As before we can generate an *artificial right view*. If alpha values, colors *and disparity planes* are correct, this artificial image will be very similar to the *real right view*.

Let us now discuss how alpha values are affected by image warping. In [20], it is assumed that the alpha values of *foreground pixels* remain constant across views. However, this assumption does not provide information about the behavior of background pixels, which also need to be considered in the warping procedure. Moreover, as is shown later on, the assumption of constant foreground alpha does not necessarily hold if there are more than two disparity layers. To overcome these problems, we propose the use of a more general property, which we refer to as *solidity*. The solidity of a pixel p thereby denotes the *percentage to which p occludes pixels of lower disparities* that belong to the same cell. More formally, solidity o_p is computed in p's cell by

$$o_p = \frac{\alpha_p}{1 - \sum_{q:d_q > d_p} \alpha_q}. \quad (2)$$

In contrast to the assumption of constant foreground alpha, we make the reasonable assumption that solidity is constant across views *for all pixels*.

To compute the alpha of a pixel p in the right buffer, we first determine p's solidity o_p in its cell of the left buffer by equation (2). We then convert the solidity back to alpha using p's new cell in the right buffer. To perform this conversion, we compute the new alpha value α'_p in p's cell of the right buffer by

$$\alpha'_p = o_p(1 - \sum_{q:d_q > d_p} \alpha'_q). \quad (3)$$

Figure 4 shows three examples of this warping procedure. Here, figure 4b is of specific interest, since it shows that we can handle the occlusion problem without extra effort via image warping. An occlusion is detected if the alphas in a cell of the right buffer do not sum up to 1 (see

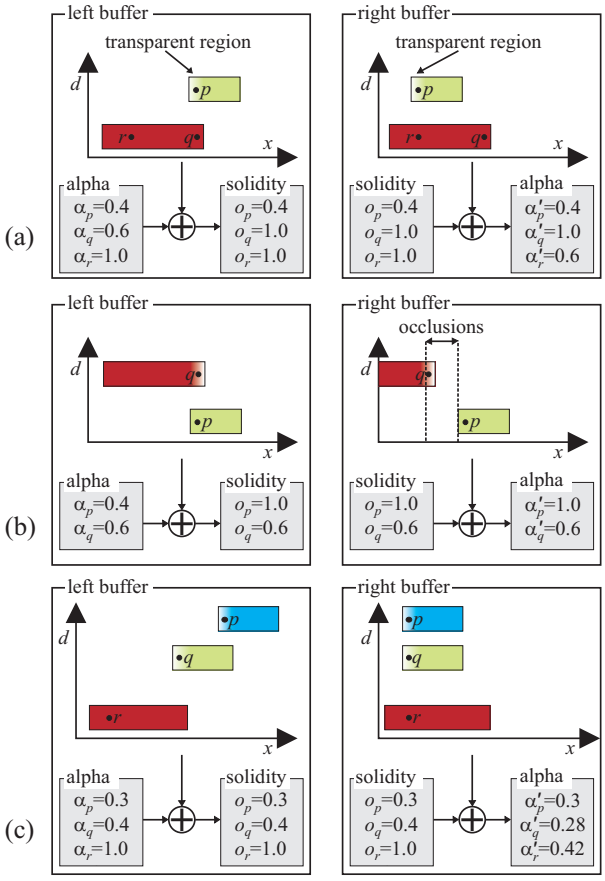


Figure 4. Examples of image warping with transparent pixels. In the left buffer, alpha is converted to solidity according to the current disparity solution. In the right buffer, solidity is converted back to alpha according to the new disparity configuration.

region between the dotted lines in figure 4b). Figure 4c shows a case where the assumption of constant foreground alphas is broken. Here, the pixel q is a foreground pixel in the reference image, but becomes partially occluded by the pixel p in the right image. Consequently, the alpha value of q is different across the views. Note that our assumption of constant solidity can tackle this case. Moreover, figure 4c shows that we can tackle cases in which more than two pixels project to the same coordinate in the right image.

2.2.2 Energy function

The goodness of alphas, colors and disparity planes is measured via an energy function whose individual terms are defined as follows. The overall energy E is computed as the sum of these terms (equations (6), (7), (8), (9) and (10)). However, let us first define two functions that prove to be useful in this context. The first function $asum()$ sums up the alpha values of all pixels in a cell at coordinate x:

$$asum(x) = \sum_{p:x_p = x} \alpha_p. \quad (4)$$

The second function $mc()$ computes the mixed color:

$$mc(x) = \frac{\sum_{p:x_p=x} \alpha_p \cdot c_p}{asum(x)}. \quad (5)$$

Here, the division serves to normalize the mixed color for those cells that are affected by occlusion, i.e. the cells for which $asum()$ is smaller than 1 (see figure 4b).

The first data term of our energy function operates on the left buffer. It generates the artificial left view by computing the mixed color in each cell and compares this artificial image against the real left view. The term E_l is defined by

$$E_l = \sum_{x \in \mathcal{X}_l} dis(mc(x), I_l(x)) \quad (6)$$

where $\mathcal{X}_l$ denotes all image coordinates of the left view and $I_l()$ returns the color of the real left image at given coordinate. The function $dis()$ computes the dissimilarity of two colors, which is implemented as the summed-up absolute differences of color channels in RGB color space.

For the left buffer, we strictly enforce that alpha values in a cell sum up to 1. This is implemented by the term E_{asum}:

$$E_{asum} = \sum_{x \in \mathcal{X}_l} \begin{cases} 0 & : \quad asum(x) = 1 \\ \infty & : \quad \text{otherwise.} \end{cases} \quad (7)$$

The second data term of our energy function operates on the right buffer that we generate as described in the previous section. It is important to understand that changing any of the parameters (alphas, colors or disparity planes) directly affects this term as this will lead to different artificial right views. The term E_r compares the artificial right image against the real one and is defined by

$$E_r = \sum_{x \in \mathcal{X}_r} asum(x) \cdot dis(mc(x), I_r(x)) + (1 - asum(x)) \cdot \lambda_{occ} \quad (8)$$

where $\mathcal{X}_r$ represents all coordinates in the right image and $I_r()$ returns the color in the right image. λ_{occ} is a user-defined constant penalty for occlusion. This penalty is needed to prevent the algorithm from simply maximizing the number of occluded pixels in the right buffer. Note that in equation (8) the influence of the dissimilarity function and λ_{occ} is balanced by $asum()$. The idea is that it might still be possible to match pixels that are only slightly affected by occlusion (e.g. $asum() = 0.9$) with high confidence. At pixels that are mostly or completely occluded, the occlusion penalty λ_{occ} dominates.

We expect that alpha varies smoothly within an overlapping segment.[1] This assumption is incorporated by a linear smoothness term $E_{asmooth}$ defined by

$$E_{asmooth} = \sum_{O \in \mathcal{O}} \sum_{<p,q> \in N_O} |\alpha_p - \alpha_q| \cdot \lambda_{asmooth} \quad (9)$$

where $\mathcal{O}$ is the set of overlapping segments. N_O represents all pairs of spatially neighboring pixels in the overlapping segment O. For simplicity, we have chosen 4-connectivity. $\lambda_{asmooth}$ is a user-defined parameter.

We also apply a smoothness prior on the disparity planes. The term $E_{dsmooth}$ imposes a penalty if the disparity planes of two neighboring segments are different:

$$E_{dsmooth} = \sum_{<S,V> \in N'} T[D_S \neq D_V] \cdot brd(S,V) \cdot \lambda_{dsmooth}. \quad (10)$$

Here, N' represents all pairs of neighboring segments. The function $brd()$ computes the number of pixels that lie on the common border of two segments in 8-connectivity. Note that since neighborhood and border length are not straightforward to define for fuzzy segments, we operate on the set of non-overlapping segments here. $T[]$ returns 1 if two disparity planes are different and 0 otherwise. Finally, $\lambda_{dsmooth}$ is a constant user-defined penalty.

2.2.3 Energy optimization

We start the optimization process using the alphas, colors and disparity planes computed in section 2.1 and initialize our left and right buffers accordingly. To minimize the proposed energy, we apply a two-step procedure. In the first step, we keep alphas and colors constant and optimize the disparity planes only. In the second step, we fix the disparity planes and optimize alphas and colors. These two steps are iterated until no further improvement of the energy is achieved. In our experiments, we use three iterations.

Optimization of disparities given alphas and colors Our optimization strategy works by altering the disparity planes of single overlapping segments, which is similar to the approaches taken in [2, 17]. For each overlapping segment O, we test a candidate plane set composed in the following manner. First, we add the segment's O original plane, i.e. the disparity model derived from the plane fitting step of section 2.1.1. This ensures that we do not "lose" planes in the optimization process. Second, we add the disparity planes of all neighboring segments of O, because it is likely that a correct disparity plane can be propagated from a spatial neighbor. Third, for each disparity from 0 to the maximum allowed disparity value, we add a frontoparallel plane. This allows us to "fall back" to a constant disparity model, if the correct disparity plane has not been found in the initialization. For each plane of the candidate set, we evaluate our energy function or, more precisely, the terms (8) and (10). We store the plane that gives the largest energy improvement. If the energy using this "best" plane is

[1] This is also motivated by our previous discussion that alpha is caused by the camera's PSF (highly smooth). It could be worth to consider other recent cost functions (e.g. [10]) in future work.

lower than the energy of O's current disparity plane assignment, we accept the new plane immediately and modify left and right buffers accordingly.

We have noticed that this immediate update strategy leads to considerably lower energies than the delayed update of [2, 17]. To determine the order in which overlapping segments are processed, we proceed across the image pixels in row-major order and look-up the segment at the current pixel. If the segment has not been tested already, it is processed now. We also apply a backward pass where the segment order is reversed. Forward and backward passes are iterated until there is no further energy improvement.

Optimization of alphas and colors given disparities For each overlapping segment, we apply the optimization procedure described in the next two paragraphs. We loop through the set of overlapping segments a few times.

Now we describe how the optimization for a single overlapping segment O is carried out. We start by explaining how the costs $e_{\alpha_p=\alpha,c_p=c}$ for assigning a pixel $p \in O$ to alpha value α and color c are computed. After setting $\alpha_p = \alpha$ and $c_p = c$, the alpha values of the remaining pixels in p's cell are scaled so that the cell's sum of alphas is equal to 1 (equation (7)). We compute the value v_l as the color difference between p's real left color and the mixed pixel of p's modified cell (equation (6)). For each pixel q in $p's$ cell (including p itself), we look up its matching point q' in the right buffer. We update the cell of q', compute the modified mixed color and calculate the color difference to the real right color (equation (8)). The value v_r is then derived as the sum of these color differences computed for all qs in p's cell. The costs $e_{\alpha_p=\alpha,c_p=c}$ are finally computed by $v_l + v_r$.

Given pixel p and its alpha value α, we can calculate the color sample of lowest costs $c^*_{\alpha_p=\alpha}$ by $\arg\min_{c\in C_p} e_{\alpha_p=\alpha,c_p=c}$ with C_p representing the color samples collected for p. Using the best color sample, the costs $e'_{\alpha_p=\alpha}$ for assigning p to α are derived by $\min_{c\in C_p} e_{\alpha_p=\alpha,c_p=c}$. To compute the final alpha value of pixel p, we have to consider the smoothness constraint on alpha values imposed by equation (9). We construct the set $\mathcal{L}$ that consists of 100 discretized alpha values. For each pixel $p \in O$ and each $\alpha \in \mathcal{L}$, we compute $e'_{\alpha_p=\alpha}$. These values represent the data costs of a Markov Random Field that we optimize using the computationally efficient Belief Propagation implementation of [6]. Now that we know the final alpha value α' of p, we update the alpha values in the affected cells of left and right buffers accordingly. The pixel's p new color is thereby set to $c^*_{\alpha_p=\alpha'}$.

3. Results

We evaluate our method using the Middlebury benchmark [13]. The parameters in equations (8), (9) and (10) are set to the constant values of $\lambda_{occ} = 30$, $\lambda_{asmooth} = 0.2$ and $\lambda_{dsmooth} = 7.5$. Our unoptimized implementation takes approximately 10 minutes to process a Middlebury pair.

Results are presented in figure 5. Our method assigns multiple disparity values to a single image coordinate. For obtaining the single-valued disparity maps of figure 5a, we extract the pixel of highest alpha value in each cell of the left buffer and plot its disparity. It can be inferred from the error images in figure 5b that the computed disparity maps are of high quality. Our algorithm performs particularly well in the reconstruction of disparity borders where we can benefit from matting information. Our method can also reconstruct fine details such as the cable of the lamp in the Tsukuba set.

For visualizing the alpha information, we use two alternative methods. First, we plot the largest alpha value of each cell in figure 5c. Second, we show the assignments of disparity planes in figure 5d. Each disparity plane is thereby given a color. We plot the mixture of plane colors using our alpha values. (This result can also be interpreted as a "soft" depth segmentation.) Finally, figure 5e shows the artificial right views our method has produced in the final iteration. Since our algorithm uses the matting information in the warping procedure, we can fuse transparent foreground pixels with their novel background. In contrast to conventional stereo methods (including [16]), we can therefore avoid the color consistency problem near disparity borders.

For quantitative analysis of our results, we use the Middlebury online table. We show the results for some selected algorithms in table 1. Our method (*WarpMat*) is one of the top performers and takes rank 6 of 56 methods at the time of submission. It performs better than its competitor [16] on three of the four image pairs and is consequently assigned a better rank. We have also tested our method in conjunction with using hard segmentation (*HardSeg* in the table). We therefore operate on the pixels of the non-overlapping segments. The pixels' alpha values are set to 1 and their colors are taken from the original left image. This makes our method (and the results thereof) very similar to [2]. As is depicted from the table, this strategy performs worse than *WarpMat* on each of the test pairs. Figure 6 compares the results of *WarpMat* and *HardSeg*.

A first and natural application of our method is novel viewpoint generation. We can apply our warping procedure in combination with "scaled" disparity planes to compute views that lie in-between or beyond the input views. This is demonstrated for the Tsukuba set in figure 7. As a second application, we use our results to segment objects based on their computed disparity in figure 8. This figure also serves to give an impression about the good quality of our matting results, because quantitative evaluation is difficult in the absence of ground truth matting data for the Middlebury set.

4. Conclusions

We have presented a combined stereo and matting algorithm, which takes advantage of synergies between the two

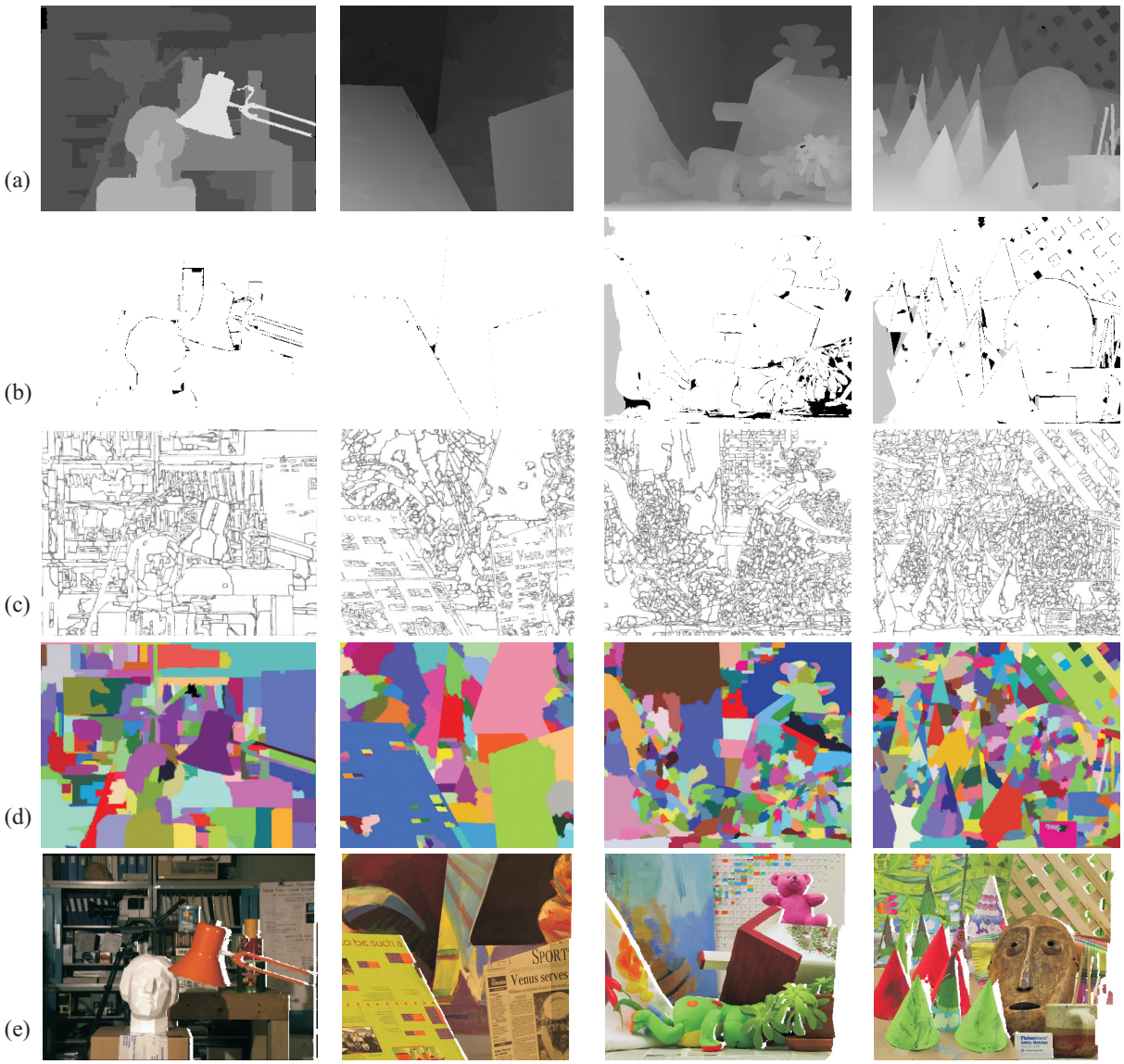
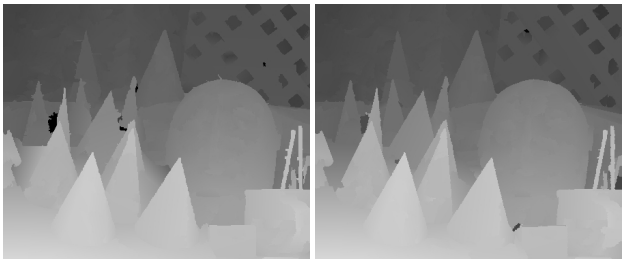


Figure 5. Results of our algorithm on the Tsukuba, Venus, Teddy and Cones image pairs from the Middlebury data set. (a) Computed disparity maps. (b) Absolute disparity errors larger than one pixel. Black pixels denote errors in visible regions, while gray pixels are errors in occluded regions. (c) Computed alpha values. (d) Assignment of pixels to disparity planes. Mixed colors represent pixels that are assigned to more than one disparity plane. (e) Artificial right views that have led to these results. Occlusions are given white color.

Algorithm	Rank	Avg. Error	Tsukuba			Venus			Teddy			Cones		
			nocc	all	disc	nocc	all	disc	nocc	all	disc	nocc	all	disc
AdaptingBP	1	4.23	1.11	1.37	5.79	0.10	0.21	1.44	4.22	7.06	11.8	2.48	7.92	7.32
SubPixBP	5	4.39	1.24	1.76	5.98	0.12	0.46	1.74	3.45	8.38	10.0	2.93	8.73	7.91
WarpMat	**6**	**4.98**	**1.16**	**1.35**	**6.04**	**0.18**	**0.24**	**2.44**	**5.02**	**9.30**	**13.0**	**3.49**	**8.47**	**9.01**
AdaptOvrSeg [16]	8	5.59	1.69	2.04	5.64	0.14	0.20	1.47	7.04	11.1	16.4	3.60	8.96	8.84
HardSeg	**11**	**5.53**	**1.20**	**1.54**	**6.07**	**0.55**	**0.64**	**5.10**	**5.50**	**9.73**	**13.5**	**3.83**	**8.66**	**10.01**
Segm+visib [2]	15	5.40	1.30	1.57	6.92	0.79	1.06	6.76	5.00	6.54	12.3	3.72	8.62	10.2

Table 1. Middlebury rankings. Our method denoted by *WarpMat* takes the sixth rank of 56 evaluated stereo algorithms. Values in the table represent error percentages measured in different image regions.



(a) (b)

Figure 6. Soft versus hard segmentation. (a) Result of using hard segmentation (*HardSeg*). (b) Our result (*WarpMat*). An improvement of disparity borders is visible.

(a) (b) (c)

Figure 7. Novel view generation. (a) Novel view generated using our disparity and matting results. (b) Due to using our matting information, the lamp naturally melts with its new background. (c) Our disparity results used in combination with hard segmentation. Disturbing artifacts at the lamp's border occur.

Figure 8. "Soft" depth segmentation. We segment objects by specifying a minimum and maximum disparity value. Pixels that fall in the disparity range are extracted. The computed objects are pasted against a white background using our matting information.

problems. Our method extends previous warping-based approaches by handling transparent pixels. We have proposed the assumption of constant solidity for transforming transparent pixels to the second view. Our approach can handle mixed colors that are the composite of pixels from more than two depth layers. The high quality of our disparity results is demonstrated by a sixth rank in the Middlebury Online Table. We have exploited our results for novel view interpolation and depth segmentation.

In future work, our formulation should be extended to handle the input images symmetrically. Currently, the results are different depending on which stereo image is used as a reference. While our method handles segmentation errors that lie inside the unknown regions of segments, it cannot handle very large segmentation errors. This problem can be overcome by iteratively updating the unknown regions.

Acknowledgments

Michael Bleyer is financed by the Austrian Science Fund (FWF) under project P19797. Christoph Rhemann is supported by Microsoft Research Cambridge through its PhD Scholarship Program. The authors thank Alex Rav-Acha for initial discussions and Tom Wilson for proofreading.

References

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

[2] M. Bleyer and M. Gelautz. A layered stereo matching algorithm using image segmentation and global visibility constraints. *ISPRS Journal*, 59(3):128–150, 2005.

[3] M. Bleyer and M. Gelautz. Simple but effective tree structures for dynamic programming-based stereo matching. In *VISAPP*, volume 2, pages 415–422, 2008.

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

[5] Y. Deng, Q. Yang, X. Lin, and X. Tang. A symmetric patch-based correspondence model for occlusion handling. In *ICCV*, pages 542–567, 2005.

[6] P. Felzenszwalb and D. Huttenlocher. Efficient belief propagation for early vision. *IJCV*, 70(1), 2006.

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

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

[9] J. Jia. Single image motion deblurring using transparency. In *CVPR*, 2007.

[10] A. Levin, D. Lischinski, and Y. Weiss. A closed-form solution to natural image matting. *TPAMI*, 30(2):228–242, 2008.

[11] A. Levin, A. Rav-Acha, and D. Lischinski. Spectral matting. *TPAMI*, 30(10):1699–1712, 2008.

[12] C. Rhemann, C. Rother, A. Rav-Acha, and T. Sharp. High resolution matting via interactive trimap segmentation. In *CVPR*, 2008.

[13] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1/2/3):7–42, 2002. http://vision.middlebury.edu/stereo/.

[14] D. Singaraju and R. Vidal. Interactive image matting for multiple layers. In *CVPR*, pages 1–7, 2008.

[15] R. Szeliski and P. Golland. Stereo matching with transparency and matting. In *ICCV*, pages 517–525, 1998.

[16] Y. Taguchi, B. Wilburn, and L. Zitnick. Stereo reconstruction with mixed pixels using adaptive over-segmentation. In *CVPR*, pages 1–8, 2008.

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

[18] J. Wang and M. Cohen. An iterative optimization approach for unified image segmentation and matting. In *ICCV*, volume 2, pages 936–943, 2005.

[19] J. Wang and M. Cohen. Optimized color sampling for robust matting. In *CVPR*, pages 1–8, 2007.

[20] W. Xiong and J. Jia. Stereo matching on objects with fractional boundary. In *CVPR*, pages 1–8, 2007.

[21] L. Zitnick, S. Kang, M. Uyttendaele, S. Winder, and R. Szeliski. High-quality video view interpolation using a layered representation. *ACM Transaction on Graphics*, 23(3):600–608, 2004.