Graph-based surface reconstruction from stereo pairs using image segmentation

Michael Bleyer and Margrit Gelautz

Institute for Software Technology and Interactive Systems
Vienna University of Technology
Favoritenstrasse 9-11/188/2, A-1040 Vienna, Austria

ABSTRACT

This paper describes a novel stereo matching algorithm for epipolar rectified images. The method applies colour segmentation on the reference image. The use of segmentation makes the algorithm capable of handling large untextured regions, estimating precise depth boundaries and propagating disparity information to occluded regions, which are challenging tasks for conventional stereo methods. We model disparity inside a segment by a planar equation. Initial disparity segments are clustered to form a set of disparity layers, which are planar surfaces that are likely to occur in the scene. Assignments of segments to disparity layers are then derived by minimization of a global cost function via a robust optimization technique that employs graph cuts. The cost function is defined on the pixel level, as well as on the segment level. While the pixel level measures the data similarity based on the current disparity map and detects occlusions symmetrically in both views, the segment level propagates the segmentation information and incorporates a smoothness term. New planar models are then generated based on the disparity layers' spatial extents. Results obtained for benchmark and self-recorded image pairs indicate that the proposed method is able to compete with the best-performing state-of-the-art algorithms.

Keywords: stereo matching algorithm, image segmentation, graph cuts;

1. INTRODUCTION

Given two images that are recorded from slightly different views, a stereo matching algorithm tries to identify corresponding points in both images that refer to the same scene point. Once these correspondences are known, the world coordinates of each image point can be reconstructed by triangulation. To simplify the search for correspondences, the image pair is commonly transformed into epipolar geometry, so that the stereo problem is reduced to a one-dimensional search along corresponding scanlines. The offset between x-coordinates in the left and right images is then referred to as *disparity*. However, assigning each point to its correct disparity proves to be difficult. Common stereo algorithms show weak performance in occluded regions, as well as in regions of low texture. Furthermore, extracting precise depth boundaries presents a challenging task. In our work, we propose a method that tries to overcome those problems by the use of image segmentation. We thereby embed two basic assumptions. We assume that disparity inside each colour segment varies smoothly and depth discontinuities coincide with segment borders. The stereo problem is then formulated as an optimization task, so that each segment is assigned to a planar disparity model in order to minimize a global cost function. A strong local optimum of this cost function is then computed by a robust optimization technique that uses graph cuts. In the following, we give an overview of the previous work that we consider most relevant to our approach. For an extensive survey on stereo algorithms, the reader is referred to Scharstein and Szeliski.[1]

Recently, several papers on stereo methods that take advantage of colour segmentation were published. Tao and Sawhney[2] present a stereo algorithm that models each segment's disparity by a planar equation. Plane models are thereby propagated among neighbouring segments in a hypothesis testing framework. The quality of the resulting disparity map is then measured by warping the reference image to the second view. The basic idea behind this procedure is that if the disparity map was correct, the warped view should be very similar to

Further author information:
Michael Bleyer: E-mail: bleyer@ims.tuwien.ac.at, Telephone: +43-1-58801-18865
Margrit Gelautz: E-mail: gelautz@ims.tuwien.ac.at, Telephone: +43-1-58801-18849

the real second view. Inspired by those ideas, Bleyer and Gelautz[3] present an algorithm that employs a layered model. Planar disparity segments are thereby clustered to form a single robust layer in the layer extraction step of the method. Each segment's disparity is then described by one of the extracted disparity layers. The assignment of segments to layers is optimized in a layer assignment step in order to optimize a cost function that also uses image warping. A Z-buffer is thereby employed to detect occlusions in both views and to make decisions concerning visibility. Furthermore, the cost function models smoothness across segments. The layer extraction and assignment steps are then iterated, so that new layer models are generated throughout the whole process. However, the greedy algorithm that is employed for optimization in the layer assignment step is likely to fall into a "weak" optimum due to its local nature. Wei and Quan[4] use the results of segmentation in a progressive framework to avoid the computational costs of global optimization. Reliable regions are identified and used to constrain subsequent matches via visibility and smoothness constraints.

Graph-based optimization techniques have been shown to give strong results on various computer vision tasks, including image restoration, segmentation, stereo and motion. The stereo algorithm of Boykov et al.[5] applies graph cuts to optimize a cost function consisting of a data term that measures the pixel dissimilarity and a smoothness term that puts a constant penalty on neighbouring pixels assigned to different disparities. This work was extended by Kolmogorov and Zabih[6] to handle occlusions by enforcing the uniqueness constraint. Since the smoothness term of these approaches aims at generating piecewise constant disparities, the reconstruction of slanted surfaces might not be optimal. Birchfield and Tomasi[7] therefore represent the scene by a set of planar layers, so that each pixel is assigned to a plane model instead of a discrete disparity value. Given a set of planes, the algorithm assigns each pixel of the reference view to exactly one of those layers by the use of graph cuts. The plane models are then refined based on their spatial extents found in the layer assignment step. Similarly to our approach, those two steps are iterated until convergence. Lin and Tomasi[8] extend this approach to handle occlusions symmetrically. Furthermore, they apply a spline model to describe surfaces' disparities.

Recently, Hong and Chen[9] proposed a stereo algorithm that combines image segmentation with graph-based optimization. Instead of operating on the pixel level, the stereo problem is formulated on the segment level. Each segment is thereby assigned to a plane model. Graph-based minimization of a cost function that consists of a data and a smoothness term is then applied to optimize those assignments. However, it is hard to handle occlusions on the segment level and the authors therefore try to identify occluded pixels before the optimization process. In contrast to the work of Hong and Chen, in our approach we embed the reasoning about occlusion into the optimization algorithm. We therefore operate on the pixel, as well as on the segment level. Occlusions are detected symmetrically in both images on the pixel level, while the segmentation information is propagated from the segment level.

2. ALGORITHMIC OVERVIEW

The overall algorithm can be divided into several major steps as shown in figure 1. In the first step, the reference (left) image is segmented into regions of homogeneous colour. We assume that disparity inside such segments varies smoothly and depth discontinuities coincide with segment borders. The disparity inside a segment is modelled by a planar equation. An initial disparity map is then computed by a window-based method and serves to initialize each segment's planar model.

Disparity estimates in small segments tend to be unreliable due to the small spatial extent over which their planar models were computed. In the *layer extraction* step of the algorithm (see figure 1), we therefore try to identify groups of segments that can be well approximated by the same plane equation. We refer to these groups as *disparity layers*. Such layers are extracted from the segments' original plane equations using a clustering method. A layer's robust plane model is then derived by fitting a plane over the larger region formed by all segments belonging to that particular layer. The disparity layers can be seen as a set of planar surfaces that are likely to occur in the scene. Their number is usually much lower than the number of segments.

Once the disparity layers are known, we assign each segment, as well as each pixel of both views (except occluded pixels, see below) to exactly one of those layers. This is done in the *layer assignment* step of the algorithm. Occlusions are modelled in both views by allowing pixels to be assigned to no plane at all. The assignment of segments influences the assignment of pixels and vice versa. However, although we also operate on

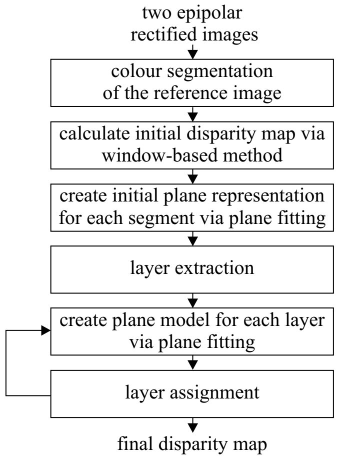


Figure 1. Algorithmic outline.

the pixel level, the final output of this step is the assignment of each segment to exactly one disparity layer. A cost function measures the quality of the current assignment. We search for a local minimum of the cost function by using a robust graph-based optimization technique.

Information about each disparity layer's spatial extent is then used to generate new planar layers. For each disparity layer that is present in the computed solution, a plane is fitted over all segments that belong to this particular layer. The layer assignment step is then invoked again to test whether any of those new layers can decrease the solution's costs. If this is the case, the procedure is iterated. Otherwise, the final disparity map is generated using the current assignment of segments to disparity layers.

3. COMPUTATION OF DISPARITY LAYERS

The following steps are discussed in detail by Bleyer and Gelautz[10] and briefly reviewed in this section.

3.1. Colour segmentation

We assume that for regions of homogeneous colour the disparity varies smoothly and depth discontinuities coincide with the boundaries of those regions, which holds true for most natural scenes. To ensure that this assumption is met, it is safer to oversegment the image. In our implementation, the mean-shift-based colour segmentation algorithm proposed by Christoudias et al.[11] is employed. We show the results of colour segmentation in figure 2b, where pixels of the same colour belong to the same segment.

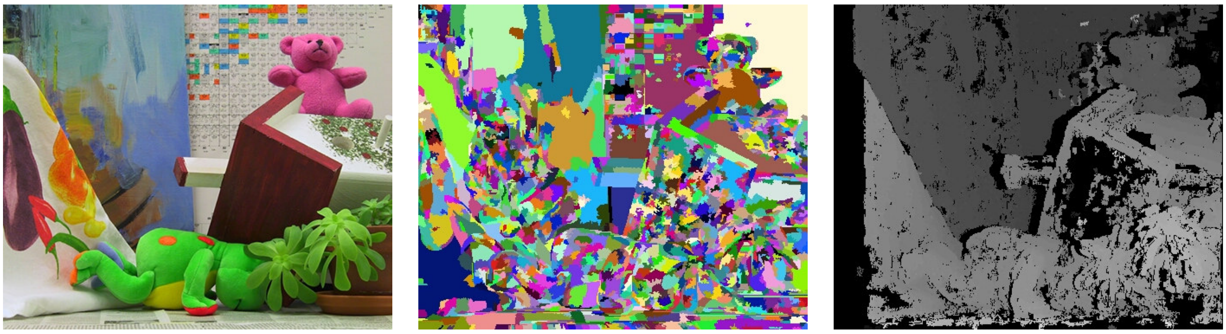
3.2. Initial disparity map

An initial disparity map is then created by computing the sum-of-absolute-differences (SAD) of colour values within a square window. Since this purely local procedure fails to find accurate disparity estimates in untextured and occluded regions, those regions are filtered out via cross validation. Optionally, the algorithm attempts to fill in unassigned regions by increasing the window size. This method is able to generate detailed disparity information for regions of rich texture by using small window sizes, while robust disparity estimates for poorly textured regions are produced by larger window sizes. The resulting disparity map using 3×3 and 7×7 windows is shown in figure 2c.

3.3. Plane fitting

The initial disparity map is used to initialize the planar equation of each segment. We represent a segment's disparity by a function

$$d(x, y) = a \cdot x + b \cdot y + c \tag{1}$$



(a) (b) (c)

Figure 2. Colour segmentation and initial disparity map. (a) Left image. (b) Computed colour segmentation. (c) Computed initial disparity map. Invalid points are coloured black.

with x and y being image coordinates and a, b and c being the plane parameters. For fitting the plane, we use a robust version of the method of least squared errors. The plane is thereby fitted to all valid disparity estimates inside the segment.

3.4. Layer Extraction

One single surface that contains texture is usually divided into several segments by applying colour segmentation. However, for segments of the same surface, their planar models should be very similar, as long as the surface can be well approximated by a plane. Following this idea, we project each segment into a five-dimensional feature space that consists of the two image coordinates of the segment's center of gravity and its three plane parameters. A modified mean-shift algorithm is then applied to find clusters of segments. Segments of the same cluster build a single disparity layer, whose plane model is derived by fitting a plane over all valid points inside those segments using the initial disparity map.

4. DISPARITY LAYER ASSIGNMENT

4.1. Problem definition

In the previous step, we estimated a set of planes that are likely to occur in the scene. In the following, each pixel of both views is assigned to at most one disparity layer. In figure 3, two disparity layers with layer 1 representing the foreground and layer 2 representing the background were extracted. The illustrated configuration assigns white pixels to the foreground and black pixels to the background. In our implementation, pixels assigned to the same surface are not restricted to have constant disparity, since their disparity is modelled by a planar equation. In future work, the planar representation could easily be replaced by a more elaborate surface model like a spline representation as used by Lin and Tomasi.[8]

By assigning a pixel to a layer, its disparity becomes defined through the use of the layer's plane model. Those pixels whose corresponding pixel is occluded in the other view pose special problems, since it is difficult to correctly assign them to a layer. Pixels can therefore be assigned to no plane, meaning that they are occluded. The occlusion problem provides a major motivation for our approach to use all pixels of *both* images in the layer assignment step. If one pixel that is visible in both views is assigned to a particular disparity layer, its matching point must also be assigned to the same layer. In this case, using the second image would not bring additional information. However, occlusions are different in each view, and using only one image would result in an unsymmetrical treatment of those. Furthermore, not all the information that is present in the image pair would be exploited, since information about occlusions in the second view is not included.

Unfortunately, the disparity layer assignment problem has a very large solution space. Given an image pair with $2 \cdot M$ pixels, where each of them is assigned to zero or one of N disparity layers, results in $(2 \cdot M)^{N+1}$

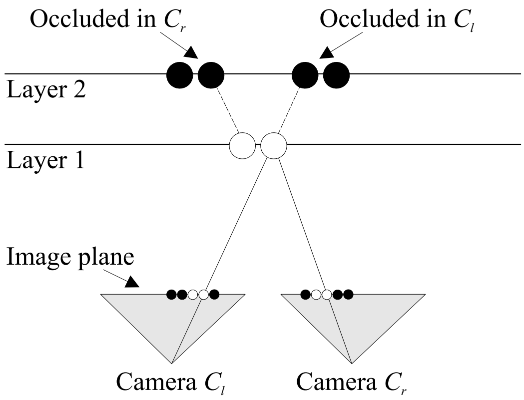


Figure 3. Assignment of pixels to disparity layers. Each pixel of both views gets assigned to at most one surface.

different possible solutions. A large solution space increases the chances for finding a suboptimal solution. We therefore apply a set of assumptions that put constraints on this solution space.

One major contribution of this work is the introduction of what we refer to as *segment consistency assump−tion*. As previously stated, we assume that disparities inside a segment can be modelled by the same planar model, i.e. all pixels of the same segment are assigned to the same disparity layer, if they are not occluded. For implementing this assumption, we do not only assign each pixel to at most one disparity layer, but also each segment. We therefore differentiate between a *pixel level* and a *segment level*. The segment level cannot completely replace the pixel level, since handling of occlusions is performed on the pixel level. The algorithm takes its strength from the collaboration of both levels. We only use the segmentation result of the left image, since segmenting both images would lead to inconsistent segmentation results across views due to different image formations. The segmentation information allows the algorithm to assign occluded pixels to meaningful disparity values using the segment's disparity layer assignment, which is of course only possible for pixels of the reference view.

4.2. Layer assignment as labelling problem

Assigning pixels and segments to disparity layers can be regarded as a labelling problem. A dedicated label 0 thereby denotes pixels and segments which are occluded and therefore not assigned to any layer. The labels $1, 2, \ldots, N$ correspond to the N disparity layers that were extracted. A labelling function $f(\cdot)$ is defined for both segments and pixels.

Let $p = (x, y, v)$ be a pixel defined by its image coordinates x and y and its view $v \in \{LEFT,\ RIGHT\}$. The set $I = I_{LEFT} \cup I_{RIGHT}$ denotes the union of all pixels p from both views, with I_{LEFT} being the left image and I_{RIGHT} being the right image. The labelling function $f(p)$ on the pixel level projects each pixel $p \in I$ to exactly one label k:

$$\forall p \in I : \qquad f(p) = f(x, y, v) = k, \qquad k \in \{0, 1, 2, \ldots, N\} \tag{2}$$

Let S be the set of segments extracted in the left view. The labelling function $f(s)$ on the segment level projects each segment $s \in S$ to exactly one label k:

$$\forall s \in S : \qquad f(s) = k, \qquad k \in \{0, 1, 2, \ldots, N\} \tag{3}$$

Labelling a pixel by a label $k \neq 0$ defines the corresponding point in the other view. The matching point $m[k](p)$ of pixel $p = (x, y, v)$ assigned to label k is obtained by computing the disparity according to equation (1) at the point (x, y, v) using the plane model of the kth depth layer and adding it to x. Formally expressed,

$$m[k](p) = m[k](x, y, v) = (x + d[k](x, y, v), y, \neg v) \tag{4}$$

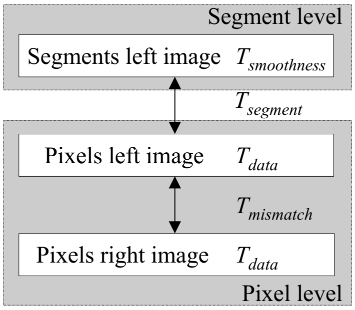


Figure 4. Basic components of the cost function and propagation of layer assignments.

with $d[k](x, y, v)$ being the disparity at point (x, y, v) according to the plane model of the kth layer and $\neg LEFT = RIGHT$ and vice versa. The plane parameters used for computation of $d[k](x, y, v)$ depend on the view v. A transformation from $LEFT$ to $RIGHT$ is done using the original plane parameters, which results in negative disparity values, whereas a transformation in the opposite direction is accomplished using the parameters of the "inverse" plane, which gives positive disparity values. The computed disparity is then rounded to the closest neighbour.

4.3. Cost function

In the following, we design a cost function $C(f)$, which measures the optimality of the label configuration f. We therefore define a set of terms that incorporate our basic assumptions. Some of these terms operate directly on the pixel level or on the segment level, while others propagate information between components. This is illustrated in figure 4. The overall cost function $C(f)$ is then built by the summation of these terms.

$$C(f) = T_{data} + T_{mismatch} + T_{segmentation} + T_{smoothness} \rightarrow \min \tag{5}$$

4.3.1. Colour consistency

The colour consistency assumption states that if both views receive contribution from the same scene point, the pixels to which the point projects should have similar colours in the left and in the right images. We incorporate this assumption on the pixel level by measuring the pixel dissimilarity at each visible point of both images. According to the literature, we refer to this term as data term T_{data}, which is defined by

$$T_{data} = \sum_{p \in I} \begin{cases} dissimilarity(p, m[f(p)](p)) & : \quad f(p) \neq 0 \\ -1 & : \quad \text{otherwise} \end{cases} \tag{6}$$

with $dissimilarity(p_1, p_2)$ being a function that computes the non-negative colour dissimilarity of two pixels p_1 and p_2. We use the pixel dissimilarity measurement of Birchfield and Tomasi[12] that is insensitive to image sampling. We extended this measurement to work on RGB values.

4.3.2. View consistency

The view consistency term propagates layer assignments from the reference image to the second view and vice versa. It motivates a consistent layer assignment across views, meaning that if a pixel in one image is assigned to a particular layer, also its matching point in the other image should be assigned to the same disparity plane. Assignments that violate this constraint are penalized by adding a non-negative constant mismatch penalty to the solution's costs. We define the view consistency term $T_{mismatch}$ by

$$T_{mismatch} = \sum_{p \in I} \begin{cases} \lambda_{mismatch} & : \quad f(p) = 0 \ \vee \ f(p) \neq f(m[f(p)](p)) \\ 0 & : \quad \text{otherwise} \end{cases} \tag{7}$$

with $\lambda_{mismatch}$ being a non-negative constant penalty.

The view consistency term incorporates the uniqueness assumption, which states that each pixel of one view matches at most one pixel in the other view. Let us assume that in one image there are two pixels assigned to different disparity layers that are matched to the same point in the other image. At most one of these pixels can be view consistent with its matching point, meaning that the points carry the same label. The inconsistent assignment will therefore be penalized by adding the mismatch penalty.

Furthermore, the term $T_{mismatch}$ accounts for penalizing pixels assigned to the occlusion label. Such pixels have to be penalized, since otherwise declaring all pixels as being occluded would form a trivial optimum of the cost function. In the definition of the data term T_{data} in equation (6), occluded pixels generate costs of -1. The costs produced by a view inconsistent pixel are therefore always larger than the costs for assigning this pixel to the occlusion label, even if the pixel dissimilarity of the view inconsistent point is 0. As a consequence, a view inconsistent pixel will be declared as being occluded. As a by-product, view consistent outliers having a pixel dissimilarity larger than $\lambda_{mismatch}$ will also be labelled occluded.

4.3.3. Segment consistency

The segment consistency term propagates layer assignments from the segment level to the reference image and vice versa. It enforces the assumption of smoothly varying disparity inside a segment. We embed this assumption by imposing a penalty set to infinity if a pixel is assigned to a different layer than the segment to which it belongs. However, the pixel is allowed to be assigned to no layer at all. In other words, if one pixel is assigned to a particular disparity layer, then any other pixel belonging to the same segment must either be assigned to this particular layer or be declared as occluded. Solutions that violate this constraint generate infinite costs. The segment consistency term is defined by

$$T_{segment} = \sum_{p \in I_{LEFT}} \begin{cases} \infty & : \; f(p) \neq 0 \wedge f(p) \neq f(segment(p)) \\ 0 & : \; \text{otherwise} \end{cases} \tag{8}$$

with $segment(p)$ being a function that returns the segment to which the pixel p belongs.

4.3.4. Smoothness

The smoothness assumption states that disparity varies smoothly almost everywhere, except at depth boundaries. We apply this assumption on the segment level by penalizing neighbouring segments that are assigned to different disparity layers. The smoothness term $T_{smoothness}$ is computed by

$$T_{smoothness} = \sum_{s_i, s_j \in S \wedge (s_i, s_j) \in NB} \begin{cases} \lambda_{disc} \cdot borderlength(s_i, s_j) \cdot coloursimilarity(s_i, s_j) & : \; f(s_i) \neq f(s_j) \\ 0 & : \; \text{otherwise} \end{cases} \tag{9}$$

with λ_{disc} being a non-negative constant penalty for discontinuity and NB being the set of all neighbouring segments. The function $borderlength(s_1, s_2)$ is defined by the number of neighbouring pixels (p_1, p_2) in 4-connectivity with p_1 belonging to segment s_1 and p_2 to segment s_2. Another function $coloursimilarity(s_1, s_2)$ measures the colour similarity of segments s_1 and s_2. It motivates segments of similar colour to be assigned to the same disparity layer. In our implementation, we define the function $coloursimilarity(s_1, s_2)$ by

$$coloursimilarity(s_1, s_2) = (1 - \frac{\min(|meancolour(s_1) - meancolour(s_2)|, 255)}{255}) \cdot 0.5 + 0.5 \tag{10}$$

with $meancolour(s)$ being the componentwise summed up RGB values of pixels inside segment s divided by their number. The absolute difference of the two RGB values is computed by summing up the absolute differences of each component, which gives a maximum value of $3 \cdot 255$ using an 8-bit coding for each colour channel. For identical mean colour values, the colour similarity function returns a value of 1, whereas for colour differences larger or equal to 255, it gives a value of 0.5. The costs of assigning two neighbouring segments of similar colour to different depth layers are therefore higher than separating two segments of low colour similarity.

5. OPTIMIZATION VIA GRAPH CUTS

Unfortunately, finding the labelling f of minimum cost $C(f)$ is shown to be np-complete. However, Boykov et al.[5] present an efficient optimization strategy based on graph cuts for labelling problems in computer vision. In this work, we adopt their α-expansion move to our problem formulation. An α-expansion move thereby changes the assignment of a subset of pixels and segments to the label α and leaves the other pixels and segments assigned to their old labels. Formally expressed, let f be the current label configuration of pixels and segments. The configuration f' is within one α-expansion move from f, if for each pixel p, $f'(p) = f(p)$ or $f'(p) = \alpha$ and for each segment s, $f'(s) = f(s)$ or $f'(s) = \alpha$. The problem of finding the optimal α-expansion move for our cost function, i.e. the move that gives the largest improvement of costs, among all possible α-expansion moves can be solved efficiently to optimality by computing the minimum cut in a special purpose graph.

Let $G = (V, E)$ be a weighted graph with two special vertices, called the source *src* and the sink *snk*. A cut is a partition of the vertices in V into two disjoint sets SRC and SNK with $src \in SRC$ and $snk \in SNK$. The costs of the cut are defined by the sum of all edges that go from SRC to SNK, and the minimum cut is the cut of minimum costs. According to the theorem of Ford and Fulkerson, solving the minimum cut problem is equivalent to computing the maximum flow. In our implementation, we apply the maximum flow algorithm of Boykov and Kolmogorov[13] that is specifically optimized for graphs arising in computer vision.

We construct a graph so that each segment as well as each pixel is represented by exactly one vertex v_i. Each vertex is connected to the source vertex *src* and the sink vertex *snk* by an edge. If after the computation of the cut $v_i \in SRC$, the old label is kept. Otherwise, if $v_i \in SNK$, the label α is taken. We insert edges into the graph in a way that the costs of each cut in the graph are equal to the costs of the resulting label configuration. Since the computed cut on the graph is the one of minimum costs, also the resulting configuration has minimum costs within one α-expansion move. For building the desired graph, we use the construction rules given by Kolmogorov and Zabih.[14] We illustrate the graph in figure 5.

The α-expansion move is embedded into a greedy algorithm. In the initial label configuration, all pixels and segments are assigned to the occlusion label. The algorithm then computes the cheapest α-expansion move for each disparity layer in fixed or random order. Additionally to the extracted layers, we also test a special layer that carries the occlusion label. If a move decreases the costs, then this is the new label configuration. This procedure is iterated until there is no disparity plane that further decreases the costs, which is usually the case after very few iterations. Since the α-expansion move changes a large number of assignments simultaneously, its application allows us to compute a strong local minimum.

6. EXPERIMENTAL RESULTS

To evaluate the proposed algorithm, we used the test bed provided by Scharstein and Szeliski.[1] They provide a set of four image pairs with corresponding ground truth. Authors that want to participate in the evaluation are asked to run their stereo algorithms on these stereo pairs using constant parameter settings. The computed disparity maps are then compared against the ground truth by computing the percentage of wrong pixels in *unoccluded* regions. A pixel is judged to be erroneous, if its absolute deviation from the ground truth is larger than one. The algorithms are then ranked according to their overall performance. In the online version of their paper, Scharstein and Szeliski currently tabulate 30 different stereo algorithms. We applied our algorithm to the proposed stereo pairs and were ranked on second place. In this section, we show results for the Tsukuba and Venus test sets that are used in the Middlebury benchmark. Furthermore, we present results for the more complex Teddy and Cones stereo images that were taken from Scharstein and Szeliski.[15] Finally, we show results for a self-recorded test set.

As a first image pair we present the Teddy test set shown in figure 6a and figure 6b. The corresponding ground truth is shown in figure 6c. The Teddy image pair is challenging for stereo algorithms, since it has a complex scene structure, a large disparity range ($0 \cdots 64$ pixels) and untextured, as well as large occluded regions. We show the disparity estimates on the pixel level for the left and right images in figures 6d and 6e. It can be seen that most occluded pixels (coloured blue) are correctly identified in both images, although some visible pixels erroneously carry the occlusion label, which is due to the outlier removal property of the view consistency term. Figures 6g and 6h show the corresponding layer assignments on the pixel level. As a consequence of the

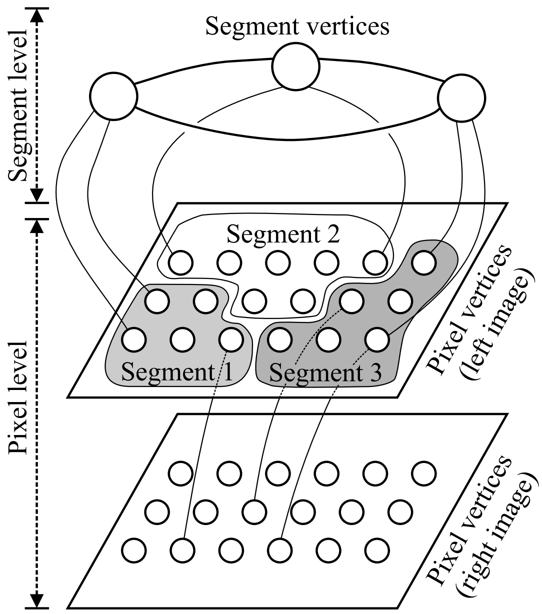


Figure 5. Layout of the graph. Not all edges are shown for legibility. Each of the illustrated vertices is connected to the source and sink vertices.

view consistency term, the assignments are consistent across views. The disparity map on the segment level, which also represents the final output of our algorithm, is presented in figure 6f. On the segment level, surfaces are represented by their planar model and therefore by a continuous-valued function, yielding subpixel-precision. Furthermore, occluded regions are filled in by meaningful disparity values as a consequence of the segmentation information and the smoothness term of the cost function. The disparity layer assignments of segments are then presented in figure 6i. These assignments are consistent with the assignments on the pixel level as a consequence of the segment consistency term. We compare the computed disparity map against the ground truth in figure 6j. We therefore plot pixels that have a disparity error larger than one pixel. Erroneous pixels in visible regions are coloured black and wrong pixels in occluded regions are assigned to grey. The percentage of pixels exceeding an error threshold of one is 4.77%, if only *unoccluded* pixels are considered. The error percentage for *all* pixels including occluded ones is 6.77%. To give a further impression of the accuracy and detail of the computed disparity information, we show a reconstruction of the scene in figure 6k.

In figure 7 we present results for standard test sets, as well as for a self-recorded one. The corresponding right images and ground truth for the standard images can be found on the Middlebury Stereo Vision website. The well-known Tsukuba test set and computed results are shown in figures 7(a1-a3). Wrong disparity assignments for this image pair are mainly caused by segments that overlap a depth discontinuity (e.g. tripod). Moreover, representing the head by only two layers oversimplifies the real surface. The Venus test set along with computed results is presented in figures 7(b1-b3). The algorithm correctly finds all five planes of which the scene consists. We point out that the newspaper at the right of figure 7(b1) consists of two planes that are joined by a crease edge, which is also accurately reconstructed by the algorithm. The more complex Cones image pair and corresponding results are then shown in figures 7(c1-c3). Wrong disparity values are mostly obtained in occluded regions. However, the scene is reconstructed quite accurately by a large number of disparity layers. Finally, we show a self-recorded stereo pair and the computed disparity map in figures 7(d1-d3). The background of the scene is represented to a large extent by a single layer, whereas the disparity of the teddy, that has a more complex structure, is accurately reconstructed using more disparity layers. Furthermore, the algorithm was able to capture the thin structures represented by the legs of the table.

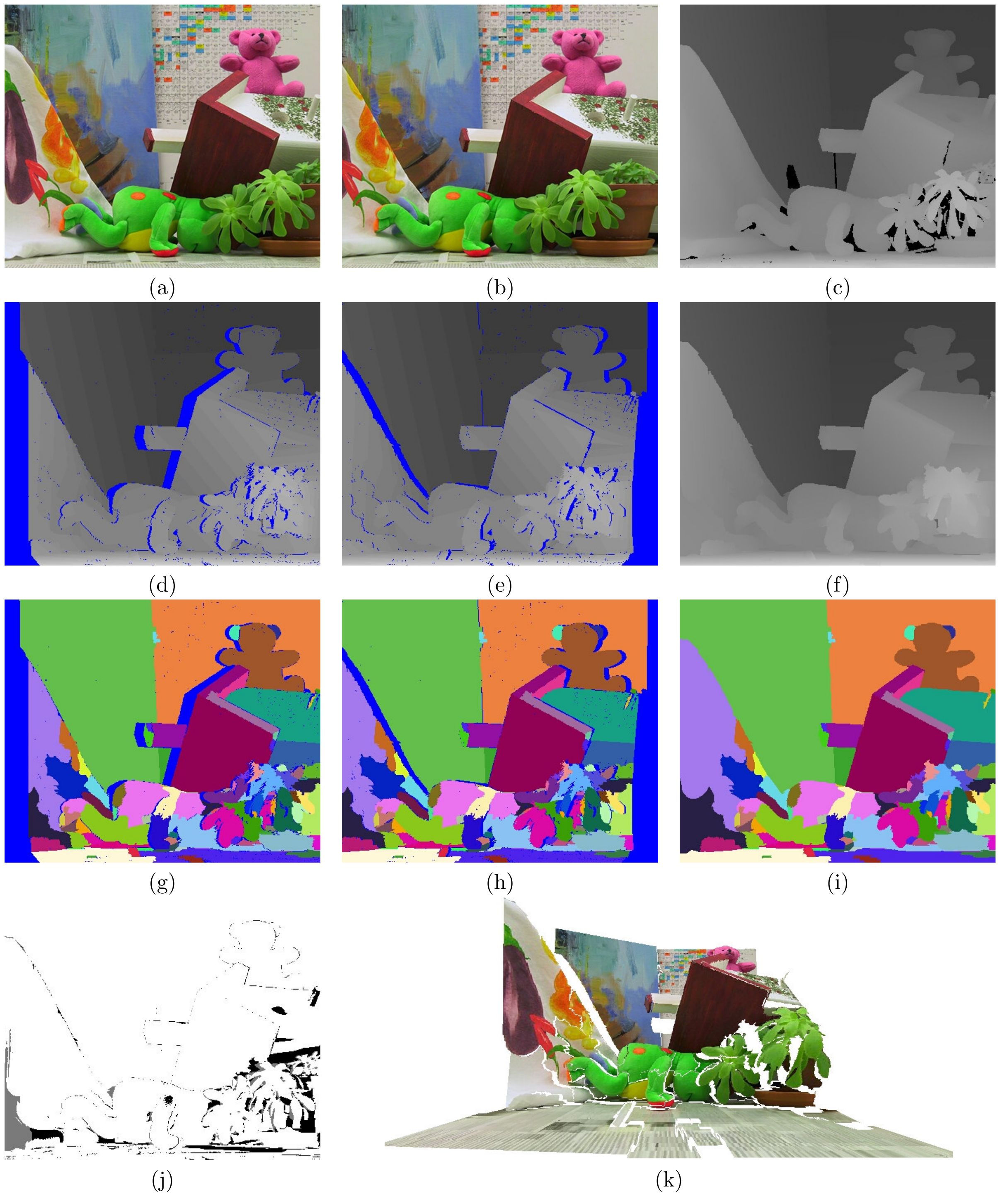


Figure 6. Results for the Teddy test set. (a) Left image. (b) Right image. (c) Ground truth provided with image pair. (d) Disparity assignments for pixels of the left view. Pixels assigned to the occlusion label are coloured blue in the colour version and black in the grey-level version of this paper. (e) Disparity assignments for pixels of the right view. (f) Disparity assignments for segments of the left view. (g) Layer assignments for pixels of the left view. (h) Layer assignments for pixels of the right view. (i) Layer assignments for segments of the left view. (j) Comparison of the disparity map (f) against the ground truth (c). (k) Reconstructed view.

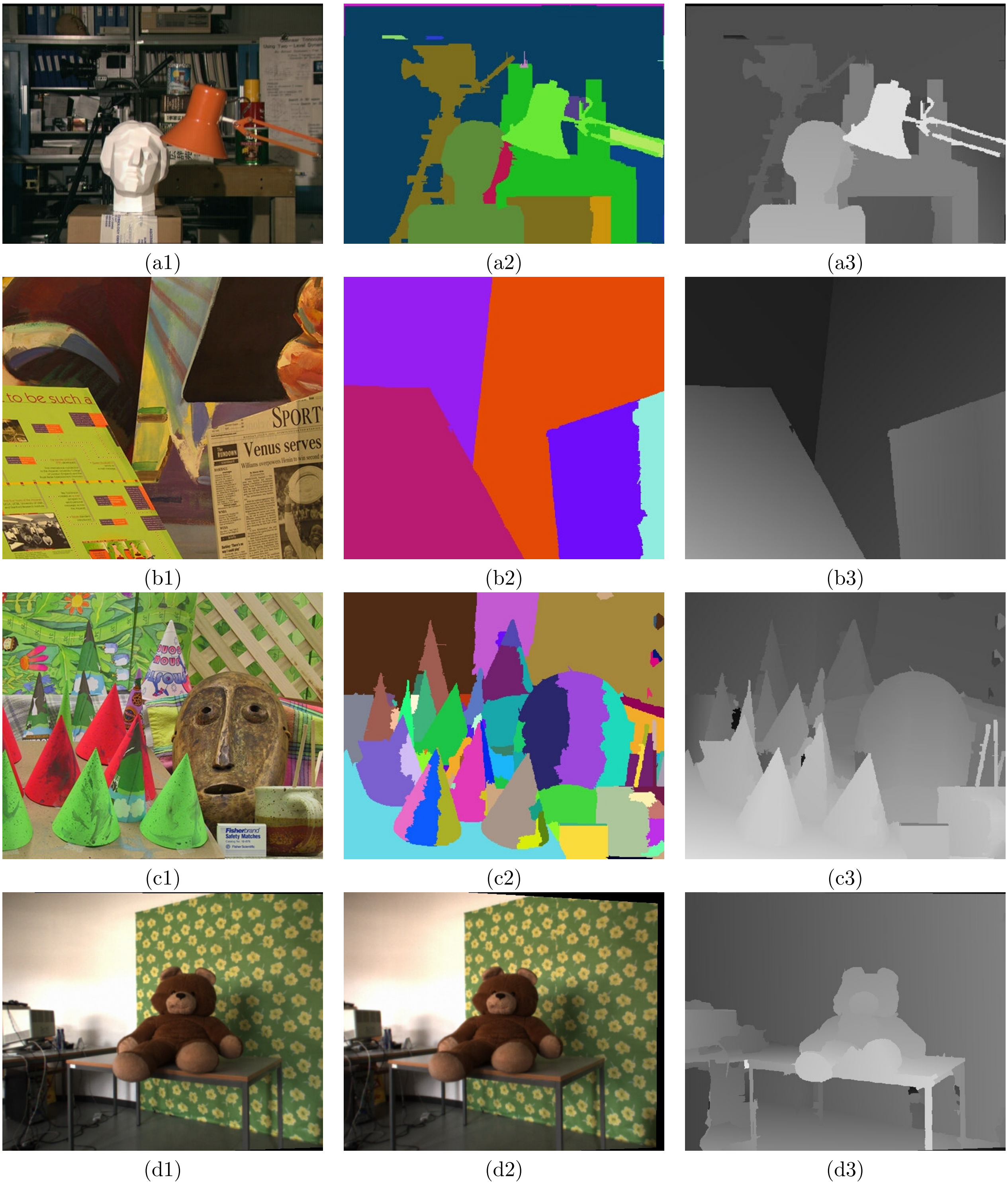


Figure 7. Results on standard and self-recorded image pairs. (a1) Left image of the Tsukuba test set. (a2) Disparity layers. (a3) Disparity map. (b1) Left image of the Venus test set. (b2) Disparity layers. (b3) Disparity map. (c1) Left image of the Cones test set. (c2) Disparity layers. (c3) Disparity map. (d1) Left image of a self-recorded image set. (d2) Right image. (d3) Disparity map.

7. CONCLUSION

We have described a new graph-based stereo algorithm for epipolar rectified image pairs. The proposed method applies colour segmentation on the reference image. Disparity inside each segment is assumed to vary smoothly, which is incorporated by modelling the disparity by a planar equation. Furthermore, we assume that depth boundaries coincide with segment borders. A set of disparity layers is extracted from initial disparity segments in a clustering process. Each pixel and segment are then assigned to zero or one of those disparity layers. A global cost function measures the quality of assignments on the pixel and segment levels. The algorithm takes advantage of the collaboration of both levels. Occlusions are handled symmetrically on the pixel level. The segmentation information is enforced by the segment consistency term of the cost function. Furthermore, a smoothness term on the segment level aims at generating smooth disparity solutions and propagates meaningful disparities to occluded regions. Robust minimization of the cost function is achieved by graph-based optimization. Results obtained for the Middlebury test set and a self-recorded image pair show the high performance of the proposed method. Very good reconstruction results are also achieved in untextured and occluded regions, which are challenging for stereo algorithms. In future work, we plan to extend the proposed method to tracking motion.

ACKNOWLEDGMENTS

Financial support for this work was obtained from the Austrian Science Fund (FWF) under project P15663.

REFERENCES

1. D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision* **47**(1/2/3), pp. 7–42, 2002. http://www.middlebury.edu/stereo/ accessed October 21, 2004.
2. H. Tao and H. Sawhney, "Global matching criterion and color segmentation based stereo," in *Workshop on Applications of Computer Vision*, pp. 246–253, 2000.
3. M. Bleyer and M. Gelautz, "A layered stereo algorithm using image segmentation and global visibility constraints," *To appear in International Conference on Image Processing* , 2004.
4. Y. Wei and L. Quan, "Region-based progressive stereo matching," in *Conference on Computer Vision and Pattern Recognition*, **1**, pp. 106–113, 2004.
5. Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *Transactions on Pattern Analysis and Machine Intelligence* **23**(11), pp. 1222–1239, 2001.
6. V. Kolmogorov and R. Zabih, "Computing visual correspondence with occlusions using graph cuts," in *International Conference on Computer Vision*, **2**, pp. 508–515, 2002.
7. S. Birchfield and C. Tomasi, "Multiway cut for stereo and motion with slanted surfaces," in *International Conference on Computer Vision*, pp. 489–495, 1999.
8. M. Lin and C. Tomasi, "Surfaces with occlusions from layered stereo," *Transactions on Pattern Analysis and Machine Intelligence* **26**(8), pp. 710–717, 2004.
9. L. Hong and G. Chen, "Segment-based stereo matching using graph cuts," in *Conference on Computer Vision and Pattern Recognition*, **1**, pp. 74–81, 2004.
10. M. Bleyer and M. Gelautz, "A layered stereo matching algorithm using image segmentation and global visibility constraints," *Submitted to ISPRS Journal of Photogrammetry and Remote Sensing* , 2004.
11. C. Christoudias, B. Georgescu, and P. Meer, "Synergism in low-level vision," in *International Conference on Pattern Recognition*, **4**, pp. 150–155, 2002.
12. S. Birchfield and C. Tomasi, "A pixel dissimilarity measure that is insensitive to image sampling," *Transactions on Pattern Analysis and Machine Intelligence* **20**(4), pp. 401–406, 1998.
13. Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *International Workshop on Energy Minimization Methods in Computer Vision and Pattern Recognition* , pp. 359–374, 2001.
14. V. Kolmogorov and R. Zabih, "What energy functions can be minimized via graph cuts?," *Transactions on Pattern Analysis and Machine Intelligence* **26**(2), pp. 147–159, 2004.
15. D. Scharstein and R. Szeliski, "High-accuracy stereo depth maps using structured light," in *Conference on Computer Vision and Pattern Recognition*, pp. 195–202, 2003.