

# Graph-cut-based stereo matching using image segmentation with symmetrical treatment of occlusions

Michael Bleyer\*, Margrit Gelautz

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

Received 27 November 2006; accepted 29 November 2006

---

## Abstract

This paper describes a dense stereo matching algorithm for epipolar rectified images. The method applies colour segmentation on the reference image. Our basic assumptions are that disparity varies smoothly inside a segment, while disparity boundaries coincide with the segment borders. The use of these assumptions 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. This cost function is based on the observation that occlusions cannot be dealt with in the domain of segments. Therefore, we propose a novel cost function that is defined on two levels, one representing the segments and the other corresponding to pixels. The basic idea is that a pixel has to be assigned to the same disparity layer as its segment, but can as well be occluded. The cost function is then effectively minimized via graph-cuts. In the experimental results, we show that our method produces good-quality results, especially in regions of low texture and close to disparity boundaries. Results obtained for the Middlebury test set indicate that the proposed method is able to compete with the best-performing state-of-the-art algorithms.

© 2006 Elsevier B.V. All rights reserved.

*Keywords:* Stereo matching; Segmentation-based matching; Graph-cuts; Occlusion problem

---

## 1. Introduction

Given two images that are recorded from slightly different perspectives, 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*.

The computation of a dense disparity map is known to be difficult in practice. A good indicator for the problem complexity is the huge number of

---

\*Corresponding author. Tel.: +43 1 58801 18865; fax: +43 1 58801 18898.

E-mail address: [bleyer@ims.tuwien.ac.at](mailto:bleyer@ims.tuwien.ac.at) (M. Bleyer).

papers that have been published on this correspondence problem ever since the early days of computer vision. Major challenges in stereo computation are twofold. Firstly, matching often fails in the absence of discriminative image features that can be uniquely matched in the other view. This is the case in *untextured regions* as well as in the presence of texture with only horizontal orientation (*aperture problem*). Secondly, a pixel's matching point can be *occluded* in the other view. Those occlusions often occur at disparity discontinuities, which makes it specifically challenging to precisely outline object boundaries. Nevertheless, accurate identification of disparity discontinuities is often required for modern applications such as novel view generation [19,29]. A large number of stereo algorithms fail in this respect, since the fact that there are occlusions is simply ignored.

In this work, we propose a layered approach for the stereo correspondence problem that combines the merits of *segmentation-based matching* and robust global optimization via *graph-cuts*. Our method explicitly addresses the problems outlined above. The major motivation for using the segmentation information is that graph-cut-based approaches often optimize cost functions whose smoothness terms bias towards the reconstruction of simple object shapes, i.e. they aim at minimizing border lengths. We believe that complex disparity boundaries can be identified more accurately by the use of monocular cues, such as the partition of the reference image into regions of homogeneous colour. Special care is taken on the accurate treatment of occlusions. This, together with the segmentation information, leads to an improved performance in regions close to disparity boundaries. Using a region-based approach, we take benefit of increased robustness in regions of poor texture as well as of the capability to hypothesize disparities for occluded areas, which in our problem formulation is even possible if the whole segment is occluded. In the following, we review graph-cut and segmentation-based approaches to the stereo correspondence problem. For a more elaborate summary and evaluation of recent stereo algorithms, the reader is referred to the work of Scharstein and Szeliski [20].

From a technical perspective, the stereo correspondence problem is known to be *ill-posed*. Commonly, a smoothness assumption is employed in order to make the problem tractable. This assumption states that the disparity field is expected

to be *piecewise smooth*. Recent advances in the computation of dense correspondences can to a large extent be attributed to modern optimization schemes, i.e. graph-cuts [2,7,11,12,14–16,18] and belief propagation [22]. Those optimization techniques are capable of effectively minimizing global energy functions that employ a two-dimensional smoothness term. They have been extensively used in stereo vision and have shown to give the best results at the current state-of-the-art in this field [20].

Among the first works employing graph-cuts, Roy and Cox [18] compute a global optimal solution with a single minimum cut in a special purpose graph in order to solve the stereo correspondence problem. Ishikawa [12] proves that a global minimum is indeed reachable via graph-cuts in polynomial time if the smoothness term is convex. Although this is of theoretical interest, convex smoothness terms overpenalize large jumps in disparity and therefore those approaches tend to blur depth boundaries. However, even the simplest discontinuity preserving smoothness function, i.e. the Potts model, results in a problem formulation whose optimization is proven to be np-complete [15]. Nevertheless, Boykov et al. [7] show that a strong local optimum, which is guaranteed to lie within a known factor of the real optimum, can be calculated for non-convex smoothness terms by iterative application of their  $\alpha$ -expansion move. The optimal  $\alpha$ -expansion move is derived by computing the minimum cut on a weighted graph, which can be accomplished in almost linear time when using specialized minimum cut/maximum flow algorithms [6]. The  $\alpha$ -expansion algorithm is successfully applied on different problem formulations that account for handling of occlusions by enforcing the uniqueness constraint [14], slanted surfaces [2] and strictly symmetrical treatment of occlusions [16].

Segmentation-based techniques for the stereo correspondence problem [5,11,23,25,27,29] have recently gained attention. Although quite different from each other, all of those approaches exploit monocular cues by using the observation that discontinuities in disparity are usually reflected by discontinuities in the intensity image. These methods assume that inside a segment of homogeneous colour the disparity values follow some particular smooth disparity model, while depth discontinuities coincide with the boundaries of those regions. These assumptions are quite reasonable for images of

natural scenes. The segmentation constraint is used in conjunction with different optimization schemes, including progressive approaches [25], cooperative stereo [27], special purpose optimization [5,23,29] and graph-cuts [11].

In comparison to the algorithm proposed in this paper, we consider the methods of Birchfield and Tomasi [2] and Lin and Tomasi [16] as being similar in the sense that they formulate the correspondence problem in two steps. First, they estimate a set of layer models that correspond to different depth surfaces occurring in the scene. In the second step, they then assign each pixel to exactly one of those layers. Both algorithms are often also categorized as being segmentation-based, since the layer assignment step divides the image into regions of homogeneous disparity. However, they do not benefit from colour segmentation.

Among prior work, the closest related to our approach is the stereo method presented by Hong and Chen [11]. Similar to our technique, they combine region-based matching with graph-based optimization. They heuristically identify occlusions in a preprocessing step, which then allows them to optimize a very simple energy function. However, the results of their algorithm obviously depend on the success of this preprocessing step and it is not clear how well an a-priori identification of occlusions can work, especially in the presence of large pixel displacements. In contrast to this, our cost function “knows” about the existence of occlusions. Disparities and occlusions are computed simultaneously, which we believe results in a more accurate reconstruction of both.

The remainder of this paper is organized as follows. Section 2 describes the basic idea behind our algorithm. We then explain the proposed technique, which we divide into two steps. In the first step (Section 3), we segment the reference image, estimate an initial disparity map and fit a disparity model to each segment. We then attempt to identify those disparity models that represent the dominant disparity surfaces of the scene. (These dominant disparity models are referred to as *layers*.) In the layer assignment step of the algorithm (Section 4), each part of the image is assigned to one of the extracted disparity layers and occlusions are identified. We therefore design a global cost function that models region-based matching with treatment of occlusions. Moreover, we show how our approach enforces the uniqueness constraint in order to detect occluded pixels. The resulting cost

function is then minimized using the  $\alpha$ -expansion framework. In the experimental results (Section 5), we apply our algorithm on standard image pairs and on a self-recorded test set. Finally, we give our conclusions in Section 6.

## 2. Basic idea

The major contribution of the proposed technique to the field of stereo computation lies in that we show how region-based matching can be modelled in a graph-cut approach with explicit treatment of occlusions. Our idea is motivated by the following observation. Let us consider the two views of a stereo pair illustrated in Fig. 1a. The images show two segments  $S_1$  and  $S_2$  in the left image as well as the corresponding segments  $S'_1$  and  $S'_2$  in the right view. The segment  $S_2$  is thereby slightly displaced in the right image as indicated by the arrows, while the disparity of segment  $S_1$  is zero.<sup>1</sup> As a consequence of the displaced foreground object, occlusions occur in both frames, which we colour red in Fig. 1b. Note that  $S_1$  and  $S'_1$  are partially affected by occlusions. If we now match the complete segment  $S_1$  of the left image in the right view using its correct disparity (zero in this case), this results in high matching costs in exactly these occluded regions. (We mark these areas by diagonal hatches in the left view of Fig. 1c.) As a consequence of the occlusion problem, it is quite likely that  $S_1$  gets erroneously assigned to a wrong disparity model that shows lower matching costs. Ignoring occlusions therefore does not only result in an incomplete problem formulation, but also has negative effects on the algorithm's results.

Unfortunately, occlusions cannot be dealt with in the domain of segments. For an explanation, let us again consider segment  $S_1$  from Fig. 1a. When modelling the problem on the segment level only, we can simply state that the disparity of segment  $S_1$  is zero. However, this is insufficient, since this would also mean that the occluded parts of  $S_1$  correctly match the second view at disparity zero. Therefore, the correct statement must be:  $S_1$  has disparity zero (*segment level*), but contains a set of occluded pixels  $O_1$  (*pixel level*). Consequently, occlusion detection requires the involvement of the pixel domain, which is as well the basic idea behind our algorithm.

<sup>1</sup>This example only serves to illustrate the basic idea behind the proposed approach. Segments cannot have zero disparity in a real scene.

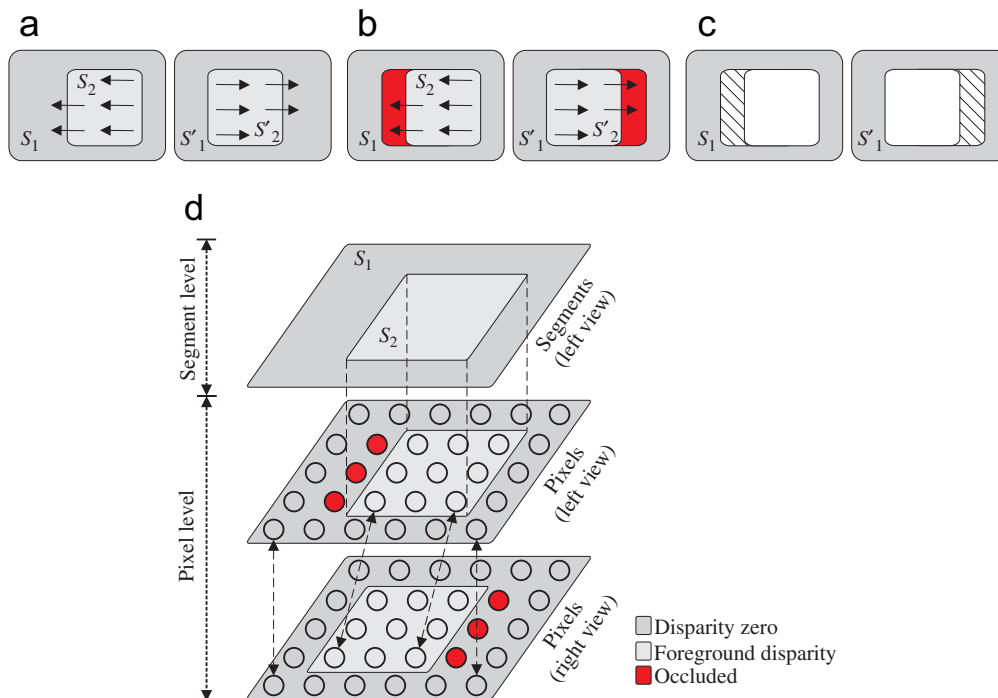


Fig. 1. The occlusion problem in segmentation-based matching and our proposed solution. An explanation is given in the text.

In the proposed approach, we overcome the problem of partially occluded segments by including the set of all pixels of the reference view into a global cost function. This is illustrated by the middle layer of Fig. 1d. Therefore, in addition to all segments, we also assign every pixel of the reference image to a disparity model. In our formulation, there is a dependency between the disparity model assignments of segments and pixels of the reference view. This link is built by the segmentation assumption. Recall that the segmentation assumption states that all pixels inside a segment follow the same disparity model. Our formulation implements this segmentation constraint by enforcing that every (visible) pixel is assigned to the same disparity model as the segment to which it belongs. However, and this is the important point, a pixel is also allowed to be occluded. Fig. 1d shows that by inclusion of the middle layer representing the pixels of the left image we are now able to correctly model the partial occlusion to the left of segment  $S_2$  in the reference view. The dashed lines between segment and pixel levels shall thereby indicate that the segmentation assumption is enforced on the pixel level.

Nevertheless, at this point, we still have not modelled the complete information that is present in the input image pair, since occlusions in the second view have not been considered so far. To account for those occluded regions, we also include every pixel of the right image into our problem formulation. This is represented by the bottom layer of Fig. 1d. The disparity model assignments of pixels in the left image thereby depend on the assignments of points in the right view and vice versa. Our basic consistency constraint is that a (visible) pixel and its matching point in the other image must both have identical disparity model assignments. As seen from Fig. 1d, we are now able to model the occlusions to the right of segment  $S_2$  in the second view. The arrows between middle and bottom layers illustrate the consistency constraint that operates between the left and right views. Modelling the pixels of both images in our cost function allows us to treat occlusions symmetrically. Moreover, it serves to model the uniqueness constraint, as we will describe later in this work. Details of our algorithm are given in the following.

### 3. Colour segmentation and layer extraction

#### 3.1. Colour segmentation

In the first step, we apply colour segmentation to the reference image. Since our basic assumption states that the disparity values inside a colour segment vary smoothly, it is important that a segment does not overlap a disparity discontinuity. It is therefore safer to use oversegmentation. To provide the reader with an idea of what such a segmentation looks like, we present our segmentation results on the left (reference) frame of the Teddy image pair (taken from [21]) in Fig. 2b, where pixels of the same colour belong to the same segment. We apply the mean-shift-based segmentation algorithm described by Christoudias et al. [8] in the current implementation.

#### 3.2. Layer extraction

When using a layered representation, the first questions one has to answer are: How many layers are present in the scene and what are their model parameters? There exist a variety of algorithms that solve this problem, for example, using  $k$ -means clustering [24], linear subspaces [13] or region growing [26]. Since the major focus of this work lies on the more complex layer assignment task, we do not aim at developing a new layer extraction procedure, but apply the algorithm that we have previously proposed in [5]. For completeness, this algorithm is briefly summarized in the following. A more detailed description is, however, found in the corresponding paper.

The algorithm starts by computing an initial disparity map using window-based correlation.

That is, for each pixel of the left image, we shift a square window on the corresponding scanline in the second view in order to find the point of maximum correspondence. The goodness of a potential match is thereby measured by computing the sum-of-absolute-differences (SAD) of RGB values within the support window. Note that the concept of window-based correlation is a purely local one. A pixel's disparity assignment is obtained by selecting the point of highest matching score (*winner-takes-all principle*) and independently of disparity assignments of neighbouring pixels. The method is therefore not able to generate correct disparity estimates in low-textured regions and areas affected by occlusions. To filter out these erroneous matches, we apply the left-right consistency check [10]. Pixels that have been invalidated by this check are then filled in by repeating this procedure using a larger window size, while pixels that have passed the test remain unchanged. In Fig. 3, we show the initial disparity map computed for the Teddy test set by applying  $3 \times 3$ ,  $5 \times 5$  and  $7 \times 7$  windows. Moreover, this figure serves to give an impression of the quality of results that one can expect from a local stereo algorithm. The advantage of using multiple windows in our approach is that we are able to generate detailed disparity information for regions of rich texture by using small window sizes. Robust disparity estimates for poorly textured regions are then produced by larger windows.

The initial disparity map serves to initialize the disparity model of each segment. In our current implementation, we represent a segment's disparity by a planar equation, which is

$$d(x, y) = a \cdot x + b \cdot y + c \quad (1)$$

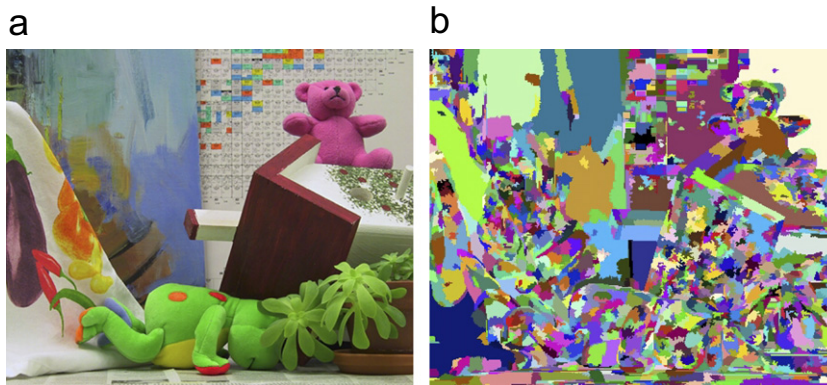


Fig. 2. Colour segmentation. (a) Left image. (b) Computed colour segmentation.

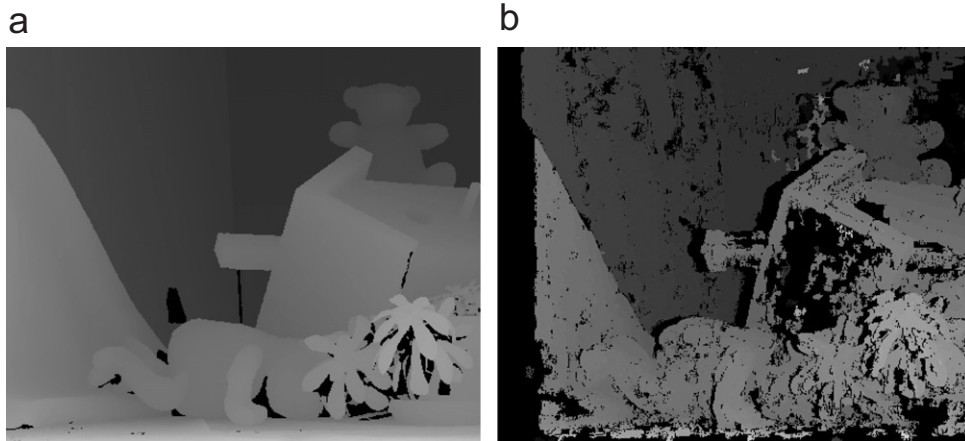


Fig. 3. Initial disparity map: (a) Ground truth data for the Teddy test set of Fig. 2a. High pixel values represent large disparities. (b) Computed initial disparity map. Pixels invalidated by the left-right consistency check are coloured black.

with  $x$  and  $y$  being image coordinates and  $a$ ,  $b$ ,  $c$  denoting the plane parameters. However, in principle, the layer assignment step of Section 4 works in combination with any smooth surface model. For fitting the plane, we use a robust version of the method of least squared errors. The plane is thereby fitted to all valid disparities inside the segment. Plane fitting might, however, not work for all segments. Since the computed initial disparity map is not dense, some segments will not capture enough disparity estimates in order to allow for the determination of a disparity plane. This is mostly the case for segments having very low texture or being occluded. Since it is relatively unlikely that correct disparity layer models can be derived from such segments, we simply exclude these segments from further processing in the layer extraction step. Nevertheless, a disparity model will be assigned to those segments in the layer assignment step of the algorithm as described in Section 4.

Once the initial disparity models are known, we attempt to identify those models that represent the dominant depth surfaces of the scene. These dominant surfaces are referred to as *layers* [24]. Note that since a surface of homogeneous disparity is, in general, not identified as a single colour segment, it is quite unrealistic to assume that layers coincide with segments. This is especially true in strongly textured regions and when applying over-segmentation (see Fig. 2b). However, for segments of the same surface, their planar models can be assumed to be very similar. This is why we apply a clustering procedure on the initial disparity segments.

For clustering disparity planes, we apply the mean-shift algorithm [9]. Its major advantage lies in that the number of clusters does not need to be known a-priori. The mean-shift algorithm requires some measurement to determine the similarity of two data points, which is usually the Euclidean distance. However, in our experiments, we found that comparing two disparity planes by computing the Euclidean distance between their parameters ( $a$ ,  $b$  and  $c$  from Eq. (1)) yields unsatisfactory results. We therefore modified the mean-shift algorithm in order to embed a more elaborate measurement taken from [23].

This measurement is illustrated in Fig. 4 and explained as follows. We first determine the normal vector on the first segment's plane, originating from that segment's center of gravity. This vector is intersected with the disparity plane of the second disparity segment. We then compute  $planedis_1$ , which is the length of the vector leading from the center of gravity to this intersection point. For symmetry, we also calculate  $planedis_2$  by taking the second segment as reference. The similarity of the two disparity planes is then determined by  $planedis_1 + planedis_2$ . We believe that this measurement is specifically well suited to the task of clustering disparity planes, since it does not only use the plane parameters, but also incorporates spatial information. Thus, spatially neighbouring segments are more likely to fall into the same cluster than two segments that are located on opposite sides of the reference image. This is a desirable property, which can be justified by the fact that neighbouring image segments are typically highly correlated,

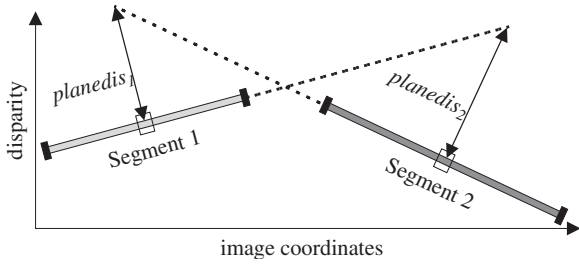


Fig. 4. Measuring the similarity of two disparity planes.

meaning that they are likely to be covered by the same disparity layer.

Once the clusters are computed, segments inside the same cluster are combined to form a single disparity layer. The plane model of a layer is then derived by fitting a plane over all those segments that build the layer by using the initial disparity map.

#### 4. Layer assignment

Knowing the set of disparity layers, the task of the assignment step is to estimate which parts of the images are covered by which layers as well as to identify occlusions. This step can thereby be regarded as more challenging than the layer extraction task, since it requires one to tackle all those points that make stereo matching difficult (textureless areas, occlusion problem, etc.). After introducing some basic notations, we set up the cost function that builds upon the observations of Section 2.

##### 4.1. Notations

We regard the task of assigning pixels and segments to disparity layers as a labelling problem. The labels  $1, 2, \dots, N$  correspond to the  $N$  disparity layers that have been computed in the layer extraction step. Moreover, a dedicated label 0 denotes pixels and segments that are occluded and therefore not assigned to any of those disparity layers. A labelling function  $f(\cdot)$  is then defined for both pixels and segments.

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

label  $k$ :

$$\forall p \in I : f(p) = f(x, y, v) = k, \quad k \in \{0, 1, 2, \dots, N\}. \quad (2)$$

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

$$\forall s \in S : f(s) = k, \quad k \in \{0, 1, 2, \dots, N\}. \quad (3)$$

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 Eq. (1) at the point  $(x, y, v)$  using the plane model of the  $k$ th disparity layer and adding it to  $x$ . Formally expressed,

$$m[k](p) = m[k](x, y, v) = (x + d[k](x, y, v), y, -v) \quad (4)$$

with  $d[k](x, y, v)$  being the disparity at point  $(x, y, v)$  according to the plane model of the  $k$ th disparity layer and  $\neg\text{LEFT} = \text{RIGHT}$  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. To derive whole-numbered image coordinates, we round the computed disparity to the closest neighbour.

##### 4.2. Cost function

Using the notation introduced above we design a cost function  $C(f)$ , which measures the optimality of a label configuration  $f$ . We therefore define a set of terms that incorporate our basic ideas. Some of these terms operate directly on the pixel level or on the segment level, while others propagate disparity layer assignments between the different layers of Fig. 1d. We give an overview of those terms and their scope in Fig. 5. The overall cost function  $C(f)$ , which is subject to minimization, is then built by summation of these terms:

$$C(f) = T_{\text{data}}(f) + T_{\text{occlusion}}(f) + T_{\text{segment}}(f) + T_{\text{mismatch}}(f) + T_{\text{smoothness}}(f). \quad (5)$$

The individual terms of  $C(f)$  are described one after the other in the following.

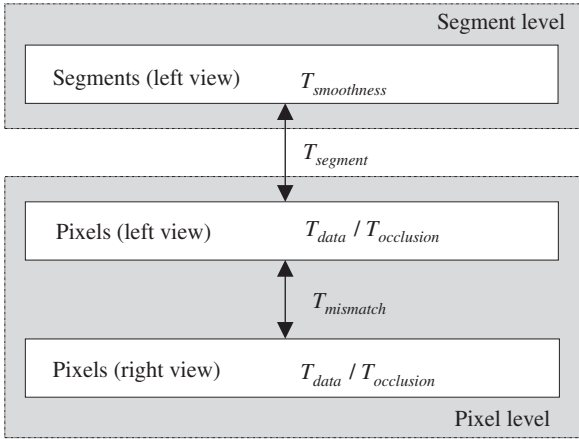


Fig. 5. Terms of the cost function  $C(f)$  and their scope.

#### 4.2.1. Data term

The first term  $T_{\text{data}}$  measures the agreement of  $f$  with the input data by exploiting the photo-consistency constraint. Thus, we assume that pixels of the left and right images that are the projections of the same scene point show similar colour values. We incorporate this assumption on the pixel level by measuring the pixel dissimilarity at each visible point of both images. More precisely, we compute the dissimilarity between a pixel  $p$  and its matching point  $m[f(p)](p)$  in the other view according to  $p$ 's current disparity layer assignment  $f(p)$ . Formally, the data term  $T_{\text{data}}$  is defined by

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

with  $\text{pixeldis}(p_i, p_j)$  being a function that computes the colour dissimilarity of two pixels  $p_i$  and  $p_j$ . In our current implementation, we use the pixel dissimilarity measurement of Birchfield and Tomasi [1]. This measurement has the advantage of being less sensitive to image sampling. Since it has been originally proposed to compute the dissimilarity of grey value pixels only, we have applied a simple modification in order to make the measurement work on RGB values as well.

#### 4.2.2. Occlusion term

The occlusion term of our cost function serves to penalize occluded pixels in both input views. This penalty is necessary, since otherwise declaring all pixels as occluded would result in a trivial minimum of  $C(f)$ . We therefore define the occlusion term

$T_{\text{occlusion}}$  by

$$T_{\text{occlusion}}(f) = \sum_{p \in I} \begin{cases} \lambda_{\text{occ}} & : f(p) = 0 \\ 0 & : \text{otherwise} \end{cases} \quad (7)$$

with  $\lambda_{\text{occ}}$  denoting a constant user-set parameter.

#### 4.2.3. Segmentation term

The segmentation term propagates disparity layer assignments between the segments and the pixels of the reference view. It enforces the assumption of smoothly varying disparity inside a segment on the pixel level. We embed this assumption by imposing a penalty set to infinity for every visible pixel of the left image that carries a different disparity layer assignment than its corresponding segment. Formally, we define the segmentation term  $T_{\text{segment}}$  by

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

with  $\text{seg}(p)$  being a function that returns the segment to which the pixel  $p$  belongs.

The consequence of the segmentation term is the following. Let us consider two pixels of the same segment. Both pixels are visible, i.e. they do not carry the occlusion label. If one pixel is now assigned to the same disparity layer as its corresponding segment, then the other pixel must also be assigned to exactly this particular disparity layer. Otherwise, the segmentation term generates infinite costs and such a configuration will therefore not be produced in the optimization part of the algorithm. Consequently, it is not possible that two pixels of the same segment are assigned to two different disparity layers. This is obviously equivalent to the statement that all non-occluded pixels inside a segment are modelled by the same disparity layer, which is exactly what our segmentation assumption requires. This is why the term introduced above enforces the segmentation constraint on the pixel level. However, note that occluded pixels are not affected by the segmentation term. Therefore, a pixel of the reference view can always carry the occlusion label independently of its segment's disparity layer assignment.

#### 4.2.4. View consistency term

The view consistency term propagates disparity layer assignments from the reference image to the second view and vice versa. It motivates consistent disparity layer assignments across views, meaning

that if a pixel in one image is assigned to a particular disparity layer, its matching point in the other image should also be assigned to exactly this disparity layer. We call assignments that violate this constraint view inconsistent. Such view inconsistent assignments are penalized by adding a constant value to the solution's costs. We define the view consistency term  $T_{\text{mismatch}}$  by

$$T_{\text{mismatch}}(f) = \sum_{p \in I} \begin{cases} \lambda_{\text{mismatch}} & : f(p) \neq 0 \wedge f(p) \neq f(m[f(p)](p)) \\ 0 & : \text{otherwise} \end{cases} \quad (9)$$

with  $\lambda_{\text{mismatch}}$  being a user-defined penalty. This term is also used in the work of Lin and Tomasi [16]. Ideally, view consistency should be enforced by penalizing inconsistent solutions with infinite costs. This is, however, not possible in our formulation for reasons related to the optimization part of the algorithm. More precisely, view inconsistent solutions are generated in intermediate steps of the optimization method.

#### 4.2.5. Smoothness term

The last term of our cost function is the smoothness term. Note that smoothness is, to some extent, already enforced due to the region-based nature of the proposed algorithm. However, we apply a strong oversegmentation and therefore image areas that can be well modelled by the same disparity layer will, in general, be represented by more than one segment. Consequently, it makes sense to incorporate an explicit smoothness term into our cost function in order to propagate disparity layer assignments across neighbouring segments. Our cost function implements the smoothness assumption on the segment level by penalizing neighbouring segments that are assigned to different disparity layers. Formally, the smoothness term  $T_{\text{smoothness}}$  is computed by

$$T_{\text{smoothness}}(f) = \sum_{(s_i, s_j) \in NB} \begin{cases} \lambda_{\text{disc}} \cdot bl(s_i, s_j) \cdot cs(s_i, s_j) & : f(s_i) \neq f(s_j) \\ 0 & : \text{otherwise} \end{cases} \quad (10)$$

with  $\lambda_{\text{disc}}$  being a user-set constant penalty for discontinuity and  $NB$  being the set of all neighbouring segments. The function  $bl(s_i, s_j)$  computes the border length by counting the number of neighbouring pixels  $(p_i, p_j)$  in 4-connectivity with  $p_i$

belonging to segment  $s_i$  and  $p_j$  to segment  $s_j$ . The second function  $cs(s_i, s_j)$  measures the colour similarity of segments  $s_i$  and  $s_j$ .

The basic idea behind weighting the smoothness penalty by the function  $cs(\cdot, \cdot)$  is that we consider two segments showing similar colour as more likely to originate from the same real-world surface than two segments of completely different colour. As an example, consider an image background of relatively homogeneous colour that is divided into several segments by colour segmentation. In our implementation, we define the function  $cs(s_i, s_j)$  by

$$cs(s_i, s_j) = \left( 1 - \frac{\min(|\text{meancolour}(s_i) - \text{meancolour}(s_j)|, 255)}{255} \right) \cdot 0.5 + 0.5 \quad (11)$$

with  $\text{meancolour}(s)$  being the componentwise summed up RGB values of pixels inside segment  $s$  divided by the segment's number of pixels. 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 disparity layers are therefore higher than separating two segments of low colour similarity.

The reasons why the smoothness term operates on the segment level instead of being defined on both views in the domain of pixels are twofold. Firstly, defining the term on the pixel level would mean that smoothness is also enforced for pixels which carry the occlusion label. However, while the smoothness assumption, in general, holds true for visible compact shapes, it is not valid for occluded areas that are usually long and thin. Secondly, defining the smoothness term on the segment level allows the propagation of “meaningful” disparity layers to segments that are completely occluded, i.e. segments that do not contain a single visible pixel on the pixel level.

#### 4.3. Modelling the uniqueness assumption

The uniqueness assumption states that a pixel of one view matches at most a single pixel in the other image. This constraint is often used to identify occlusions by enforcing one-to-one correspondences for visible pixels across images (e.g. [14,28]).

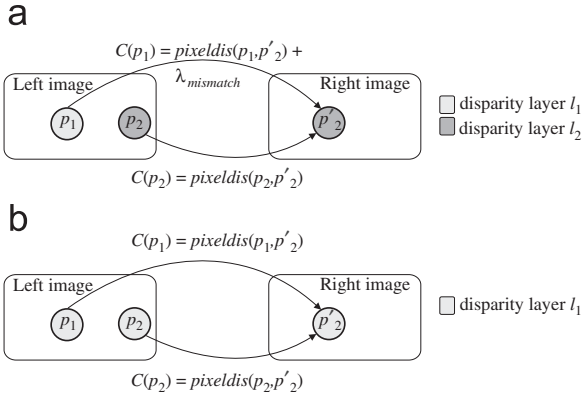


Fig. 6. Modelling the uniqueness assumption. (a) Two pixels assigned to different disparity layers match the same pixel in the other view. (b) Two pixels assigned to the same disparity layer match the same pixel in the other view. More details are found in the text.

Incorporation of the uniqueness assumption into our approach is relatively simple, since it just requires careful setting of two parameters of the cost function, as we describe in the following.

Let us first construct an example where the uniqueness constraint can help us to detect an occluded pixel. We illustrate such a case in Fig. 6a. The illustration shows two pixels  $p_1$  and  $p_2$  assigned to different disparity layers  $l_1$  and  $l_2$ , respectively. According to their current disparity layer assignments, both pixels project to the same matching point  $p'_2$ . Therefore, if we assume that the uniqueness constraint is valid,  $p_1$  and  $p_2$  cannot be visible at the same time. Obviously, our cost function penalizes configurations such as the one in the illustration. More precisely, since  $p_1$  and  $p_2$  originate from different surfaces (i.e. they are modelled by different disparity layers), only one of them can have a view consistent disparity layer assignment with its matching point  $p'_2$ . This is the pixel  $p_2$  in our example, and the costs for assigning  $p_2$  are solely those produced by the data term of Eq. (6) (see also Fig. 6a). In contrast to this, the costs for the assignment of  $p_1$  are not only those given from the data term, but also those of the view consistency term of Eq. (9), which adds  $\lambda_{\text{mismatch}}$  to the overall costs (see again Fig. 6a).

The basic idea now is to not only penalize configurations that violate the uniqueness constraint, but rather to avoid them completely. Our simple solution to this is to set the occlusion penalty  $\lambda_{\text{occ}}$  given by the occlusion term of Eq. (7) to a lower

value than that of the mismatch penalty  $\lambda_{\text{mismatch}}$ .<sup>2</sup> By doing so, we guarantee that the costs for declaring a pixel as occluded are always lower than the costs for assigning it to a view inconsistent disparity layer. Referring again to Fig. 6a, in the best case, the view inconsistent assignment of  $p_1$  generates costs of  $\lambda_{\text{mismatch}}$  (if there is a perfect agreement in colour values between  $p_1$  and  $p'_2$  so that the pixel dissimilarity is zero). However, the costs for assigning  $p_1$  to the occlusion label are  $\lambda_{\text{occ}}$  with  $\lambda_{\text{occ}} < \lambda_{\text{mismatch}}$  and therefore this is the configuration that will be produced by the optimization algorithm.

An aspect that has been overlooked until recently is that for horizontally slanted surfaces the uniqueness assumption is violated [17]. Let us consider a slanted plane recorded with two cameras. The projections of such a plane will, due to the slant, show different widths in the two images. As a consequence of this different sampling, there are points in one view that correspond to more than actually one pixel of the other view. Application of the uniqueness constraint for the reconstruction of slanted surfaces can therefore lead to suboptimal results.

In Fig. 6b, we illustrate two pixels  $p_1$  and  $p_2$  that are both assigned to disparity layer  $l_1$ , which means that they lie on the same surface. Due to some slant on their surface, both pixels have the same matching point  $p'_2$  in the other view. For this reason, this configuration clearly violates the uniqueness constraint. However, both disparity layer assignments (i.e. that of  $p_1$  as well as that of  $p_2$ ) are view consistent with the assignment of the matching point  $p'_2$ , since all of the three pixels are assigned to disparity layer  $l_1$ . Therefore, our cost function does not penalize this configuration by the view consistency term (see Fig. 6b). So what we actually implement is not strictly the uniqueness constraint, but rather an improved form of it that can also handle slanted surfaces.

#### 4.4. Optimization

To approximate the minimum of  $C(f)$ , we iteratively apply the  $\alpha$ -expansion move [7]. In our formulation, an  $\alpha$ -expansion move changes the assignment of a subset of pixels and segments to the layer  $\alpha$  and leaves the other pixels and segments assigned to their old layers. The optimal  $\alpha$ -expansion move is the one that results in the largest

<sup>2</sup>In our experiments, we use  $\lambda_{\text{occ}} := \lambda_{\text{mismatch}} - 1$ .

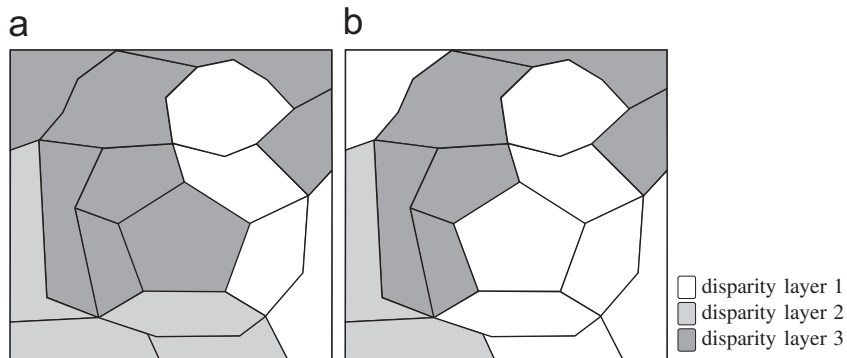


Fig. 7.  $\alpha$ -Expansion on the segment level. (a) The image is divided into a set of segments. Each segment is assigned to one of the three disparity layers. (b)  $\alpha$ -Expansion of disparity layer 1. Some segments change their assignments to disparity layer 1, while the others keep their original assignments.

improvement of costs. We give an example of an  $\alpha$ -expansion move on the segment level in Fig. 7.

Kolmogorov and Zabih [15] show that for a restricted class of cost functions the optimal  $\alpha$ -expansion move can be derived by computing the minimum-cut in a special-purpose graph. We can prove that our cost function  $C(f)$  belongs to this class by checking the condition given in their paper. Since this proof is fairly complex, it is omitted in this paper, but can be found in [3]. We build the desired graph by applying the construction rules given by Kolmogorov and Zabih [15]. Details of this graph construction are also given in [3]. However, to provide the reader with an approximate idea of what its structure looks like, we illustrate this graph in Fig. 8. The minimum-cut in this graph is then computed using the algorithm of Boykov and Kolmogorov [6].

The  $\alpha$ -expansion move is embedded into a greedy algorithm. All pixels and segments are set to the occlusion label in the initial configuration. Starting from this configuration, the algorithm computes the optimal  $\alpha$ -expansion move for each disparity layer. In addition 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 configuration. This procedure is iterated until there is no layer that can further decrease the costs by application of the  $\alpha$ -expansion move. This is usually the case after very few iterations.

## 5. Experimental results

To test our algorithm, we first run the layer extraction step of Section 3 on the input image pair. Knowing the disparity layers, we then invoke the layer

assignment procedure of Section 4. Upon convergence of the  $\alpha$ -expansion algorithm, we refit each disparity layer that is present in the generated solution over its new spatial extent. We again run the  $\alpha$ -expansion algorithm in order to check whether any of those new disparity layer models can produce a solution of lower cost. If this is the case, this procedure is iterated. Otherwise, the algorithm returns the current disparity layer assignment as final output.

To evaluate the proposed algorithm, we use the test bed provided by Scharstein and Szeliski [20]. They provide a set of four image pairs with corresponding ground truth. Authors who 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,<sup>3</sup> Scharstein and Szeliski tabulate approximately 40 different stereo algorithms. We applied our algorithm to the proposed stereo pairs and submitted the results to the Middlebury Stereo Vision Research Page. Currently, our algorithm (that is denoted by *Graph + segm.* in the table) is ranked on fourth position.<sup>4</sup> In the following, we show results for the

<sup>3</sup><http://www.middlebury.edu/stereo/>

<sup>4</sup>Note that we ranked our algorithm in 2005 at the time of publication of a preliminary conference paper [4] describing this algorithm. However, in the meantime, Scharstein and Szeliski

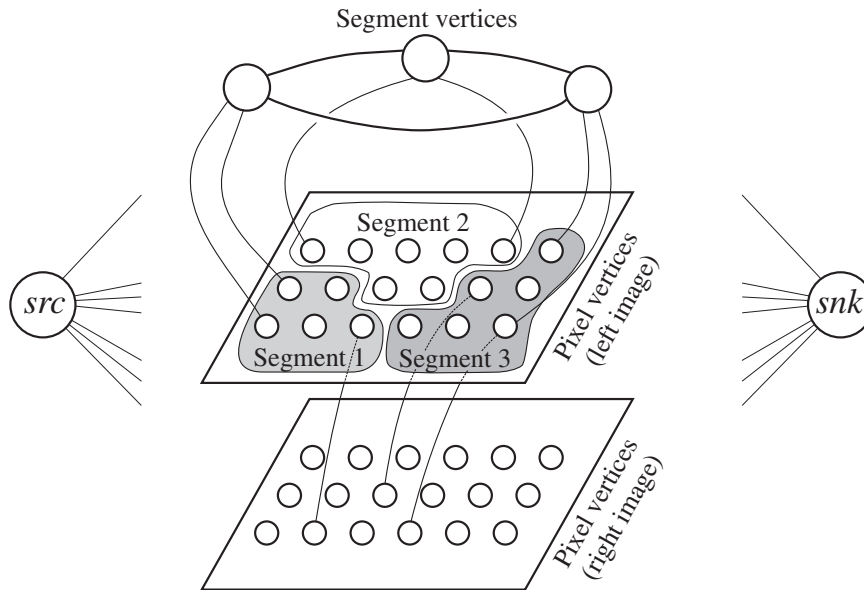


Fig. 8. Layout of the graph. (Not all edges are shown for legibility.) Vertices in the graph correspond to segments and pixels, while edges represent the terms of our cost function. Each of the illustrated vertices is connected to two special nodes: the source  $src$  and the sink  $snk$ . The minimum cut forms a partition of the vertices into two disjoint sets  $SRC$  and  $SNK$  with  $src \in SRC$  and  $snk \in SNK$ . The edges in the graph are adjusted in a way that the minimum cut yields the optimal configuration within one  $\alpha$ -expansion move from the current configuration.

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 [21]. Finally, we show results for a self-recorded test set.

As a first image pair we present the Teddy test set shown in Figs. 9a and b. The corresponding ground truth is presented in Fig. 9c. The Teddy image pair is challenging for stereo algorithms, since it has a complex scene structure, a large disparity range (0, ..., 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 Figs. 9d and e. It can be seen that most occluded pixels (coloured blue in the colour version and black in the grey-level version of this work) are correctly identified in both images, although some visible pixels erroneously carry the occlusion label. This happens for gross outliers whose pixel dissimilarity is large. Figs. 9g and h show the corresponding layer assignments on the pixel level. As a consequence of

the 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 Fig. 9f. 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 Fig. 9i. These assignments are consistent with the assignments on the pixel level as a consequence of the segmentation term. We compare the computed disparity map against the ground truth in Fig. 9j. 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. From this comparison against the ground truth it can be seen that our algorithm performs specifically well in the reconstruction of disparity discontinuities. This can be attributed to the incorporation of colour segmentation into our approach as well as to the accurate treatment of occlusions in both views. For quantitative evaluation, we compute two error percentages. First, we

(footnote continued)

changed the image pairs used for evaluation. More precisely, they included the Teddy and Cones test sets. Very recent methods are therefore listed on a new table.

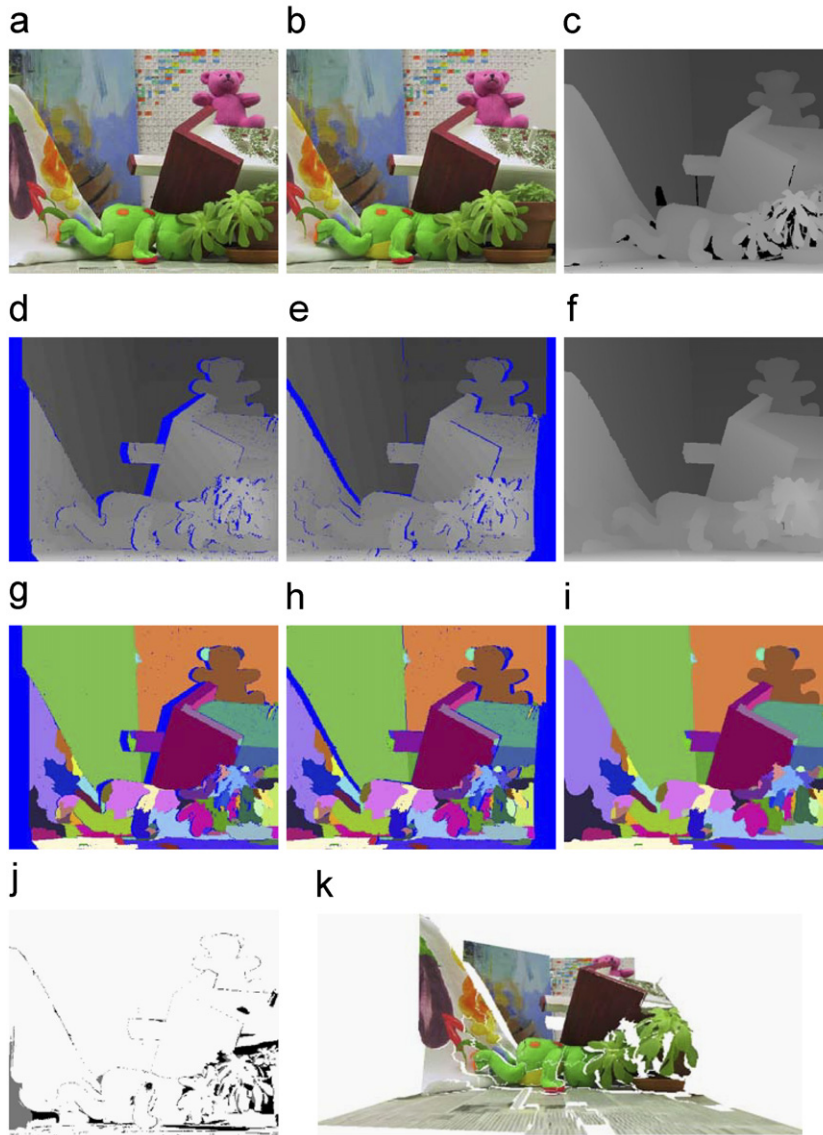


Fig. 9. Results for the Teddy test set. (a,b) Left and right images. (c) Ground truth provided with image pair. (d,e) Disparity assignments for pixels of the left and right views. Pixels assigned to the occlusion label are coloured blue. (f) Disparity assignments for segments of the left view. (g,h) Disparity layer assignments for pixels of the left and right views. (i) Disparity layer assignments for segments of the left view. (j) Comparison of the disparity map (f) against the ground truth (c). (k) Reconstructed view.

calculate the percentage of pixels exceeding an error threshold of one when considering *unoccluded* pixels only, which is also the error metric used in the Middlebury benchmark. Using this measurement, the error percentage is 4.77%. Second, we compute the error percentage for *all* pixels including occluded ones. The percentage of wrong pixels according to this metric is 6.77%. To give a further impression of the accuracy and detail of the computed disparities, we show a 3d-reconstruction in Fig. 9k.

As a second test image pair, we use the well-known Tsukuba set. The results for this image pair are shown in Fig. 10. Wrong disparity assignments for this image pair are mainly caused by segments that overlap a depth discontinuity (e.g. the tripod). Moreover, representing the head by two planar disparity layers oversimplifies the real surface. However, this could easily be improved by the use of a more sophisticated disparity model. The error percentage computed over all *unoccluded* pixels is

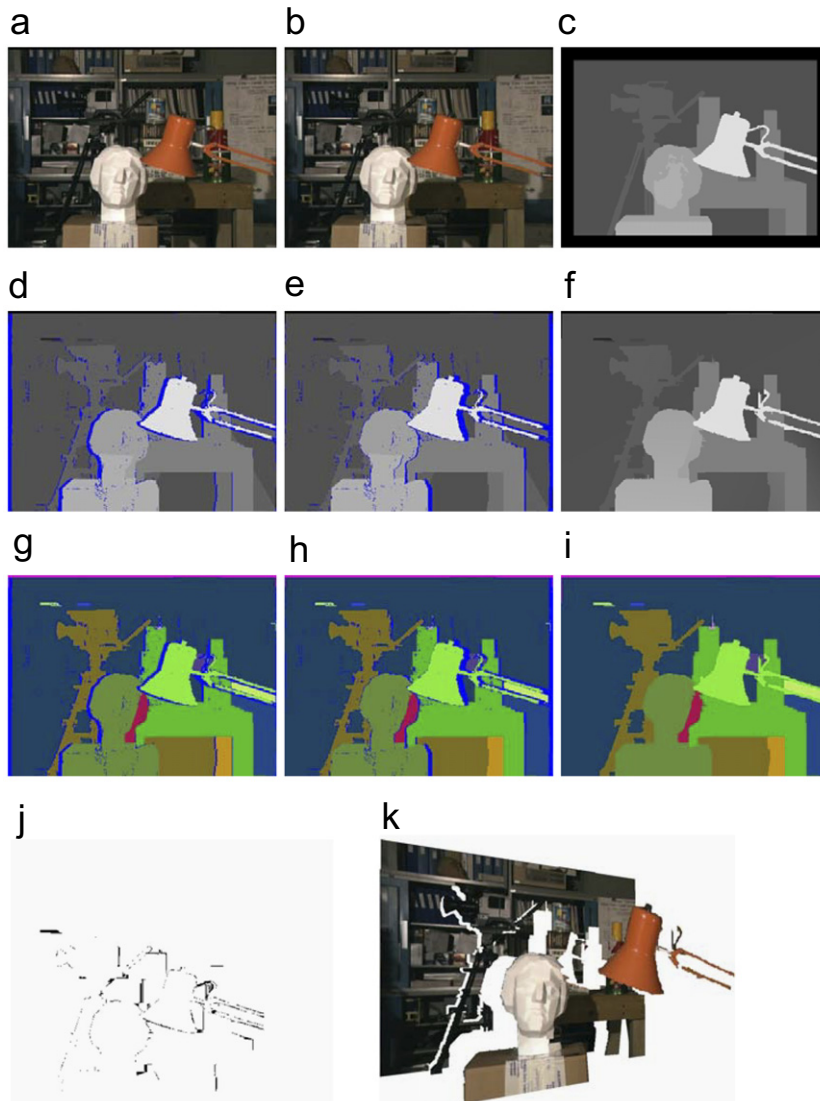


Fig. 10. Results for the Tsukuba test set. (a,b) Left and right images. (c) Ground truth provided with image pair. (d,e) Disparity assignments for pixels of the left and right views. Pixels assigned to the occlusion label are coloured blue. (f) Disparity assignments for segments of the left view. (g,h) Disparity layer assignments for pixels of the left and right views. (i) Disparity layer assignments for segments of the left view. (j) Comparison of the disparity map (f) against the ground truth (c). (k) Reconstructed view.

1.63%, while the percentage of *all* wrong pixels including occluded ones is 1.99%.

In Fig. 11 we present additional results for standard test sets as well as for a self-recorded one. The corresponding right images and ground truth data for the standard images can be found on the Middlebury Stereo Vision website. The Venus test set along with computed results is presented in Figs. 11(a1–a3). The algorithm correctly finds all five planes of which the scene consists. We point out that the newspaper at the right of Fig. 11(a1)

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 Figs. 11(b1–b3). 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 Figs. 11(c1–c3). The background of the scene is represented to a large extent by a single layer,

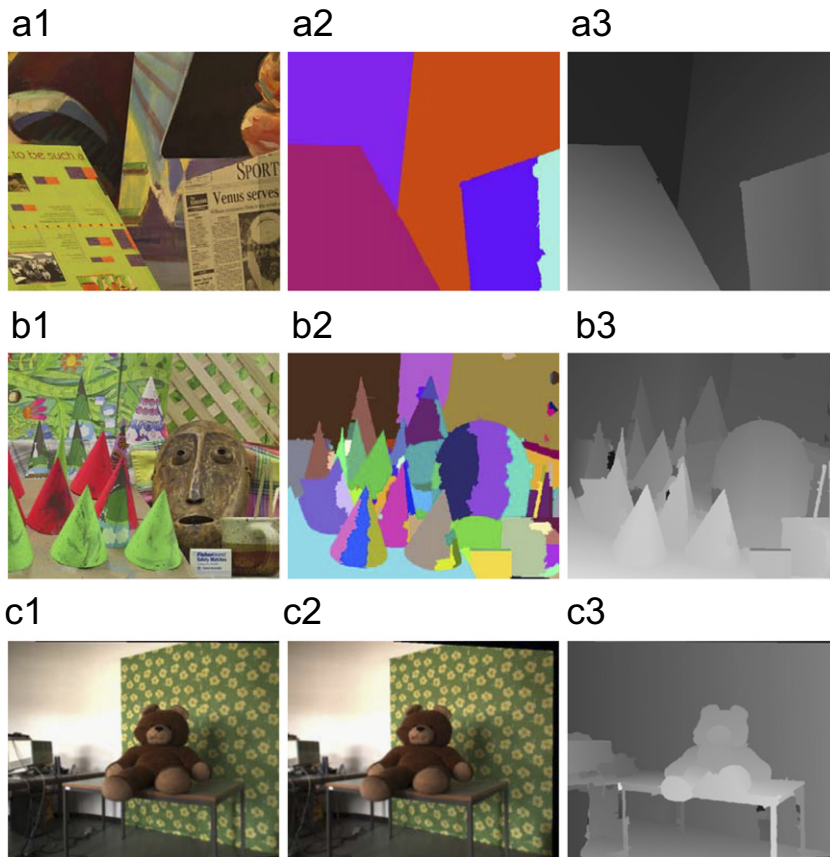


Fig. 11. Results on standard and self-recorded image pairs. (a1) Left image of the Venus test set. (a2) Disparity layers. (a3) Disparity map. (b1) Left image of the Cones test set. (b2) Disparity layers. (b3) Disparity map. (c1) Left image of a self-recorded image set. (c2) Right image. (c3) Disparity map.

whereas the disparity of the teddy, which has a more complex structure, is well reconstructed using more disparity layers. Furthermore, the algorithm was able to capture the thin structures represented by the legs of the table.

## 6. Conclusions

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 segment is then assigned to exactly 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 traditionally challenging for stereo algorithms.

Further research will concentrate on overcoming two limitations of our approach. The algorithm

currently describes the image disparity using a planar model. This may result in an oversimplification of the real disparity. However, the planar model could easily be replaced by a more sophisticated one without major changes in our implementation. A more severe problem is related to the segmentation assumption. As is the nature of an assumption, the segmentation constraint is not guaranteed to hold true. Our current remedy to this is to apply a strong oversegmentation. However, since this does not completely overcome this problem, our algorithm could benefit from an operation that allows segments to be split. It would also be interesting to develop a special purpose colour segmentation method that avoids (as far as possible) producing segments which overlap a depth discontinuity.

### Acknowledgments

The authors would like to thank Danijela Markovic for recording the stereo pair shown in Figs. 11(c1) and (c2). The authors are also grateful to Efstathios Stavrakis for pointing out the usefulness of a colour similarity function in the definition of the smoothness term in Eq. (10). Moreover, the authors wish to thank Allan Hanbury for proof-reading. Financial support for this work was obtained from the Austrian Science Fund (FWF) under project P15663. A preliminary conference version of this paper has appeared under [4].

### References

- [1] S. Birchfield, C. Tomasi, A pixel dissimilarity measure that is insensitive to image sampling, *IEEE Trans. Pattern Anal. Mach. Intell.* 20 (4) (1998) 401–406.
- [2] S. Birchfield, C. Tomasi, Multiway cut for stereo and motion with slanted surfaces, in: *International Conference on Computer Vision*, 1999, pp. 489–495.
- [3] M. Bleyer, Segmentation-based stereo and motion with occlusions, Ph.D. Thesis, Vienna University of Technology, 2006.
- [4] M. Bleyer, M. Gelautz, Graph-based surface reconstruction from stereo pairs using image segmentation, in: *SPIE Symposium on Electronic Imaging (Videometrics VIII)*, vol. 5665, 2005, pp. 288–299.
- [5] M. Bleyer, M. Gelautz, A layered stereo matching algorithm using image segmentation and global visibility constraints, *ISPRS J. Photogramm. Remote Sensing* 59 (3) (2005) 128–150.
- [6] Y. Boykov, V. Kolmogorov, An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision, *Trans. Pattern Anal. Mach. Intell.* 26 (9) (2004) 1124–1137.
- [7] Y. Boykov, O. Veksler, R. Zabih, Fast approximate energy minimization via graph cuts, *Trans. Pattern Anal. Mach. Intell.* 23 (11) (2001) 1222–1239.
- [8] C. Christoudias, B. Georgescu, P. Meer, Synergism in low-level vision, in: *International Conference on Pattern Recognition*, vol. 4, 2002, pp. 150–155.
- [9] D. Comaniciu, P. Meer, Distribution free decomposition of multivariate data, *Pattern Anal. Appl.* 1 (2) (1999) 22–30.
- [10] P.V. Fua, Combining stereo and monocular information to compute dense depth maps that preserve depth discontinuities, in: *International Joint Conference on Artificial Intelligence*, 1991, pp. 1292–1298.
- [11] L. Hong, G. Chen, Segment-based stereo matching using graph cuts, in: *Conference on Computer Vision and Pattern Recognition*, vol. 1, 2004, pp. 74–81.
- [12] H. Ishikawa, Exact optimization for markov random fields with convex priors, *Trans. Pattern Anal. Mach. Intell.* 25 (10) (2003) 1333–1336.
- [13] Q. Ke, T. Kanade, A subspace approach to layer extraction, in: *Conference on Computer Vision and Pattern Recognition*, 2001, pp. 255–262.
- [14] V. Kolmogorov, R. Zabih, Computing visual correspondence with occlusions using graph cuts, in: *International Conference on Computer Vision*, vol. 2, 2002, pp. 508–515.
- [15] V. Kolmogorov, R. Zabih, What energy functions can be minimized via graph cuts?, *Trans. Pattern Anal. Mach. Intell.* 26 (2) (2004) 147–159.
- [16] M. Lin, C. Tomasi, Surfaces with occlusions from layered stereo, *Trans. Pattern Anal. Mach. Intell.* 26 (8) (2004) 1073–1078.
- [17] A.S. Ogale, Y. Aloimonos, Stereo correspondence with slanted surfaces: critical implications of horizontal slant, in: *Conference on Computer Vision and Pattern Recognition*, 2004, pp. 568–573.
- [18] S. Roy, I. Cox, A maximum-flow formulation of the n-camera stereo correspondence problem, in: *International Conference on Computer Vision*, 1998, pp. 492–499.
- [19] D. Scharstein, View synthesis using stereo vision, *Lecture Notes in Computer Science (LNCS)* (1999) 1583.
- [20] D. Scharstein, R. Szeliski, A taxonomy and evaluation of dense two-frame stereo correspondence algorithms, *Int. J. Comput. Vision* 47 (1/2/3) (2002) 7–42 (<http://www.middlebury.edu/stereo/>).
- [21] D. Scharstein, R. Szeliski, High-accuracy stereo depth maps using structured light, in: *Conference on Computer Vision and Pattern Recognition*, 2003, pp. 195–202.
- [22] J. Sun, N.N. Zheng, H.Y. Shum, Stereo matching using belief propagation, *Trans. Pattern Anal. Mach. Intell.* 25 (7) (2003) 787–800.
- [23] H. Tao, H. Sawhney, R. Kumar, A global matching framework for stereo computation, in: *IEEE International Conference on Computer Vision*, 2001, pp. 532–539.
- [24] J. Wang, E. Adelson, Representing moving images with layers, *IEEE Trans. Image Process.* 3 (5) (1994) 625–638.
- [25] Y. Wei, L. Quan, Region-based progressive stereo matching, in: *Conference on Computer Vision and Pattern Recognition*, vol. 1, 2004, pp. 106–113.
- [26] J. Xiao, M. Shah, Motion layer extraction in the presence of occlusion using graph cuts, *IEEE Trans. Pattern Anal. Mach. Intell.* 27 (10) (2005) 1644–1659.

- [27] Y. Zhang, C. Kambhamettu, Stereo matching with segmentation-based cooperation, in: European Conference on Computer Vision, 2002, pp. 556–571.
- [28] C. Zitnick, T. Kanade, A cooperative algorithm for stereo matching and occlusion detection, *IEEE Trans. Pattern Anal. Mach. Intell.* 22 (7) (2000) 675–684.
- [29] L. Zitnick, S. Kang, M. Uyttendaele, S. Winder, R. Szeliski, High-quality video view interpolation using a layered representation, *ACM Trans. Graphics* 23 (3) (2004) 600–608 (<http://research.microsoft.com/users/mattu/>).