Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

ISPRS Journal of Photogrammetry & Remote Sensing 59 (2005) 128–150

PHOTOGRAMMETRY & REMOTE SENSING

www.elsevier.com/locate/isprsjprs

# A layered stereo matching algorithm using image segmentation and global visibility constraints

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 9 June 2004; received in revised form 14 February 2005; accepted 14 February 2005
Available online 7 April 2005

**Abstract**

This work describes a stereo algorithm that takes advantage of image segmentation, assuming that disparity varies smoothly inside a segment of homogeneous colour and depth discontinuities coincide with segment borders. Image segmentation allows our method to generate correct disparity estimates in large untextured regions and precisely localize depth boundaries. The disparity inside a segment is represented by a planar equation. To derive the plane model, an initial disparity map is generated. We use a window-based approach that exploits the results of segmentation. The size of the match window is chosen adaptively. A segment's planar model is then derived by robust least squared error fitting using the initial disparity map. In a layer extraction step, disparity segments that are found to be similar according to a plane dissimilarity measurement are combined to form a single robust layer. We apply a modified mean-shift algorithm to extract clusters of similar disparity segments. Segments of the same cluster build a layer, the plane parameters of which are computed from its spatial extent using the initial disparity map. We then optimize the assignment of segments to layers using a global cost function. The quality of the disparity map is measured by warping the reference image to the second view and comparing it with the real image. Z-buffering enforces visibility and allows the explicit detection of occlusions. The cost function measures the colour dissimilarity between the warped and real views, and penalizes occlusions and neighbouring segments that are assigned to different layers. Since the problem of finding the assignment of segments to layers that minimizes this cost function is $\mathcal{NP}$-complete, an efficient greedy algorithm is applied to find a local minimum. Layer extraction and assignment are alternately applied. Qualitative and quantitative results obtained for benchmark image pairs show that the proposed algorithm outperforms most state-of-the-art matching algorithms currently listed on the Middlebury stereo evaluation website. The technique achieves particularly good results in areas with depth discontinuities and related occlusions, where missing stereo information is substituted from surrounding regions. Furthermore, we apply the algorithm to a self-recorded image set and show 3D visualizations of the derived results.


*Keywords:* Stereo matching; Layered stereo; Colour segmentation; Clustering; Image warping

* Corresponding author. Tel.: +43 1 58801 18865; fax: +43 1 58801 18898.
*E-mail address:* bleyer@ims.tuwien.ac.at (M. Bleyer).

0924-2716/$ - see front matter © 2005 International Society for Photogrammetry and Remote Sensing, Inc. (ISPRS). Published by Elsevier B.V. All rights reserved.
doi:10.1016/j.isprsjprs.2005.02.008

## 1. Introduction

Given a pair of images taken from slightly different views, the task of a binocular stereo matching algorithm is to find points in both images that represent the same scene point. In a fully calibrated camera system, the images of both cameras can be resampled to fulfill the epipolar constraint. For an epipolar rectified image pair, each point in one image lies on the same horizontal scanline in the other image. The correspondence problem is therefore reduced to a one-dimensional search along corresponding scanlines. The offset between a pixel in the left and its corresponding pixel in the right image is called disparity. Once the disparity values are known, the world coordinates of a point can be determined by triangulation. Therefore, disparity is commonly used synonymously with inverse depth. Unfortunately, finding the correct disparity value for each image point is known to be difficult due to image noise, untextured regions and occlusions. Therefore, stereo matching is still an active topic of ongoing research. In order to simplify the problem, the vast majority of stereo algorithms implicitly or explicitly apply certain assumptions with the assumptions of *uniqueness* and *continuity* being the most frequently used. The uniqueness assumption states that at most a single unique match exists for each pixel. This holds true for scenes containing only opaque surfaces. The continuity assumption refers to the observation that disparity varies smoothly almost everywhere, except at depth discontinuities. In the following, we give a brief overview of important developments in stereo matching that were published over the last couple of years. The reader is referred to Scharstein and Szeliski (2002) for an extensive survey of prior work. According to the literature, stereo algorithms are divided between *local* and *global* methods depending on the optimization strategy they employ.

Local methods typically operate on windows that are shifted on the corresponding scanline in the second view to find the point of maximum correspondence. Local approaches do not impose any smoothness (or continuity) term, i.e., the matching score is independent of the disparity assignment of neighbouring pixels. This makes it difficult for them to capture the correct disparity in untextured regions. Window-based methods implicitly make the assumption of continuity by assuming constant disparity for all pixels inside the matching window. This assumption is broken at depth boundaries where occluded regions lead to erroneous matches, resulting in the familiar foreground fattening effect. Generally, the choice of an appropriate window size is a crucial decision. Small windows do not capture enough intensity variation to give correct results in less-textured regions. On the other hand, large windows tend to blur the depth boundaries and do not capture well small details and thin objects. This motivates the use of adaptive windows (e.g., Kanade and Okutomi, 1994; Hirschmüller et al., 2002). A three-dimensional data structure that records the matching score of each pixel for all possible displacements is commonly referred to as disparity space image (DSI). The DSI can be efficiently computed for fixed window sizes using an incremental approach that makes the computational complexity independent of the window size. This gives rise to real-time implementations (Hirschmüller et al., 2002; Mühlmann et al., 2002). In our work, too, we take advantage of the efficient incremental computation of the DSI in the generation of the initial disparity map. Furthermore, we use different window sizes starting with smaller windows in order to preserve fine image details wherever possible. Disparity estimates for less-textured regions are then derived by using larger window sizes.

Cooperative approaches (Zitnick and Kanade, 2000; Zhang and Kambhamettu, 2002; Mayer, 2003) locally compute matching scores using match windows. Nevertheless, they show "global behaviour" by iteratively refining the correlation scores using the uniqueness and continuity constraints. Zhang and Kambhamettu (2002) take advantage of image segmentation in the calculation of the initial matching scores. Furthermore, they exploit the results of the segmentation in their choice of local support area, preventing the support area from overlapping a depth discontinuity. Similarly to Zhang and Kambhamettu (2002), we use the output of image segmentation to propagate reliable disparity information inside a segment.

Stereo algorithms based on dynamic programming (Bobick and Intille, 1999; Birchfield and Tomasi, 1999b) belong to the global methods. They match each pair of horizontal scanlines independently. A path through a DSI slice is computed that optimizes a

global cost function, assuming the validity of the ordering constraint. The cost function usually minimizes the intensity differences and penalizes occlusions. A global optimum of the cost function can be found for the one-dimensional case by using dynamic programming. Since smoothness across scanlines is not enforced, the computed disparity maps suffer from horizontal "streaks". Nevertheless, dynamic programming approaches are computationally inexpensive and candidates for real-time implementations.

Other global approaches optimize a two-dimensional cost function, consisting of a data term that measures the pixel dissimilarity and a smoothness term that penalizes neighbouring pixels assigned to different disparities. The optimization of this cost function is shown to be $\mathcal{NP}$-complete. Boykov et al. (2001) present an efficient greedy algorithm based on graph cuts. Their work was extended by Kolmogorov and Zabih (2002), who enforce the uniqueness constraint to handle occlusions. The energy minimization framework of Birchfield and Tomasi (1999a) represents the scene by a set of planar layers. Similarly to our approach, they alternate between a layer fitting and a layer assignment step. In the layer fitting step, the planar model of each surface is computed based on the current assignment of pixels to layers by minimizing a cost function. In the layer assignment step, the spatial extent of each layer is then determined by optimizing a cost function using a graph-based method. Lin and Tomasi (2003) extend this work with the most significant differences being the strictly symmetrical treatment of input images and the use of a spline model for layers.

In this work we propose a global stereo algorithm that represents the scene as a collection of planar layers. As a result we obtain a piecewise smooth surface reconstruction as well as real-valued disparity estimates that provide a high precision. Our algorithm explicitly addresses the problems of untextured regions and occlusions. Large untextured regions are handled by applying colour segmentation to the reference image. Smoothness inside the derived segments is enforced by the use of a planar model representing each segment's disparity. Furthermore, colour segmentation allows the accurate localization of depth discontinuities. Occlusions in the reference and in the second view are detected and handled in a layer assignment step. We also model smoothness across segments. Disparities are propagated along image segments using a greedy algorithm similar to that presented by Tao and Sawhney (2000).

## 2. Algorithmic outline

The overall algorithm can be divided into several major steps that are summarized in Fig. 1. Input to our matching algorithm are two stereo images in epipolar geometry. The first processing step is the segmentation of the reference image into regions of homogeneous colour, as described in Section 3.1. Since discontinuities in the disparity map are usually reflected by discontinuities in the colour information, the borders of the segmented regions can be considered as a set of candidates for the boundaries of the disparity layers that we aim to compute. We calculate an initial disparity map using a window-based correlation technique. This process is explained in more detail in Section 3.2. In the next step, we create an initial plane representation for each extracted segment by robust fitting of a planar surface to the correlation-based disparity values inside each individual segment (Section 3.3).

The computed segments along with their plane description are the starting point for an iterative

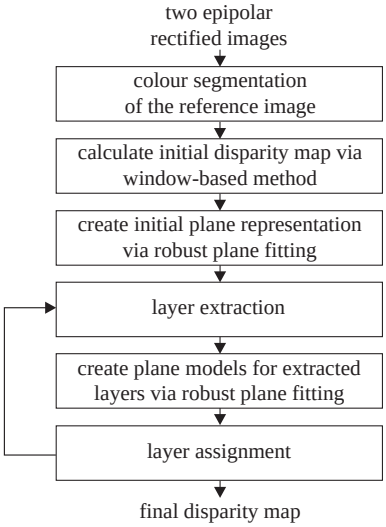


Fig. 1. Overview of the algorithm.

procedure in which segments are assigned to layers, which are groups of segments that can be approximated by one and the same planar equation. The iterative assignment starts with a *layer extraction* module (see Fig. 1) based on mean-shift clustering, which we describe in Section 4. The advantage of the layered approach is illustrated in Fig. 2. The planar models computed by fitting a plane to the disparity values derived from the initial disparity map are not very robust in small segments as a consequence of the small spatial extent over which the plane was calculated. This is sketched in Fig. 2a. A robust planar description of each layer is derived by fitting a plane over the larger region formed by all segments belonging to that particular layer, as shown in Fig. 2b.

The last block in the iteration loop from Fig. 1 is the layer assignment module, which we explain in Section 5. During this step, we try to improve the current solution based on a cost function that measures the quality of the current layer assignment. Based on the observation that erroneous assignments of segments to layers tend to arise more frequently along the layer borders rather than in their central regions, we test for each border segment whether a possible assignment to another layer might produce lower costs (i.e., a better solution). Based on the outcome of this hypothesis testing, new layers are formed by the layer extraction module during the next iteration step. The algorithm terminates if the costs could not be improved for a fixed number of iterations.

For the sake of clarity, we summarize the basic terms we use throughout this paper. Segments are regions of homogeneous colour that are computed during the initial colour segmentation step. *Layers* are groups of segments (and therefore usually larger than individual segments) that can be approximated by one and the same planar equation. The *layer extraction* module computes new layers using mean-shift clustering. During the first iteration step, the individual segments are input to the clustering algorithm. In subsequent iteration steps, the clustering algorithm seeks to merge previously defined layers, which may have been modified in the layer assignment module, into larger layers. The *layer assignment* module tries

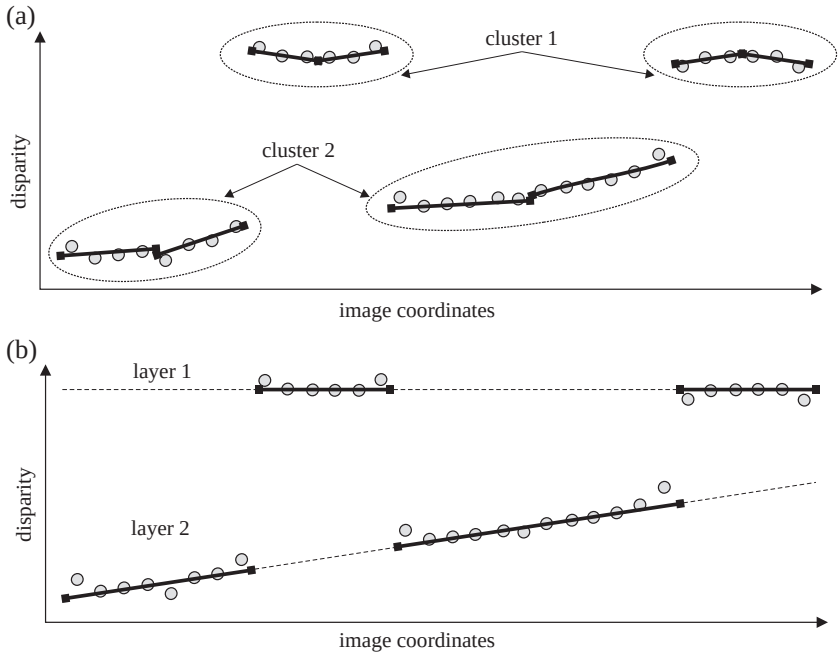


Fig. 2. Robustness of the layered representation. The less robust plane approximation of the individual segments illustrated in (a) is substituted by the more robust layer representation achieved by clustering as shown in (b).

to improve the current solution by assigning border segments to other neighbouring layers. During the iteration loop, the generated solution of lowest costs is recorded and returned as the final output of the algorithm.

## 3. Colour segmentation and planar model

### 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 (Tao and Sawhney, 2000; Ke and Kanade, 2001; Zhang and Kambhamettu, 2002), which holds true for most natural scenes. This assumption is incorporated by applying colour segmentation to the reference image and by using a planar model to represent the disparity inside the derived segments. It is generally safer to oversegment the image to ensure that this assumption is met. In principle, any algorithm that divides the reference image into regions of homogeneous colour can be used for the proposed stereo algorithm. Our current implementation uses a mean-shift-based segmentation algorithm that incorporates edge information as proposed by Christoudias et al. (2002). The resulting colour segmentation for a well-known stereo pair from the University of Tsukuba is shown in Fig. 3. Pixels belonging to the same segment are assigned the same colour. To derive the desired plane models we first compute an initial disparity map and use the computed disparity values to fit the plane for each segment.

### 3.2. Initial disparity map

We compute an initial disparity map using a local window-based method that exploits the results of the image segmentation and operates on different window sizes. We benefit from the image segmentation by exploiting the assumption of smoothly varying disparities inside a segment as introduced previously. Operating on different window sizes allows us to combine the advantages of both small and large windows. The decision of which window size to use for which region is driven by the data.

Initially, we start with a small $3 \times 3$ window. The window centered on a pixel in the left image is shifted along the corresponding scanline in the right view to find the displacement of minimum dissimilarity. To measure the dissimilarity of two pixels, we compute the summed up absolute differences of their RGB-values. We compute the DSI using an efficient incremental approach described by Mühlmann et al. (2002). The disparity of a pixel $d_{x,y}$ at coordinates $(x, y)$ is then derived from the DSI by using

$$d_{x,y} = \underset{D_{\min} \le d \le D_{\max}}{\operatorname{argmin}} \; \mathrm{DSI}(x, y, d) \tag{1}$$

with $D_{\min}$ and $D_{\max}$ denoting the minimum and maximum allowed disparity. This local optimization strategy is not able to produce correct disparity estimates in untextured or occluded regions. We filter out unreliable pixels by applying left–right consistency checking. An established match is only valid, if the matched point in the right image points back to the pixel in the left view. As Fua (1991) points out, the left–right consistency check is known to fail, if

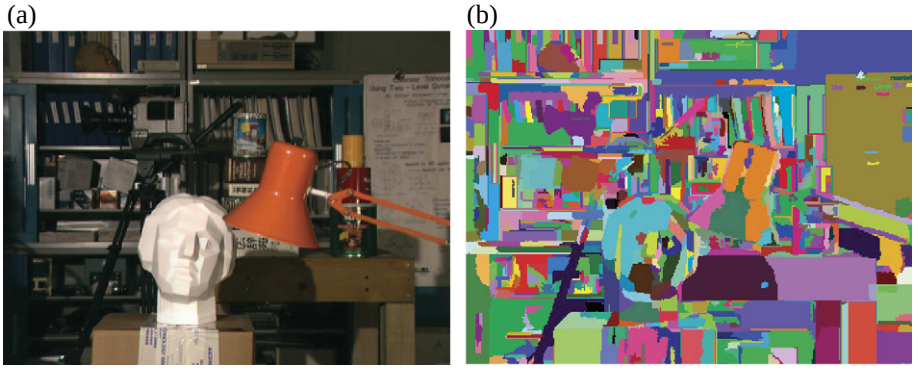


Fig. 3. Colour segmentation. (a) Left image. (b) Computed colour segmentation.

the areas to be correlated have little texture or in the presence of an occlusion. We further reject points with insufficient support by removing connected regions of equal disparity smaller than a predefined threshold.

We then reduce the search scope for each segment depending on a measurement of the segment's confidence. A similar approach was taken by Zhang and Kambhamettu (2002). We follow their idea to measure the reliability of a segment's disparity information by the density of valid points. Segments having a ratio of valid points larger than 50% are labelled as being *reliable*. We search the points with minimum disparity $d_{\min_i}$ and maximum disparity $d_{\max_i}$ inside the $i$th *reliable* segment

$$d_{\min_i} = \min_{(x,y)\in V_i} d_{x,y} \quad d_{\max_i} = \max_{(x,y)\in V_i} d_{x,y} \tag{2}$$

with $V_i$ being the set of all valid points of the corresponding segment. We then compute the best correlation score for all unassigned pixels in a reduced search range

$$d_{x,y} = \underset{d_{\min_i}-t_{\text{tolerance}}\le d\le d_{\max_i}+t_{\text{tolerance}}}{\operatorname{argmin}} \mathrm{DSI}(x,y,d)$$
$$\forall (x,y)\in U_i \tag{3}$$

with $U_i$ denoting the set of the $i$th *reliable* segment's unassigned points and $t_{\text{tolerance}}$ representing a small value for tolerance. In our implementation, the threshold $t_{\text{tolerance}}$ is set to the fixed value of one pixel. The reduction in search range helps to capture points with little texture information, for which the correct match was overwhelmed by noise due to the larger search scope (Zhang and Kambhamettu, 2002). Furthermore, it allows to propagate reliable disparity information inside the segment. This procedure directly reflects the assumption of smoothly varying disparity inside segments. We further apply a left–right consistency check using the reduced search range and remove points with insufficient support.

More matches are then gathered by increasing the window size leaving the already found valid points unchanged. First, we use the full search range for segments with density of valid points < 50%. We then determine again the reliability of each segment. For all reliable segments, we reduce the search scope. Finally, the window size is further increased and the process is repeated. Since we are starting with a small $3\times 3$ window, our approach is able to capture thin structures and generates a detailed disparity map. Disparity information for less-textured regions is then obtained by the use of larger windows. We therefore combine advantages of both strategies. Fig. 4 shows a block diagram of the described algorithm. The initial disparity map calculated for the Tsukuba image pair using a $3\times 3$, $5\times 5$ and $7\times 7$ window is presented in Fig. 5. Higher disparity values are encoded by bright values. Black points represent invalid pixels for which no disparity information is estimated. The calculated initial disparity map serves to obtain the planar model of a segment and does not represent the final result of our algorithm.

### 3.3. Planar model fitting

Once we have calculated the initial disparity map, we use it to derive the planar model of each segment. We represent a segment's disparity by a function

$$d(x,y) = ax + by + c \tag{4}$$

with $x$ and $y$ being image coordinates and $a$, $b$ and $c$ being the plane parameters. To derive a segment's

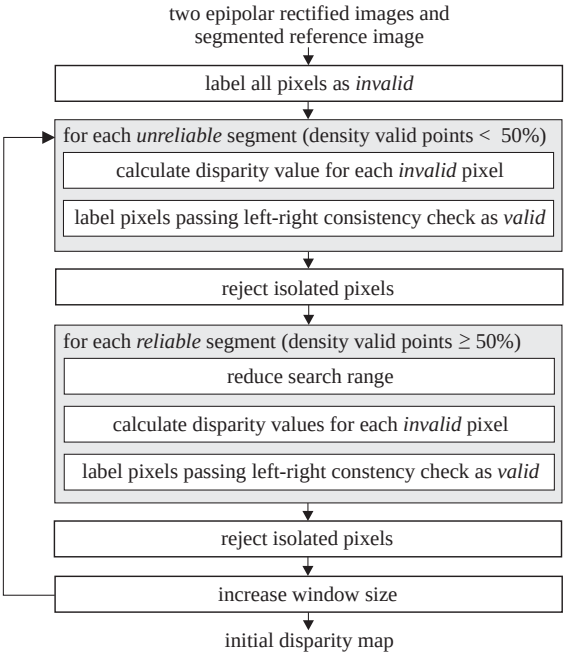


Fig. 4. Block diagram of the algorithm creating the initial disparity map.

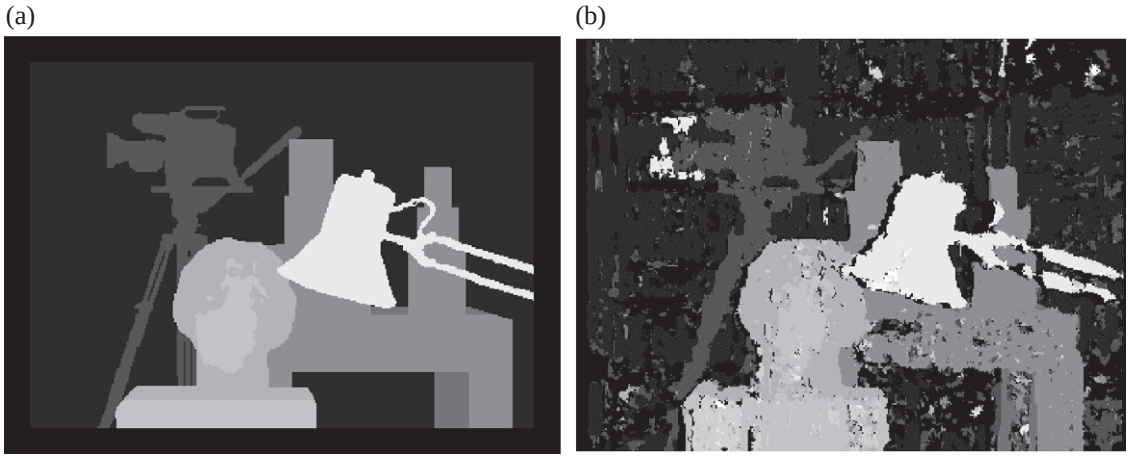


Fig. 5. Initial disparity map. (a) Ground truth provided with image pair. (b) Computed initial disparity map. Invalid points are coloured black.

plane parameters, we apply least squares error fitting to all valid points inside the segment. The least squared error solution is then given by solving

$$\begin{bmatrix} \sum_{i=1}^{m} x_i^2 & \sum_{i=1}^{m} x_i y_i & \sum_{i=1}^{m} x_i \\ \sum_{i=1}^{m} x_i y_i & \sum_{i=1}^{m} y_i^2 & \sum_{i=1}^{m} y_i \\ \sum_{i=1}^{m} x_i & \sum_{i=1}^{m} y_i & \sum_{i=1}^{m} 1 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{m} x_i d_i \\ \sum_{i=1}^{m} y_i d_i \\ \sum_{i=1}^{m} d_i \end{bmatrix} \tag{5}$$

with $m$ being the number of valid points inside the segment, $x_i$ and $y_i$ being the coordinates of the $i$th valid point and $d_i$ its corresponding disparity value. Unfortunately, the method of least squared errors is sensitive to outliers. Although we already try to remove outliers in the computation of the initial disparity map, there may still be erroneous points due to edge fattening, repetitive patterns or noisy image data. Fig. 6 illustrates the implemented plane fitting algorithm that is robust to outliers. To derive a segment's planar description we fit a plane to all valid points of the initial disparity map inside the segment. This is shown in Fig. 6a. Not every image coordinate is represented by a point in disparity space because of invalid points in the initial disparity map. There are three valid points of high disparity representing outliers that attract the computed plane. To eliminate outliers we search all valid points of the segment that have a distance to the computed plane that is larger than the predefined threshold $t_{\text{outlier}}$ and reject them as shown in Fig. 6b. Formally expressed, the new set of valid points $V'$ is derived by

$$V' = \{(x_i, y_i) \in V \mid d_i - (ax_i + by_i + c) \leq t_{\text{outlier}}\} \tag{6}$$

with $V$ being the set of all valid points inside the segment and $t_{\text{outlier}}$ being a threshold that is set to the constant value of one pixel for all our computations. A new plane is then fitted to the points in $V'$ using Eq. (5). This process is then iterated until

$$(a' - a)^2 + (b' - b)^2 + (c' - c)^2 \leq t_{\text{convergence}} \tag{7}$$

with $a'$, $b'$ and $c'$ being the parameters of the new plane, $a$, $b$ and $c$ being the parameters of the plane that was derived in the previous iteration and $t_{\text{convergence}}$ being a very small value (typically $10^{-6}$). Fig. 6c illustrates the plane derived after removal of three outliers.

## 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 the planar models should be very similar, as long as the surface can be well approximated as a plane. Following this idea, we define a measurement for the dissimilarity of two disparity planes and use this measurement in a clustering method to identify segments belonging to the same surface.

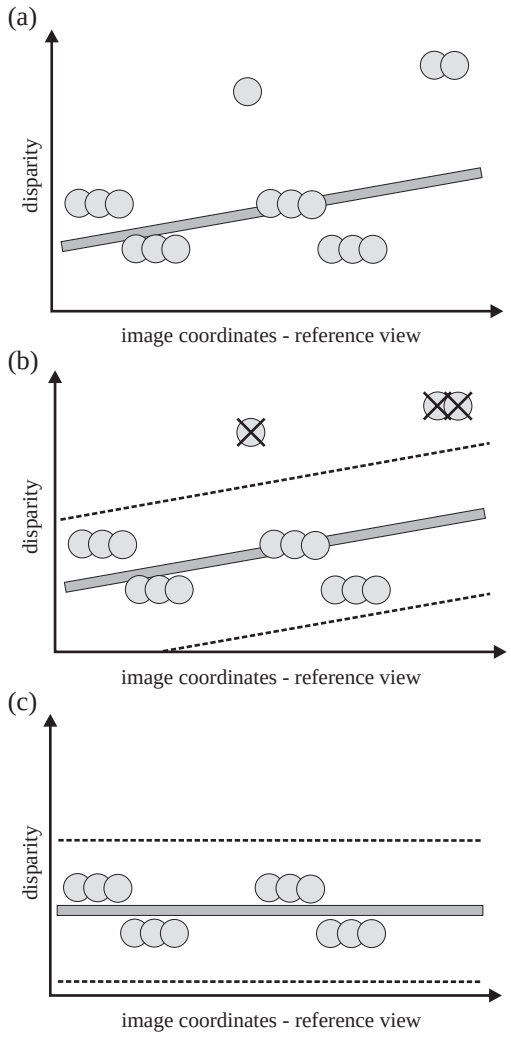


Fig. 6. Robust plane fitting. (a) Initial computed plane. (b) Removal of outliers. (c) Final plane.

We exploit a distance measurement originally introduced by Tao et al. (2001). The similarity of two disparity segments is measured by calculating the intersection point of the normal vector on the first segment's plane, originating from that segment's center of gravity, with the disparity plane of the second segment. We then compute the length of the vector from the first segment's center of gravity to the point of intersection, which is denoted by $\mathrm{dis}_1$. For symmetry, we also compute $\mathrm{dis}_2$, which is the distance between the second segment's center and the first segment's disparity plane. The term $\mathrm{dis}_1+\mathrm{dis}_2$ then describes the amount of dissimilarity between two planes. We illustrate this process in Fig. 7 for the two-dimensional case. We believe that this measurement is specifically well suited to the task of clustering disparity planes, since it incorporates spatial information as well as the plane parameters.

We project each segment into a five-dimensional feature space, consisting of the three plane parameters and two spatial parameters represented by the $x$ and $y$ components of the center of gravity. We do not project the $z$ component, since it can be deduced from the other five parameters. We employ the mean-shift algorithm (Comaniciu and Meer, 1999), which we modify to embed the described plane dissimilarity measurement, to extract clusters in this feature space. A specific advantage of the mean-shift algorithm is that the number of clusters does not need to be known beforehand. To apply the mean-shift to a data point $y_k$ at iteration $k$, we determine its neighbourhood $N(y_k)$ by

$$N(y_k) = \{x \in \mathrm{DP} | \mathrm{dis}(x, y_k) \leq r\} \tag{8}$$

with DP being the set of all data points, dis denoting the plane dissimilarity function and $r$ being the radius of the mean-shift. We then compute the mean value of all data points inside the neighbourhood. Since data points represent segments covering areas of different sizes, the reference image is not uniformly sampled. A layer containing a rich amount of texture and therefore a large number of segments will be represented by more region samples than a layer representing large homogeneously coloured regions. The layer of homogeneous colour may not have enough samples to form a dense cluster in feature space. We overcome this problem by weighting each data point according to the area of the segment it describes. We then derive the location of the shifted data point $y_{k+1}$ by computing the weighted mean value

$$y_{k+1} = \sum_{x \in N(y_k)} \frac{a_x}{A} x \tag{9}$$

with $a_x$ being the number of pixels inside the segment described by the data point $x$ and $A$ being the summed up areas of all segments inside the neighbourhood $N(y_k)$. The mean-shift is then iteratively applied until the magnitude of the shift becomes smaller than the

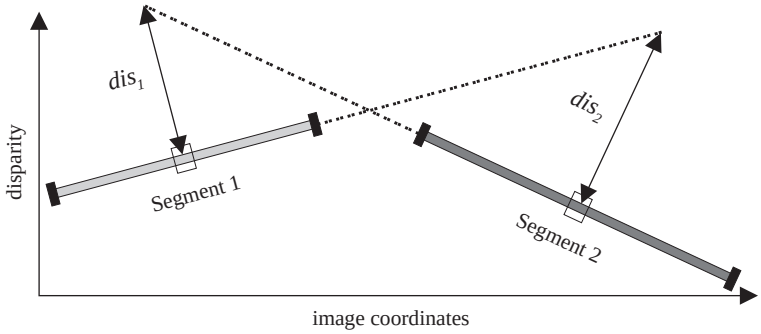


Fig. 7. Measuring the dissimilarity of two disparity planes.

threshold $t_{\text{convergence}}$ set to a very small number (typically $10^{-6}$). The data point is thereby shifted to a local density maximum. This procedure is applied for each data point. In the fusion step, we then investigate the points of convergence to derive the points building a cluster. Points having a distance smaller than a threshold, set to $r/2$ in our implementation, are merged to form a single cluster. The distance is again computed by using the plane dissimilarity measurement defined above.

Members of the same cluster build a layer. For deriving a layer's plane equation, we use the initial disparity map. Robust plane fitting is applied to the valid points of all segments belonging to the layer.

## 5. Layer assignment

We try to improve the current solution by optimizing the assignment of segments to layers. A cost function that uses image warping is designed to measure the quality of the current assignment. We describe an efficient hypothesis testing framework in order to optimize the specified cost function.

### 5.1. Cost function

We measure the quality of a disparity map by warping the reference view according to the current disparity map. The basic idea behind this procedure is that if the disparity map was correct, the warped image should be very similar to the real image from this viewpoint. We implemented a warping procedure based on a Z-buffer to explicitly model visibility. To obtain the second view, we warp each segment according to its current disparity plane. A naive approach would reconstruct the second view by projecting each individual pixel of the reference image into the second view using its disparity value. Consequently, pinholes would occur in the warped image for areas that are undersampled in the reference image. Therefore, a more elaborate strategy is used. A segment is represented by the set of all its horizontal scanline runs. A segment's scanline runs are derived by tracing each horizontal scanline from left to right. Whenever the left border of the segment is encountered, the corresponding coordinates are stored as the starting point of a run. Whenever the segment's right border is hit, the corresponding coordinates are stored as the ending point of the run. The warped view of a segment is generated by transforming all its scanline runs. Therefore, the coordinates of the starting and ending point in the warped view are computed using the segment's planar model. For all points between the warped starting and ending point, we compute the exact coordinates in the reference image according to the segment's disparity plane. The colour values for those pixels are then derived by linear interpolation of the colour values of the pixels left and right to the exact position in the reference view. This process is illustrated in Fig. 8.

We reconstruct the second view by warping all segments of the reference view. In this procedure, pixels from the second view may receive a contribution from more than a single segment. For those pixels, we have to make a decision concerning visibility. We use a Z-buffer representing the second view, which naturally enforces visibility. Each Z-buffer cell corresponds to a single pixel of the right

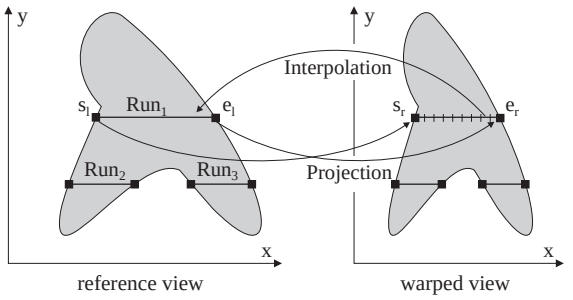


Fig. 8. Warping a segment to the second view according to its disparity plane.

view. If a Z-buffer cell contains more than one pixel, only the pixel with the highest disparity is visible, since it is the one closest to the camera. The others are therefore occluded in the second view. Furthermore, we are also able to detect pixels occluded in the reference image, since they correspond to empty Z-buffer cells. We illustrate this in Fig. 9.

We will now use the observations made above to formulate a cost function which is designed to measure the quality of a derived disparity map. The first term of the cost function is based on the idea that a good disparity map should produce a warped view with a high similarity to the real second image. Translated to our cost function, we calculate the colour dissimilarity between the warped and real views for all pixels visible in the warped image. According to the literature, we refer to this term as data term that is defined by

$$T_{\text{data}} = \sum_{p \in \text{Vis}} \text{dis}(W(p), R(p)) \tag{10}$$

with $W(p)$ denoting the pixel $p$ in the warped image and $R(p)$ being the pixel $p$ in the real second view. The set of visible pixels Vis is defined by the union of all pixels that have the highest disparity in their individual Z-buffer cells. Formally, the set Vis is computed by

$$\text{Vis} = \left\{ \cup_{x,y} p \in Z_{x,y} | \forall q \in Z_{x,y} : d(p) > d(q) \vee p = q \right\} \tag{11}$$

with $Z_{x,y}$ denoting the set of all pixels inside the Z-buffer cell at image coordinates $x$ and $y$ and $d(p)$ being the disparity of pixel $p$. The colour dissimilarity function $\text{dis}(p_i,p_j)$ is defined as the summed up absolute differences of RGB values of pixels $p_i$ and $p_j$. We write

$$\begin{aligned} \text{dis}(p_i, p_j) = &|r(p_i) - r(p_j)| + |g(p_i) - g(p_j)| \\ &+ |b(p_i) - b(p_j)| \end{aligned} \tag{12}$$

with $r(p)$ being the red, $g(p)$ being the green and $b(p)$ being the blue colour components of pixel $p$.

The second term of the cost function accounts for occlusions. It is necessary for our cost function to penalize occlusions, since otherwise declaring all pixels as occluded would form a trivial optimum. We therefore introduce an occlusion term that penalizes occlusions in the left and right images. This term is defined by

$$T_{\text{occlusion}} = (|\text{Occ}_\text{R}| + |\text{Occ}_\text{L}|)\lambda_{\text{occ}} \tag{13}$$

with $\text{Occ}_\text{R}$ being pixels that are occluded in the right view, $\text{Occ}_\text{L}$ denoting occlusions in the left image and $\lambda_{\text{occ}}$ being a constant penalty for occlusion. The set $\text{Occ}_\text{R}$ is defined by the union of all pixels that are occluded by a pixel of higher

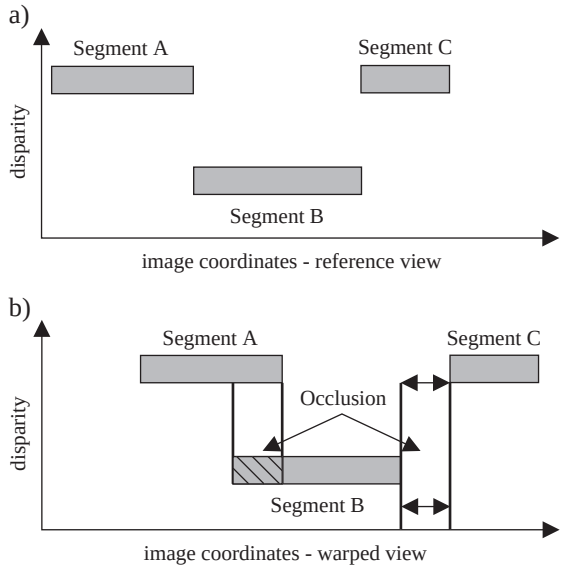


Fig. 9. The warping operation. (a) Segments and corresponding disparity in the reference view. (b) Segments warped to the second view according to their disparity planes.

disparity in their individual Z-buffer cells. This set is computed by

$$\mathrm{Occ}_{\mathrm{R}} = \left\{ \cup_{x,y}\, p \in Z_{x,y} | \exists\, q \in Z_{x,y} : d(p) < d(q) \right\}. \quad (14)$$

The set of occlusions in the left image $\mathrm{Occ}_{\mathrm{L}}$ is then defined by the union of all empty Z-buffer cells given by

$$\mathrm{Occ}_{\mathrm{L}} = \left\{ \cup_{x,y}\, Z_{x,y} | Z_{x,y} = \theta \right\}. \quad (15)$$

The last term of the cost function motivates smoothness across segments. We introduce a discontinuity penalty that is applied when two neighbouring pixels (in 4-connectivity) are assigned to different layers in the reference image. We define this term by

$$T_{\mathrm{smoothness}} = \sum_{(p_i,p_j)\in N} \begin{cases} \lambda_{\mathrm{disc}} & : \quad \mathrm{layerid}(p_i) \neq \mathrm{layerid}(p_j) \\ 0 & : \quad \text{otherwise} \end{cases} \quad (16)$$

with layerid($p$) being a function that returns the id of the disparity layer to which the segment containing pixel $p$ is assigned and $\lambda_{\mathrm{disc}}$ being a constant penalty for discontinuity. The set $N$ defined for the left image $I_{\mathrm{L}}$ denotes pairs of pixels $(p_i, p_j)$ with $p_i, p_j \in I_{\mathrm{L}}$ and $i<j$ that are neighbours in 4-connectivity.

Putting this together we finally obtain the cost function

$$C = T_{\mathrm{data}} + T_{\mathrm{occlusion}} + T_{\mathrm{smoothness}} \quad (17)$$

measuring the quality of a disparity map. We are therefore searching an assignment of layers to segments that minimizes $C$.

### 5.2. Optimization

Unfortunately, finding the assignment that minimizes $C$ is non-trivial. Given $S$ segments and $L$ distinct layers there are $S^L$ different possible assignments. The large solution space indicates the complexity of the problem. Moreover, finding the layer assignment with minimum value for $C$ is shown to be $\mathcal{NP}$-complete and therefore not solvable by a complete algorithm in finite time. In our approach, we employ an efficient greedy search strategy to find a local optimal solution. This search strategy is similar to that used by Tao and Sawhney (2000), although they do not optimize an explicit cost function, but always take the local optimal decision that minimizes colour dissimilarity between the real and warped views until convergence.

The basic idea behind the algorithm is to propagate correct disparity information from neighbouring segments. Segments can be assigned to planes giving poor disparity estimates, since a segment can be affected by occlusion or may not have enough texture information to allow correct disparity estimation. Nevertheless, the chances for a neighbouring segment to be assigned to the correct disparity model are high, since usually disparity varies smoothly, except at depth boundaries. We exploit this idea in a hypothesis testing framework. For a segment, we hypothesize that its current layer assignment is wrong and a layer of a neighbouring segment better describes the segment. To test this hypothesis, we replace the plane model of the current segment by the neighbouring layer's plane equation. We then warp the reference image to the second view according to the current layer assignment and evaluate the cost function. If the costs are improved, the hypothesis is accepted and rejected otherwise. For a segment, we test the hypotheses of all neighbouring layers as shown in Fig. 10. In this figure the segment $S_1$ has five neighbouring segments assigned to layers 1, 2 and 3, which we refer to as the neighbouring layers of $S_1$. We avoid testing layer 1 on $S_1$, since this is the current assignment. The layer hypotheses of layers 2 and 3 need to be checked. Although there may be a large number of neighbouring segments, the layer neighbourhood is usually very small. Consequently, the number of layer hypotheses that need to be checked is small. If a segment is surrounded only by segments assigned to the same layer as the segment, which is

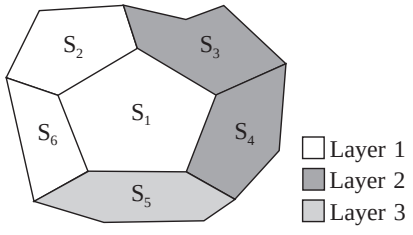


Fig. 10. Hypothesis testing.

usually the case for the majority of segments, no tests need to be applied at all. These observations represent major arguments for the algorithm's efficiency.

We embed the ideas of hypothesizing neighbouring layers into a greedy algorithm as follows. In the initial solution, we use the layer assignment derived from the layer extraction step. For each segment, we test the neighbouring layer hypotheses as described above. In the testing phase, the assignment of all other segments remains fixed. If there are layers generating smaller costs than the current solution, we record the one giving the largest improvement. Otherwise, the current assignment was found to be the best, and we record this assignment. After all segments have been tested, every segment gets assigned to its recorded layer. This process is then iterated and terminates if there has not been an improvement of costs for a fixed number of iterations. The generated solution with lowest costs is returned. Keeping the segments fixed during the hypothesis testing stage and updating them after all segments have been checked makes the algorithm independent of the order of applied operations. The greedy nature of the algorithm is reflected by always picking the layer hypothesis that locally gives the highest improvement of costs. An aspect concerning the computational efficiency of the proposed algorithm is that we only need to test segments if their neighbourhood has changed in the previous iteration. Otherwise, we would unnecessarily repeat the tests from the previous iteration without getting new results. Furthermore, since only small parts of the warped view are changed in the hypothesis testing, it would not be efficient to always warp the whole image. We therefore employ an incremental image warping procedure described in Appendix A. The block diagram of the greedy algorithm is shown in Fig. 11.

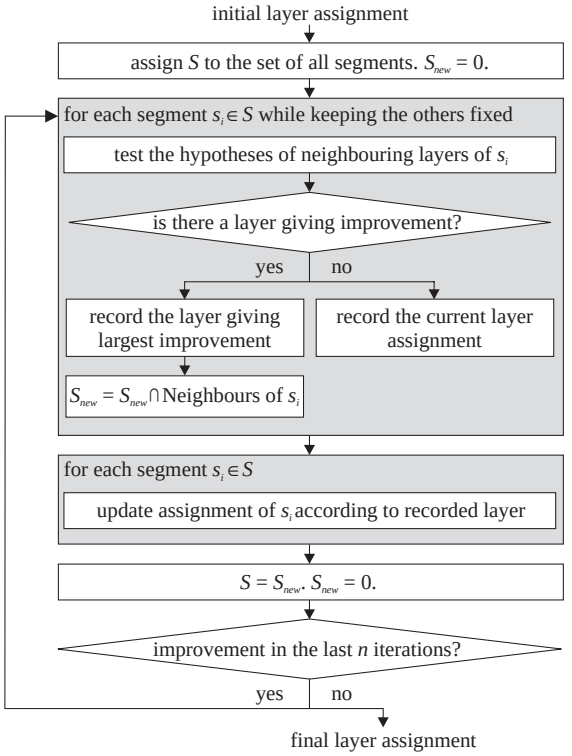


Fig. 11. Block diagram of the greedy algorithm.

## 6. Experimental results

We evaluated our algorithm using the test bed proposed by Scharstein and Szeliski (2002). The authors provide a set of four test pairs along with the corresponding ground truth. Researchers who want to participate in the test are asked to run their stereo algorithms on these image pairs using constant parameter settings. The resulting disparity maps are then compared against the corresponding ground truth. For quantitative evaluation, Scharstein and Szeliski (2002) measure the percentage of unoccluded pixels whose absolute disparity error is greater than one. The evaluated stereo algorithms are then ranked according to this error metric. We applied our algorithm to the evaluation sets and submitted the results to the online version of the test bed. At the time of writing, our method was ranked as having the second best overall performance among 30 different stereo algorithms tabulated. In the following, we show and discuss in more detail the results obtained for two of the four test sets. For the examples shown in this section, disparity maps are created using individual parameter values. We discuss the sensitivity of results to different settings of the parameters in Appendix B. For additional results as well as for results achieved with constant parameter values, the reader is referred to the Middlebury Stereo Vision website. Furthermore, we present disparity maps for a more complex scene that was taken from Scharstein and Szeliski (2003) and for a self-recorded stereo pair.

The first image pair we ran our algorithm on is the "head and lamp" data set of the University of Tsukuba, which became a standard test set for the stereo community. The image pair is presented in Fig. 12a and b. It shows a rather complex scene containing untextured regions (e.g., table) and thin objects (e.g., lamp arm), which make it hard for a stereo algorithm to capture the correct disparity information. The hand-labelled ground truth for the left image is shown in Fig. 12c. The presented disparity maps are scaled by a factor of 16 for visualization, i.e., a disparity value of one pixel is mapped to the gray value 16. We present the layers that were computed by our algorithm in Fig. 12d. Pixels belonging to the same layer are assigned to the same colour in the figure. Our algorithm divides

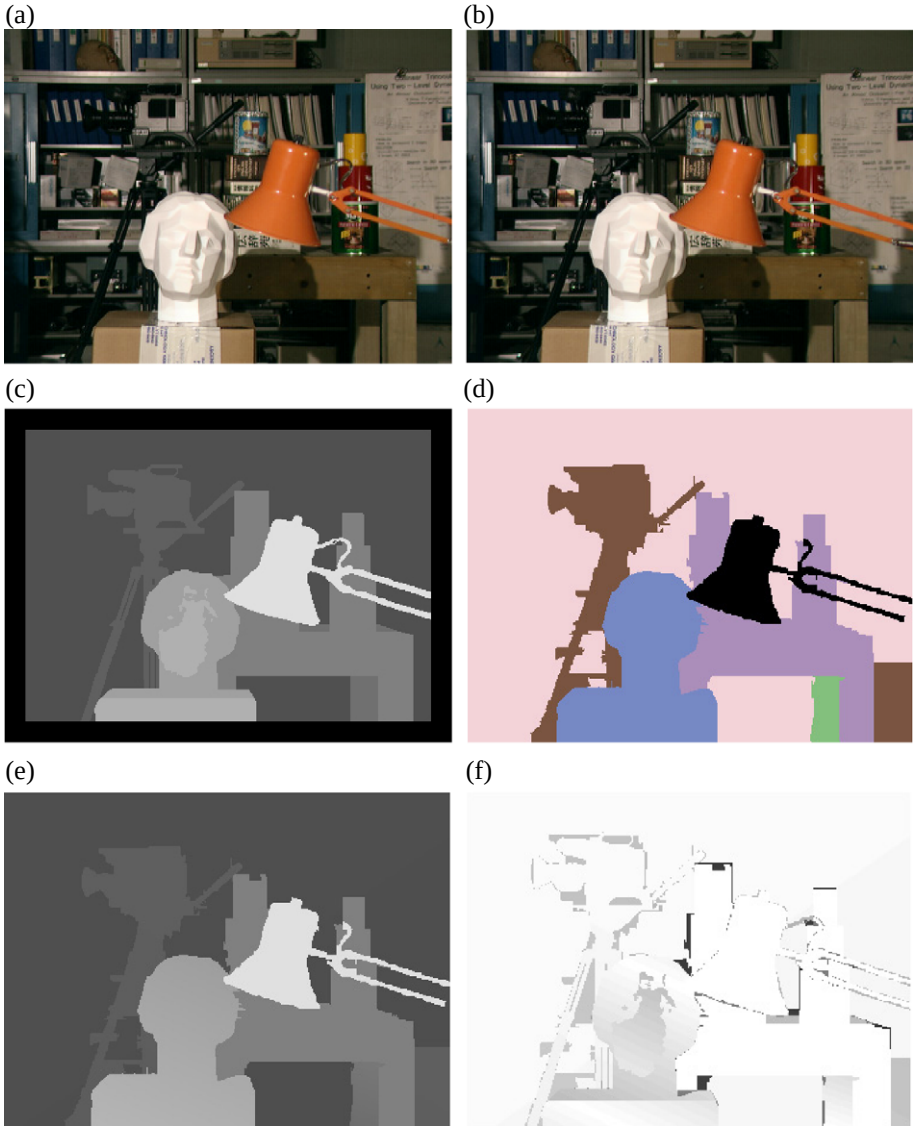


Fig. 12. Results for the Tsukuba test set. (a) Left image. (b) Right image. (c) Ground truth provided with image pair in the geometry of the left image. The presented disparity maps are scaled by a factor of 16 for visualization. (d) Computed layers. (e) Computed disparity map. (f) Absolute errors scaled by a factor of 64.

the reference image into six layers. Although we do not aim for a semantic segmentation, the derived layers correspond well to objects of the real world (head, lamp, camera, table, leg of the table, and background). The computed disparity map is then presented in Fig. 12e. We used the following parameter settings: $r=0.8$, $\lambda_{occ}=20.0$ and $\lambda_{disc}=20.0$. To visualize the quality of the derived disparity map we compare it against the ground truth in Fig. 12f. We thereby show the absolute error with darker pixels representing higher deviations from the ground truth. White pixels indicate a perfect correspondence between the computed disparity map and the ground truth. We applied a scaling factor of 64 to the computed errors and inverted the image for better visibility. Apart from some errors occurring at depth borders, which are mainly caused by colour segments that overlap depth boundaries, small errors appear on the head where the planar representation oversimplifies the real surface. To get a more accurate result for this region of the image it would be advantageous to set the mean-shift radius $r$ to a lower value. The head would then be reconstructed by a larger number of layers. However, this would also lead to a less robust reconstruction of the background, which would then be represented by more than one layer.

In Fig. 13 we compare the computed disparity map against the results generated by some of the best-performing stereo algorithms tabulated on the Middlebury Stereo Vision website. For comparison, we use two different error metrics. The first one computes the percentage of wrong *unoccluded* pixels exceeding a specified disparity error threshold. This corresponds to the metric used in Scharstein and Szeliski (2002) when the threshold is set to one. For the second error metric, we do not exclude occluded pixels from the evaluation and compute the percentage of *all* erroneous pixels. For both error measurements, we only consider pixels for which ground truth is available and plot the resulting error percentages for different settings of the maximum allowed disparity error. For comparison, we choose six different stereo algorithms as representatives of several different matching strategies. We only consider methods that have already been published at the time of writing this paper. The disparity maps generated by these algorithms are obtained from the Middlebury Stereo

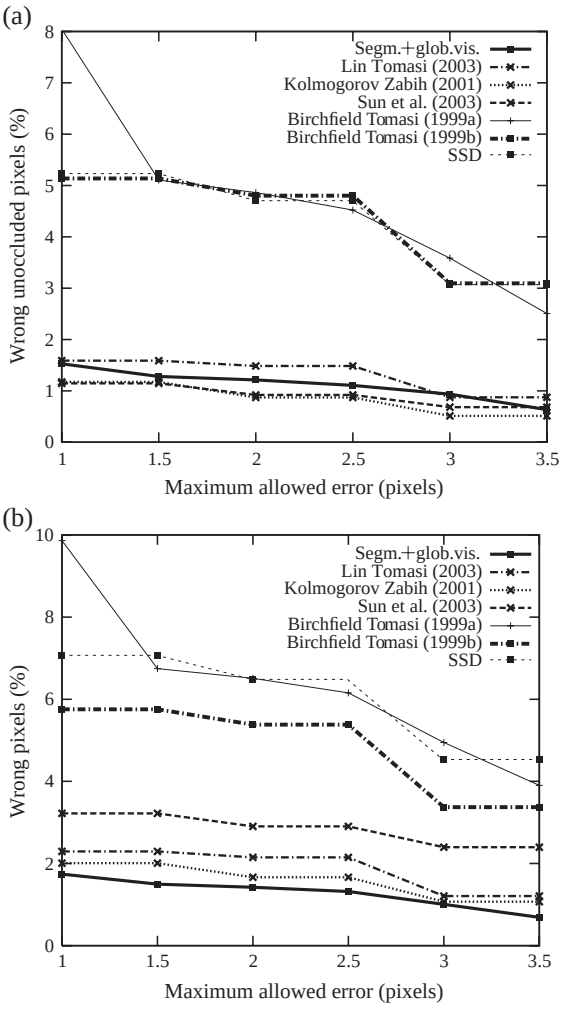


Fig. 13. Quantitative results for the Tsukuba test set. The percentage of wrong pixels for different disparity error thresholds is presented for two error metrics. (a) Only unoccluded pixels are considered. (b) All pixels are considered.

Vision website. Apart from the proposed algorithm, which is referred to as Segm.+glob.vis. in the figure, we present results from two layered methods (Birchfield and Tomasi, 1999a; Lin and Tomasi, 2003), the graph-based method of Kolmogorov and Zabih (2002), a belief propagation algorithm (Sun et al., 2003), an algorithm using dynamic programming (Birchfield and Tomasi, 1999b) and an implementation of sum-of-squared-differences (SSD) by Scharstein and Szeliski (2002) that uses shiftable windows. Concerning the first error metric, which is used in Fig. 13a, it is evident that our method is able to compete

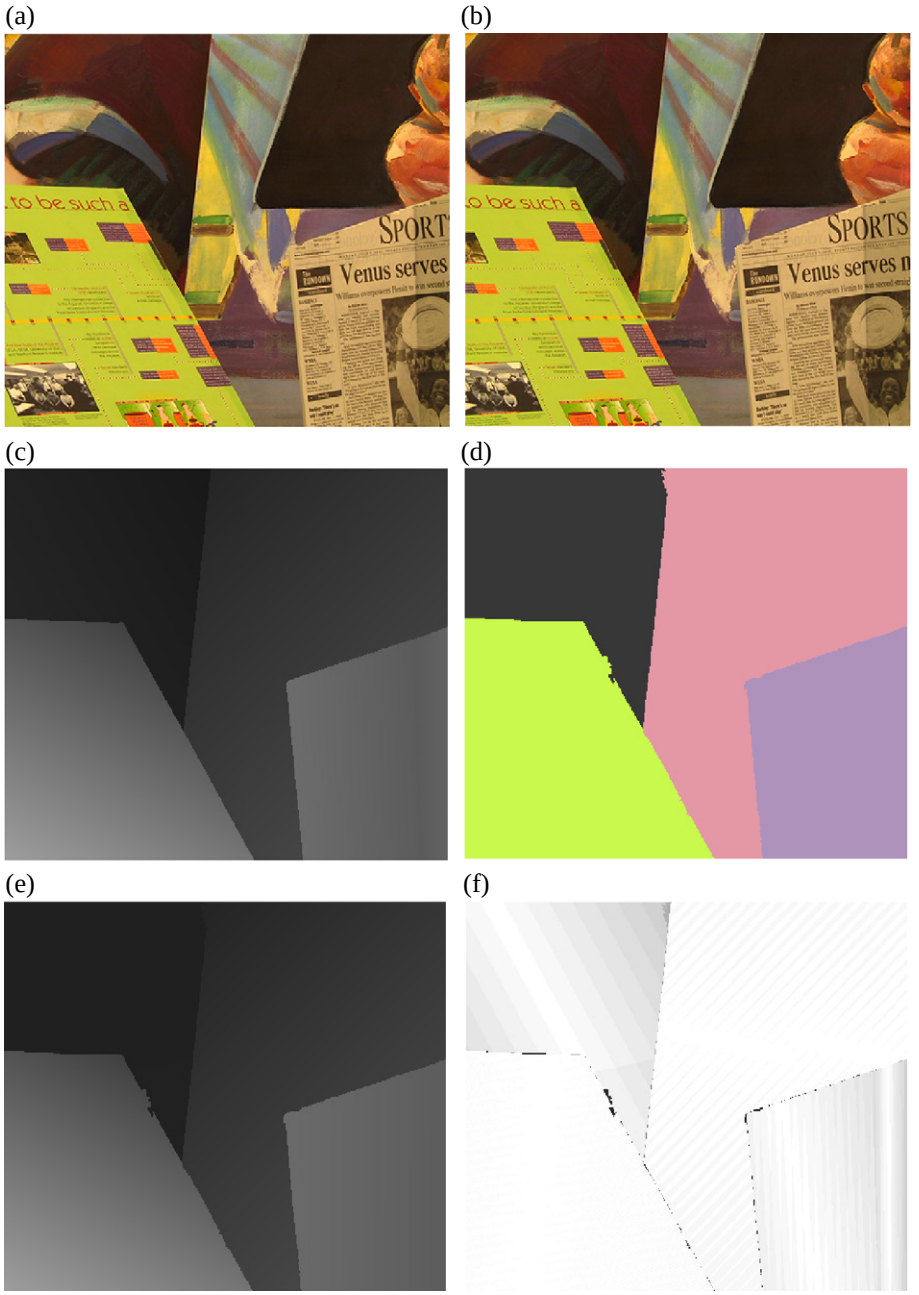


Fig. 14. Results for the Venus test set. (a) Left image. (b) Right image. (c) Ground truth provided with image pair in the geometry of the left image. The disparity maps are scaled by a factor of 8. (d) Computed layers. (e) Computed disparity map. (f) Absolute errors scaled by a factor of 32.

with the best performing algorithms with only the graph-based approach of Kolmogorov and Zabih (2002) and the belief propagation algorithm of Sun et al. (2003) giving better results. In Fig. 13b, we present the results using the second error metric that includes occluded pixels. For this error measurement,

our method outperforms the others, which proves its capability to generate meaningful results in occluded regions and precisely locate depth boundaries.

As a second test set we present the Venus image pair shown in Fig. 14a and b. The corresponding ground truth in the geometry of the left image is presented in Fig. 14c. The Venus data set consists only of planar surfaces. Although the scene structure is quite simple, there are large untextured regions that make the reconstruction difficult. Our algorithm extracts four layers as shown in Fig. 14d. The corresponding disparity map is then presented in Fig. 14e. The parameters were set to the following values: $r=0.6$, $\lambda_{occ}=15.0$ and $\lambda_{disc}=30.0$. Since the newspaper at the right of the image consists of two planes that are joined by a crease edge, the algorithm oversimplifies this surface. Nevertheless, the resulting disparity error shown in Fig. 14f is negligibly small. Our algorithm almost perfectly reconstructs the scene with the disparity planes correctly outlined. The quantitative results in Fig. 15 show that for this pair our method clearly outperforms the other algorithms for both error metrics.

We further evaluated the proposed algorithm on a more complex scene using the Teddy test set taken from Scharstein and Szeliski (2003). A large disparity range, more complex scene geometry and textureless areas make the image pair challenging for stereo algorithms. Scharstein and Szeliski are planning to add this test set to their benchmark, since they argue that current test images are getting too simple to discriminate among the best-performing stereo algorithms. The Teddy test set is shown in Fig. 16a and b. The scene consists of a large number of surfaces for which some can be well approximated as a plane (background, floor, roof and walls of the house), whereas others have a more complex surface structure (teddies, plants). The ground truth for the left image of the Teddy scene is presented in Fig. 16c. Pixels for which the method of Scharstein and Szeliski (2003) fails to produce the ground truth are coloured black. In the final configuration of the algorithm, the scene is represented by a set of 77 planes which we show in Fig. 16d. Surfaces that can be well approximated as a plane are thereby represented by a single or a small number of layers resulting in a robust reconstruction. However, for more complex shapes like the green teddy the

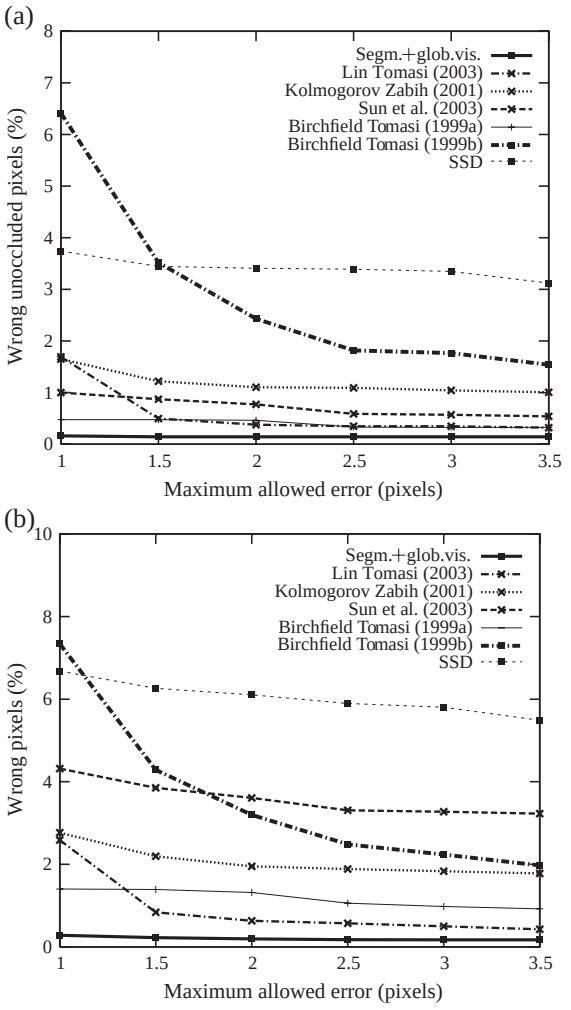


Fig. 15. Quantitative results for the Venus test set. The percentage of wrong pixels for different disparity error thresholds is presented for two error metrics. (a) Only unoccluded pixels are considered. (b) All pixels are considered.

surface is reconstructed by a larger number of layers providing a detailed description of the corresponding surface structure. The computed disparity map is then presented in Fig. 16e. We used the following parameter settings: $r=0.6$, $\lambda_{occ}=20.0$ and $\lambda_{disc}=2.5$. To illustrate the quality of the derived matching results we compare it against the ground truth in Fig. 16f. The percentage of pixels exceeding a disparity error of one including occluded regions is 6.55. If occluded regions are not considered, we get an error of 5.00%. 19.54% of pixels in occluded regions exceed the error threshold of one. Since we do not

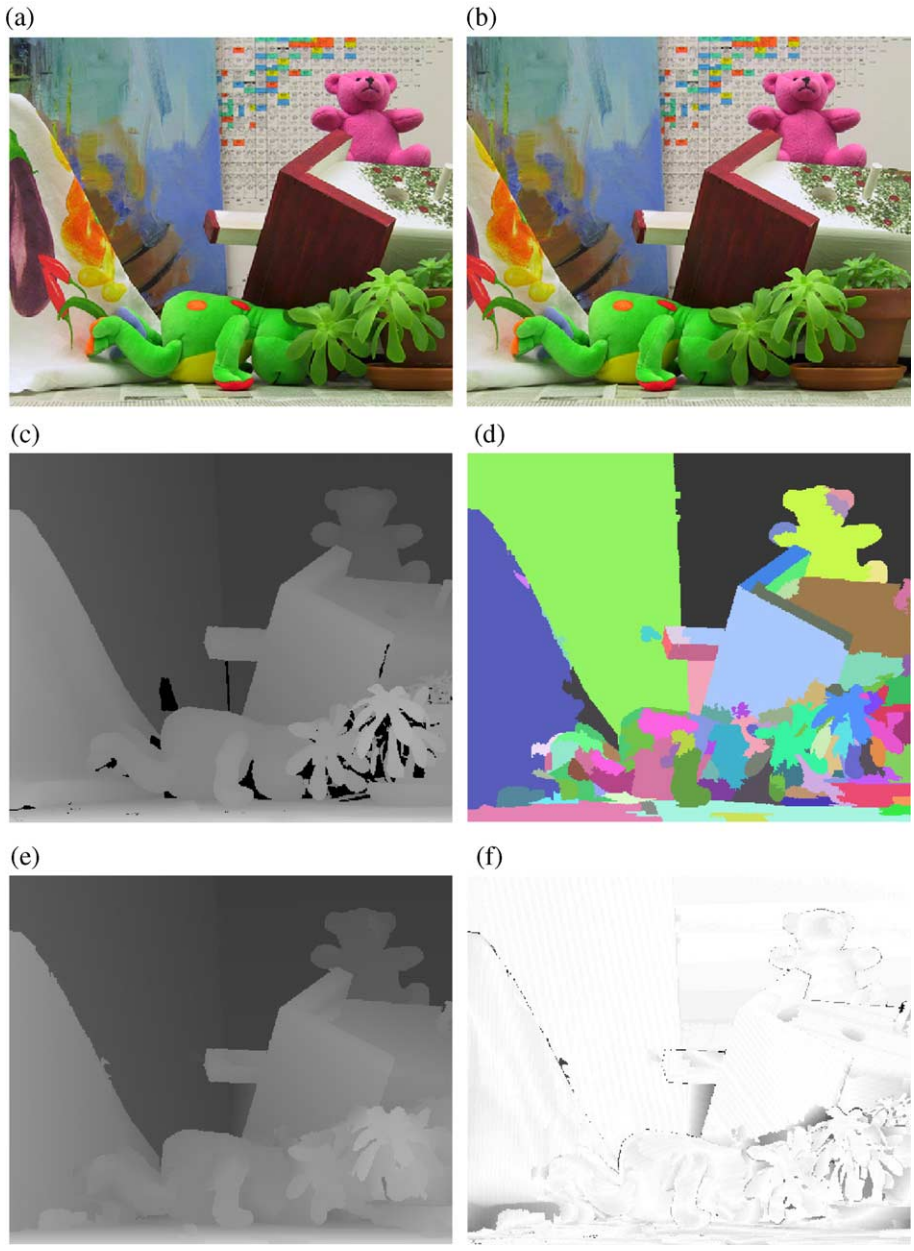


Fig. 16. Results for the Teddy test set. (a) Left image. (b) Right image. (c) Ground truth provided with image pair. The disparity maps are scaled by a factor of 4. (d) Computed layers. (e) Computed disparity map. (f) Absolute errors scaled by a factor of 16.

have the results for the other methods that we used for comparison for the previous two test sets, we present a reconstructed view of the scene in Fig. 17 to give a further impression of the accuracy and detail of the computed disparity information.

In addition, we computed an error statistic for the three discussed image pairs for which ground truth is available. As opposed to the errors shown in Figs. 13 and 15, the error values listed in Table 1 also include errors smaller than one pixel.

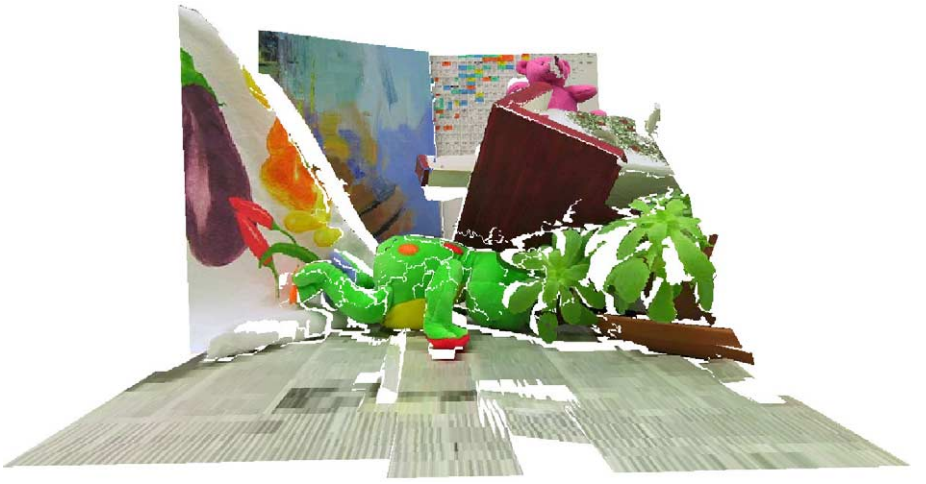

Fig. 17. Reconstructed view of the Teddy test set.

Finally, we applied our method to a stereo pair that we recorded using two Dragonfly IEEE-1394 colour cameras as provided by Point Grey Research. We calibrated the cameras using the method described in Zhang (2000) and transformed the images into epipolar geometry. The recorded stereo pair presented in Fig. 18a and b shows a person crouching in front of a wall. The scene contains untextured regions like sections of the white wall and the floor, and complex surface structure in the form of the person. We present the disparity map that was computed using the parameter settings $r=0.75$, $\lambda_{occ}=30.0$ and $\lambda_{disc}=10.0$ in Fig. 18d. Visual inspection of the disparity map indicates that the complex shape of the person's outline is correctly recovered. Furthermore, the algorithm seems to capture relatively well the person's disparity, which is better visible in the reconstructed view we present in Fig. 18e. The background is represented to a large extent by a single layer containing the left part of the white wall and most of the area covered by the wallpaper, which results in a robust reconstruction despite the poor texture of the white wall. In addition, regions belonging to the large occlusion left to the person's outline are correctly assigned to this background layer. A less accurate reconstruction is obtained for the floor, which we found was not only caused by its poor texture, but also by reflections of the wallpaper pattern on this surface.

Table 1
Error statistics computed from comparison against the ground truth

| Test set | Tsukuba | Venus | Teddy |
|---|---|---|---|
| Mean signed error (pixels) | −0.04 | 0.05 | 0.04 |
| Root mean-square error (pixels) | 0.73 | 0.31 | 1.07 |
| Maximum error (pixels) | 9.13 | 6.75 | 19.00 |

We implemented the proposed method in C++ and ran our algorithm on an Intel Pentium 4 2.0 GHz computer. For the $384\times288$ pixel Tsukuba and the $434\times383$ pixel Venus test set, the algorithm needed approximately 20 s until termination. For the $450\times375$ pixel Teddy and the $640\times480$ pixel self-recorded image pairs, the computational effort increased to 100 and 180 s, respectively. The longer running times are not only caused by the larger image sizes, but also by the more complex scene structures that require more layers to represent the scene. Therefore, the number of hypothesis tests that need to be performed is increased.

## 7. Conclusions

We have proposed a new stereo matching method that takes advantage of colour segmentation and uses planar layers to describe the scene. The algorithm is able to generate correct disparity information in untextured areas and regions close to depth boundaries, which is a challenging task in stereo matching. Our method alternates between a layer extraction and

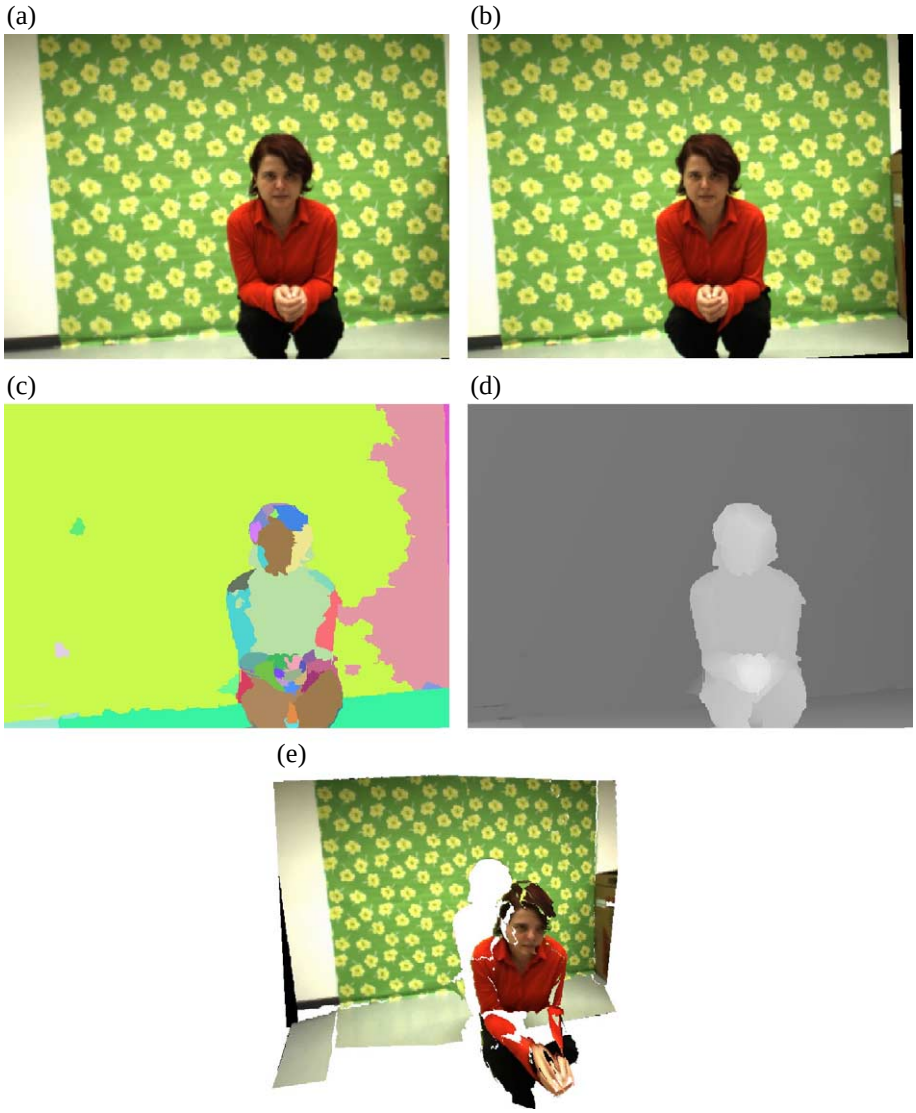


Fig. 18. Results for a self-recorded stereo pair. (a) Left image. (b) Right image. (c) Computed layers. (d) Computed disparity map in the geometry of the left image scaled by a factor of 4. (e) Reconstructed view.

an assignment step. Layers are extracted by a robust mean-shift-based clustering algorithm that takes advantage of a plane dissimilarity measurement that incorporates spatial information as well as plane parameters. The planar model of each layer is then computed based on the layer's spatial extent. The assignment of segments to layers is made in a hypothesis testing framework. Disparity information is thereby propagated across segments. Hypotheses are accepted if they improve a global cost function. For evaluating the costs of an assignment, the reference image is warped to the second view according to the disparity map. The cost function evaluates the pixel dissimilarity between the real and warped images and penalizes occlusions in both views and discontinuities between segments. Layer extraction and assignment are then iterated to find the generated disparity map with lowest costs.

We demonstrated the performance of the proposed algorithm using the test bed of Scharstein and Szeliski

(2002). Qualitative and quantitative evaluation proved the satisfactory quality of the achieved matching results. At the time of writing, the proposed method achieved second place out of 30 different stereo algorithms in the online evaluation on the Middlebury Stereo Vision website. We found that the proposed technique can provide occluded regions with more accurate disparity estimates than a set of Middlebury reference algorithms that we used for comparison. Furthermore, we applied our method to a more complex image pair taken from Scharstein and Szeliski (2003) and to self-recorded data. In the absence of reference data, we presented 3D visualizations of the reconstructed scene to demonstrate the good quality of the computed disparity layers.

One limitation of the presented approach lies in the assumption that the scene can be well approximated by a set of planes. This may not be the case, if the scene contains objects of more complex surface structures. Using a more sophisticated surface model remains a topic for further work. Furthermore, we plan to extend our layered approach from stereo images to stereo videos of moving scenes. We will explore possibilities to employ colour-segmented regions for both stereo matching and inter-frame motion tracking in order to develop techniques for video object segmentation based on combined depth and motion information.

## Acknowledgements

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

## Appendix A. Incremental image warping

From a computational point of view, warping the complete reference image according to the current layer assignment is a costly operation. Fortunately, this operation, which is called the base warp, only needs to be performed once for the initial solution. In the hypothesis testing phase usually only small parts of the warped image are changed. We therefore employ an incremental warping procedure that builds upon the base warp and only warps those segments to the second view that have a new assignment. In this process, we also incrementally calculate the costs of the formed solutions. For the implementation of the described hypothesis testing algorithm, we require two efficient basic operations. One operation serves to add a segment to the Z-buffer and the other is used to delete a segment from the Z-buffer. For each applied operation, we determine the resulting change of costs allowing an incremental computation of the current solution's costs.

To insert a segment into the Z-buffer, we apply the segment warping procedure described in Section 5.1. The coordinates in the second view and the colour values of the image points are retrieved and added to their corresponding Z-buffer cells. To calculate the change of costs in each individual Z-buffer cell occupied by the segment, we distinguish between three cases, as illustrated in Fig. 19. In the first case, a new entry is added to an empty cell. In the second case, the new entry is occluded by a pixel of the same cell having higher disparity, and in the third case, the new pixel occludes the pixel that was visible before insertion. Separating these three cases, the change of costs $\delta_{\text{add}}(p)$ implicated by adding pixel $p$ to a Z-buffer cell is computed by

$$\delta_{\text{add}}(p) = \begin{cases} \text{dis}(p) - \lambda_{\text{occ}} & : \text{ case 1} \\ \lambda_{\text{occ}} & : \text{ case 2} \\ \text{dis}(p) - \text{dis}(p_{\text{vis}}) + \lambda_{\text{occ}} & : \text{ case 3} \end{cases} \tag{A.1}$$

with $\text{dis}(p)$ being the colour dissimilarity of the pixel $p$ in the real and in the warped view, $\lambda_{\text{occ}}$ denoting the occlusion penalty and $p_{\text{vis}}$ being the pixel that was visible before the insertion of $p$. The change of costs $\Delta_{\text{add}}$ introduced by adding the segment to the Z-buffer is then computed by summing up the individual changes of costs over all cells occupied by the segment. Additionally, the discontinuity penalty $\lambda_{\text{disc}}$ is added for each pixel on the segment's border to a segment of a different layer assignment in the reference image. Since deleting a segment obviously represents the inverse operation, the change of costs $\Delta_{\text{del}}$ for deletion of a segment can be deduced analogously.

In hypothesis testing we first delete the current segment from the Z-buffer to test neighbouring layers' plane models. We record the resulting change of costs

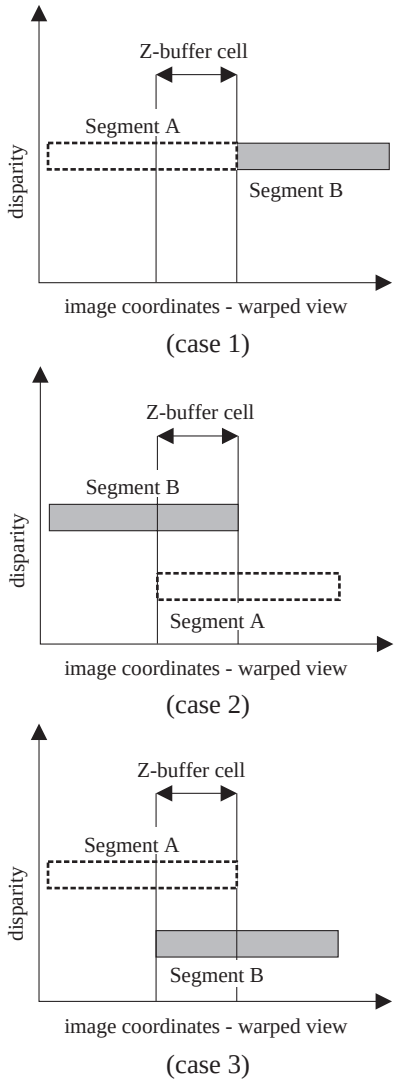


Fig. 19. Incremental computation of the costs in the Z-buffer. The insertion of Segment A implicates three different cases.

$\Delta_{del}$. We then replace the segment's planar model by the plane of a neighbouring layer and add it to the Z-buffer. The computed change of costs $\Delta_{add}$ is stored. We then use the delete function to remove the segment again. We test the hypotheses of all other neighbouring layers. Finally, we restore the Z-buffer to its original state by adding the segment using its old planar description. If there are neighbouring layers for which

$$\Delta_{del} + \Delta_{add} < 0 \qquad (A.2)$$

we found assignments for this segment that give locally lower costs than the current one. In this case, we record the assignment that gave the minimum value for this term. After all segments have been tested, we replace the old assignment by the recorded one. This update also needs to be applied to the Z-buffer using the described delete and insert functions. In each iteration of the algorithm, usually only a fraction of segments will be assigned to a new layer. Especially, when the algorithm converges to a local optimum, the number of updated segments will be very small. The use of the incremental delete and add functions for the update procedure therefore provides a significant gain of efficiency over the computational expensive operation of a base warp.

## Appendix B. Sensitivity of results to variations in parameter values

As for every global stereo matching method, the setting of parameters plays an important role. There are three parameters the user can tune to influence the algorithm's results: the mean-shift radius $r$, the occlusion penalty $\lambda_{occ}$ and the discontinuity penalty $\lambda_{disc}$. All other parameters and thresholds are set to constant values, which are given in the main text of the paper.

We take a closer look at the effects of varying the parameters using the Teddy test set shown in Fig. 16a and b. We have chosen the Teddy test set, since it has the most complex scene structure of the presented stereo pairs and thus presents the most challenging reconstruction task of the selected stereo pairs. The disparity map shown in Fig. 16e was generated using the following parameter values: $r=0.6$, $\lambda_{occ}=20.0$ and $\lambda_{disc}=2.5$. For studying the role of a specific parameter, we generate results by varying its setting. The two other parameters are thereby kept fixed and set to the values given above. Each result is then compared against the ground truth by computing the percentage of all pixels having a disparity error larger than one. The resulting plots are shown in Fig. 20 and are interpreted as follows.

The mean-shift radius $r$, whose plot is shown in Fig. 20a, controls the number of layers that are found in the layer extraction step of the algorithm. If $r$ is set to low values, the number of extracted clusters and

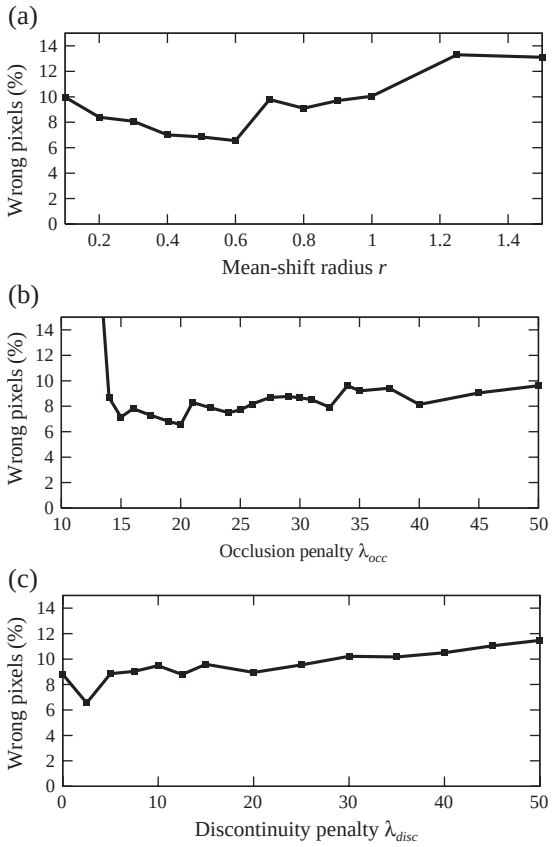


Fig. 20. Percentage of all wrong pixels for varying parameter values. (a) Different settings for the mean-shift radius $r$ ($\lambda_{occ}$=20.0, $\lambda_{disc}$=2.5). (b) Different settings for the occlusion penalty $\lambda_{occ}$ ($r$=0.6, $\lambda_{disc}$=2.5). (c) Different settings for the discontinuity penalty $\lambda_{disc}$ ($r$=0.6, $\lambda_{occ}$=20.0).

therefore layers will be high. In this case, the layers will not be very robust as a consequence of the small spatial extent over which their plane parameters were computed. On the other hand, setting the mean-shift radius to a high value causes two different surfaces to be represented by the same layer, which is not desirable either. For the Teddy test set, values in the range of [0.5, …, 0.6] represent a good trade-off between these two competing effects, as can be seen in the plot.

The plot for different settings of the occlusion penalty $\lambda_{occ}$ is shown in Fig. 20b. If $\lambda_{occ}$ is given a very low value, the algorithm tries to propagate planes that create occlusions in the warped view, which in general results in bad solutions. For $\lambda_{occ}$=10, we receive 38.1% of wrong pixels on the Teddy test set. On the other hand, overpenalizing occlusions usually decreases the performance in segments close to depth boundaries, since the algorithm then tries to generate continuous disparity transitions instead of modelling jumps in disparity that go along with occlusions. In this example, however, the results are not very sensitive to the occlusion penalty as long as it is not too low.

Finally, we show the plot for different settings of the discontinuity penalty $\lambda_{disc}$ in Fig. 20c. The plotted results show a minimum value at $\lambda_{disc}$=2.5 and relatively small variations over the rest of the displayed parameter range. A slight increase of the error rate with larger values of $\lambda_{disc}$ can be attributed to the large number of layers that is needed to accurately represent the scene. As a consequence, the boundary lengths between different layers are relatively large (e.g., the boundaries between the different plants in Fig. 16), which is penalized by $\lambda_{disc}$. Assigning large values to the discontinuity penalty $\lambda_{disc}$ therefore decreases the performance. Nevertheless, $\lambda_{disc}$ significantly contributes to the reconstruction of scenes consisting of large planar surfaces as the Venus test set shown in Fig. 14a and b.

## References

Birchfield, S., Tomasi, C., 1999a. Multiway cut for stereo and motion with slanted surfaces. International Conference on Computer Vision, 489–495.

Birchfield, S., Tomasi, C., 1999b. Depth discontinuities by pixel-to-pixel stereo. International Journal of Computer Vision 35 (3), 269–293.

Bobick, A., Intille, S., 1999. Large occlusion stereo. International Journal of Computer Vision 33 (3), 181–200.

Boykov, Y., Veksler, O., Zabih, R., 2001. Fast approximate energy minimization via graph cuts. Transactions on Pattern Analysis and Machine Intelligence 23 (11), 1222–1239.

Christoudias, C., Georgescu, B., Meer, P., 2002. Synergism in low-level vision. International Conference on Pattern Recognition (4), 150–155.

Comaniciu, D., Meer, P., 1999. Distribution free decomposition of multivariate data. Pattern Analysis and Applications 1 (2), 22–30.

Fua, P.V., 1991. Combining stereo and monocular information to compute dense depth maps that preserve depth discontinuities. International Joint Conference on Artificial Intelligence, 1292–1298.

Hirschmüller, H., Innocent, P., Garibaldi, J., 2002. Real-time correlation-based stereo vision with reduced border errors. International Journal of Computer Vision 47 (1/2/3), 229–246.

Kanade, T., Okutomi, M., 1994. A stereo matching algorithm with an adaptive window: theory and experiment. Transactions on Pattern Analysis and Machine Intelligence 16 (9), 920–932.

Ke, Q., Kanade, T., 2001. A subspace approach to layer extraction. Conference on Computer Vision and Pattern Recognition, 255–262.

Kolmogorov, V., Zabih, R., 2002. Computing visual correspondence with occlusions using graph cuts. International Conference on Computer Vision (2), 508–515.

Lin, M., Tomasi, C., 2003. Surfaces with occlusions from layered stereo. Conference on Computer Vision and Pattern Recognition, 710–717.

Mayer, H., 2003. Analysis of means to improve cooperative disparity estimation. International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences XXXIV, 25–31 (Part 3/W8).

Mühlmann, K., Maier, D., Hesser, J., Männer, R., 2002. Calculating dense disparity maps from color stereo images, an efficient implementation. International Journal of Computer Vision 47 (1), 79–88.

Scharstein, D., Szeliski, R., 2002. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. International Journal of Computer Vision 47 (1/2/3), 7–42 (http://www.middlebury.edu/stereo/ accessed 9 June 2004).

Scharstein, D., Szeliski, R., 2003. High-accuracy stereo depth maps using structured light. Conference on Computer Vision and Pattern Recognition (1), 195–202.

Sun, J., Zheng, N.N., Shum, H.Y., 2003. Stereo matching using belief propagation. Pattern Analysis and Machine Intelligence 25 (7), 787–800.

Tao, H., Sawhney, H., 2000. Global matching criterion and color segmentation based stereo. Workshop on the Application of Computer Vision, pp. 246–253.

Tao, H., Sawhney, H., Kumar, R., 2001. A global matching framework for stereo computation. International Conference on Computer Vision, 532–539.

Zhang, Z., 2000. A flexible new technique for camera calibration. Transactions on Pattern Analysis and Machine Intelligence 22 (11), 1330–1334.

Zhang, Y., Kambhamettu, C., 2002. Stereo matching with segmentation-based cooperation. European Conference on Computer Vision, 556–571.

Zitnick, C., Kanade, T., 2000. A cooperative algorithm for stereo matching and occlusion detection. Transactions on Pattern Analysis and Machine Intelligence 22 (7), 675–684.