Surface Stereo with Soft Segmentation

Michael Bleyer[1]*, Carsten Rother[2], Pushmeet Kohli[2]
[1]Vienna University of Technology, Vienna, Austria [2]Microsoft Research Cambridge, Cambridge, UK

Abstract

This paper proposes a new stereo model which encodes the simple assumption that the scene is composed of a few, smooth surfaces. A key feature of our model is the surface-based representation, where each pixel is assigned to a 3D surface (planes or B-splines). This representation enables several important contributions: Firstly, we formulate a higher-order prior which states that pixels of similar appearance are likely to belong to the same 3D surface. This enables to incorporate the very popular color segmentation constraint in a soft and principled way. Secondly, we use a global MDL prior to penalize the number of surfaces. Thirdly, we are able to incorporate, in a simple way, a prior which favors low curvature surfaces. Fourthly, we improve the asymmetric occlusion model by disallowing pixels of the same surface to occlude each other. Finally, we use the known fusion move approach which enables a powerful optimization of our model, despite the infinite number of possible labelings (surfaces).

1. Introduction

We believe that a good model for the world of all stereo images is the following: A 3D scene is a collection of a few, smooth 3D surfaces, such that a 3D point should look similar in both views (photo-consistency assumption) and that 3D points in one image, which have similar appearance, are likely to lie on the same 3D surface. In this paper, we formalize this idea in the form of an energy and optimize this energy in a powerful and efficient way.

While several of these aspects are utilized by virtually all existing stereo methods, we believe that especially one aspects has not been expressed in any existing approach in a principled way. This is the aspect of self-similar pixels belonging to the same 3D surface. One way of defining self-similarity is by performing a color segmentation on the image. Note that the concept of self-similarity is widely used in other areas of computer vision, such as image segmentation or super-resolution. In stereo, segmentation-based methods (*e.g*. [2, 6, 11, 13, 25, 29]) have become extremely popular in the last years, mostly due to their high quality results. For example, they clearly dominate the top rankings of the Middlebury benchmark [21]. The core of these algorithms is formed by the *segmentation assumption*, which states that all pixels within a segment should lie on the same 3D surface. Most segmentation-based algorithms [2, 11, 13, 25, 29] do the following two step procedure. In the first step, each segment is assigned to one unique 3D disparity plane by applying plane fitting to an initial disparity map. In the second optimization step, the extracted disparity planes are then propagated among segments to minimize a given energy function. The key drawback of all these approaches is that the segmentation assumption is a hard constraint, *i.e*. two pixel which share the same segment *must* be assigned to the same plane.

Our Approach We propose a pixel-wise MRF formulation that assigns each pixel to a 3D surface. We overcome two major problems of segmentation-based stereo: (1) We show that it is possible to express the segmentation (or self-similarity) assumption in a principled way as a *soft constraint*. Hence, in contrast to most segmentation-based methods, we can successfully recover from segmentation errors. (2) We go beyond the planar world assumption and make use of far more general B-spine surfaces.

Furthermore, we believe that a key feature of our model is the surface-based representation where each pixel is assigned to a 3D surface (planes or B-splines). While this representation has been used before [1, 17], we believe that these works have not fully capitalized on the potential of this representation. In broad terms we make the following contributions, which are outlined in detail in sec. 2:

(1) As mentioned above, we propose a higher-order prior that encodes the assumption that self-similar pixels lie on the same surface as a soft constraint. (2) We present a prior that minimizes surface curvature, which is measured in an analytical way. (3) A minimum description length (MDL) prior imposes a penalty on the number of surfaces. (4) We improve the asymmetric occlusion model in order to treat slanted surfaces correctly.

A key question is how to optimize this model given the fact that there is an infinite number of possible labelings (*i.e*. surfaces). Here, we use the recently proposed fusion move approach [16], which is efficient and global in the sense that it optimizes over all unknown variables jointly.

Surface versus Disparity Representation In order to implement our model assumptions, we believe that a

*Michael Bleyer received financial support from the Austrian Science Fund (FWF) under project P19797 and the Vienna Science and Technology Fund (WWTF) under project ICT08-019.

surface-based representation is vital and that a disparity-based representation is considerably inferior. One obvious advantage is that the result of a surface representation is a continuous 3D surface model, free of any discretization artefacts. Probably the most severe limitation of the disparity representation is that our MDL and soft segmentation priors are virtually impossible to be realized. This is due to the fact that any arbitrary disparity labeling which forms a surface has to be modeled individually. Moreover, measuring curvature in a disparity representation requires a triple clique [28], which is difficult to optimize. As we will see, the complex triple clique simplifies to a unary term in the surface representation.[1] Finally, slanted surfaces are difficult to be handled correctly with an asymmetric occlusion model in a disparity representation.

2. Related Work and Contributions

In the following, we go into more detail on the individual contributions and discuss them in the context of prior work.

Soft Segmentation Our first contribution is a new prior that is directly derived from the segmentation assumption. It states that all pixels belonging to the same segment are more likely belonging to the same 3D surface, but importantly are *not forced* to lie on the same surface. We call this a "soft segmentation" prior. In practice, we use many overlapping segmentations. We are able to enforce this higher-order constraint efficiently during the optimization by exploiting recent progress in optimization with sparse higher-order cliques via graph-cut-based quadratic pseudoboolean optimization (QPBO) [14, 19].

Despite the importance of the problem, there is only a modest amount of work dealing with the problem of relaxing the segmentation assumption. There are two common strategies to deal with this problem.

The first obvious solution is to apply an extreme over-segmentation, which is driven by the hope that as few segments as possible straddle a true surface boundary, see *e.g*. [2, 11, 13, 25, 29]. Unfortunately, this is not realistic on a large variety of practical stereo images. Further, Deng *et al*. [6] intersect segments of the left view with segments of the right image. The authors claim that this procedure reduced risk of bad segments. Some recent methods [4, 24] allow for slight modifications of segment shapes, but cannot handle large segmentation errors.

The second line of research is more similar to ours in the sense that they incorporate image over-segmentation into a pixel-wise MRF model. In the simplest form [28], this is achieved by adjusting the pairwise smoothness term. The costs for assigning two neighboring pixels to different disparities are thereby set to high values if the pixels belong to the same segment (and to low values, otherwise). However, similar to edge-sensitive smoothness weights, these "segment-sensitive" smoothness weights are affected by the well-known "shrinking-bias" in the MRF and can therefore not preserve the underlying segmentation well.

Sun *et al*. [23] incorporate segmentation by using a precomputed disparity map that is derived by fitting 3D planes to a given color segmentation. In their energy formulation, a disparity solution is penalized if it shows deviations from the precomputed plane. This is a pixel-wise soft constraint, as in our work. However, the *key difference* is that they commit to a single precomputed plane, while we allow any arbitrary surface in a segment. Hence, in [23], even for a "perfect" segmentation, the plane fitting results might still be wrong, which lets the model favor an incorrect solution.

Recently, Smith *et al*. [22] improved the idea of edge-sensitive smoothness weights by extending the range of pairwise terms considerably (within a large window). This leads to a "nonparametric smoothness prior". They demonstrate that disparity boundaries are persevered, without performing an explicit over-segmentation.[2] However, the *key difference* of approaches [22, 23, 28] to ours is that we explicitly encode the idea of self-similarity within a segment.[3]

Going Beyond Simplistic Planar Models Let us now address the second disadvantage of segmentation-based approaches, *i.e*. that 3D planes are oftentimes not general enough to represent real 3D surface shapes. Hence, we go beyond the planar model and make use of B-spline surfaces.

In this context, the work most similar to ours is [17], which has used splines in a stereo matching algorithm. The obvious difference to our work is that the authors do not use a soft segmentation term and an MDL prior. Similar to our work, they introduce a prior that regularizes surface shapes. While [17] uses a measure of the first order derivative, we enforce surfaces of low curvature (second order derivative), which is more consistent with recent work [28]. Here, our main point is that in comparison to [28], it is considerably easier to optimize this measure due to our surface-based representation (as discussed above).

MDL Prior A key strength of our surface-based model is to encode the assumption that a simple explanation of the scene (consisting of a small number of surfaces) is in general better than an unnecessarily complex one (consisting of a large number of surfaces). This global MDL-type prior has been proposed in [10] for object class recognition, but

[1] It is worth to note that [28] could have also used a surface-representation instead of a disparity-based one, since in the optimization a fusion move approach with continuous 3D surfaces is utilized. In essence, this would have given [28] a simplified form of our model, *i.e*. without soft segmentation or MDL priors.

[2] The same observation has actually been done in the work [7] where it is shown that boundaries are better persevered when a larger neighborhood is used. Hence [22] can be seen as a generalization of the standard 4-connected edge-sensitive pairwise terms.

[3] Note that the highly connected *pairwise* MRF of [22] cannot be transformed into the sparse *higher-order* MRF we use to model the soft segmentation term.

is new to stereo matching. The MDL prior serves the purpose of propagating surfaces over large distances. Consider an image where the background is divided into many unconnected regions by a foreground object, *e.g*. a wall seen through a fence such as in the Cones image of the Middlebury set (see figure 3a). Obviously, the correct solution is to use one 3D plane for the complete wall instead of a collection of slightly different planes for each background region. This is exactly what our MDL prior enforces: A penalty on the number of surfaces. Since the background regions are typically not connected and potentially far apart in the image, this cannot be accomplished by a standard pairwise smoothness term (between neighboring or close-by pixels).

Improving the Asymmetric Occlusion Model Our fourth contribution addresses occlusion handling. We avoid a symmetric occlusion strategy for the purpose of computational efficiency and apply the asymmetric occlusion model [26, 27, 28]. Our contribution is to extend this asymmetric model in order to handle slanted surfaces correctly. In its standard form, the asymmetric model cannot handle situations in which two pixels of the same surface in the left image match a single pixel of the right view. (This case arises due to the different sampling of the same surface in the two images [18, 23].) In this case, one of the pixels in the left image is erroneously declared as being occluded. Since our approach explicitly models surfaces, it is easy to prohibit pixels from the same surface to occlude each other.

Fusion Moves Finally, we make a simple but conceptually important contribution in the optimization procedure. While the fusion move algorithm [16] has been used to optimize the assignment of pixels to continuous disparity values (*e.g*. [28, 15]), we are, to our knowledge, the first to use it for optimizing surface assignments. Optimizing surface assignments is primarily difficult due to the very large number of candidate surfaces (*e.g*. after the initial plane fitting). Standard optimizers such as Belief Propagation or α-expansions can therefore easily become intractable. Consequently, previous work on surface-based stereo uses layer extraction to reduce the number of surfaces [2, 11, 17], only uses fronto-parallel surfaces [24, 29] or uses suboptimal optimization techniques such as ICM [2, 25]. Thanks to the fusion move algorithm, we can handle this large label set, while still taking advantage of powerful graph-cut-based optimization. Another important aspect of the fusion move approach is the ability to refine existing surfaces during the optimization (see also [28, 15]). This *data-driven* strategy allows the optimization to explore virtually any 3D surface, *i.e*. any 3D surface which is possible to be represented by a set of B-splines. Refitting has also been used in [17].

3. Model

Notation Let $\mathcal{I}$ denote the pixels of the reference image and $\mathcal{F}$ be the set of all 3D surfaces. Our goal is to find a mapping $F : \mathcal{I} \rightarrow \mathcal{F}$ that assigns each pixel $p \in \mathcal{I}$ to a surface $f_p \in \mathcal{F}$. Note that assigning a pixel to a surface, implicitly defines the pixel's disparity. We write $d(p, f_p)$ to refer to pixel p's disparity according to surface f_p. How to evaluate $d(p, f_p)$ depends on f_p's surface type. In our implementation, we use two types of surfaces. (1) If f_p corresponds to a plane, $d(p, f_p) = f_p[a] \cdot p_x + f_p[b] \cdot p_y + f_p[c]$ where a, b and c are the plane's parameters and p_x, p_y denote p's image coordinates. (2) In case that f_p corresponds to a B-spline surface[4], the disparity is evaluated using the spline's blending functions. We evaluate the quality of a mapping F via an energy function $E(F)$ defined as

$$E(F) = E_{data}(F) + E_{smooth}(F) + E_{seg}(F) + E_{curv}(F) + E_{mdl}(F). \quad (1)$$

Data Term Our data term determines the pixel dissimilarity for all visible pixels, while it assigns a fixed penalty for occluded ones. Formally, we define the data term as

$$E_{data}(F) = \sum_{p \in \mathcal{I}} [\rho\left(p, p - d(p, f_p)\right)\left(1 - O(p)\right) + \lambda_{occ} O(p)]. \quad (2)$$

Here, λ_{occ} is a constant occlusion penalty. This penalty prevents the algorithm from maximizing the number of occluded pixels. $O(p)$ is a function that determines the occlusion state of pixel p and is discussed later. $\rho()$ denotes the robust dissimilarity function taken from [28] and defined as

$$\rho(p, q) = -\log\left(1 + \exp(-m(p, q)^2/\sigma_d)\right) \quad (3)$$

where σ_d is a parameter and $m()$ computes the color difference. We employ mutual information [9] to implement $m()$. This has the advantage of being insensitive to radiometric distortions. (For example, the left image is darker than the right one.)[5]

Let us now discuss the function $O()$. Our occlusion definition extends the asymmetric occlusion model used in [26, 27, 28]. These papers employ the *visibility constraint*, which states: If two pixels of the left image project to the same pixel of the right view, then the one of lower disparity is occluded. However, in [18, 23], it has been realized that this constraint is violated for slanted surfaces. If two or more pixels lie on the same surface, they can all have the same matching point in the right image without any of the pixels being occluded. This is due to the different sampling of the surface in the two images. Since our stereo model explicitly uses surfaces, it easy to incorporate an additional

[4] In our implementation, we use open uniform B-spline surfaces [8].

[5] Mutual information requires an initial disparity map, which does not need to be of high quality. We employ the fast stereo matcher of [3] to obtain the initial disparity map.

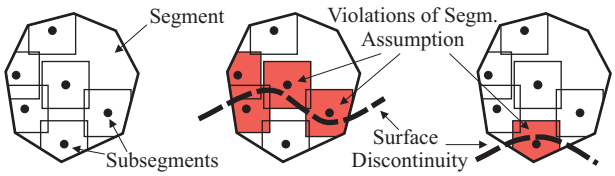


Figure 1. Soft segmentation term. (a) We construct a subsegment at each pixel by intersecting a square window with the segment. (For legibility, only some of the overlapping subsegments are shown.) A penalty is imposed if there is more than one surface within a subsegment (segmentation assumption). (b) A surface discontinuity intersects the segment, which leads to the presence of two different surfaces in the segment. The segmentation assumption is violated for a large number of subsegments (colored red). Each of them imposes a penalty. (c) Smaller deviation from the segment's shape is penalized less, because the segmentation assumption is only violated for a small number of subsegments.

condition into the visibility constraint by

$$O(p) = \begin{cases} 1 & \text{if } \begin{array}{r} \exists q \in \mathcal{I} : p - d(p, f_p) = q - d(q, f_q) \\ \wedge d(p, f_p) < d(q, f_q) \wedge f_p \neq f_q \end{array} \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

Note that in our definition a pixel q can occlude a pixel p only if the pixels lie on different surfaces, *i.e.* $f_p \neq f_q$.

Smoothness Term Our smoothness term motivates spatially neighboring pixels to lie on the same 3D surface. It is defined as

$$E_{smooth}(F) = \sum_{<p,q> \in \mathcal{N}} \lambda_{smooth} T[f_p \neq f_q] \tag{5}$$

where $\mathcal{N}$ represents the set of all spatial neighbors in 8-connectivity. The constant λ_{smooth} denotes a smoothness penalty. T is the indicator function that returns 1 if its argument is true and 0, otherwise.

One could make the smoothness penalty depended on image edges or segment borders. However, we have decided against that and solely rely on our soft segmentation term to incorporate monocular information into our model.[6]

Soft Segmentation Term Our soft segmentation term is given an arbitrary segmentation that partitions the left image into disjoint regions as an input. This segmentation can, for example, be derived from any color or texture segmentation algorithm.[7] In our implementation, we employ the commonly-used mean-shift segmentation algorithm of [5].

[6]One could potentially also introduce different penalties for different types of surface transitions, such as a penalty on (1) the absolute difference in disparity, (2) the difference in tangent planes or (3) the difference in curvature. However, we did not implement this to keep the number of parameters low. It is worth to note that our energy term (5) implicitly approximates (1-3) to some extend. For instance, the energy incurred by a smooth transition between two surfaces is likely higher than the energy where the two surfaces are replaced by a single B-spline surface.

[7]Our approach can also handle multiple segmentations.

Our term constructs a subsegment at each pixel p. To generate this subsegment, we center a square window on p, which provides the set $\mathcal{W}_p$:

$$\begin{aligned} \mathcal{W}_p = \{q \in \mathcal{I} : & p_x - \frac{w}{2} \leq q_x \leq p_x + \frac{w}{2} \\ & \wedge p_y - \frac{w}{2} \leq q_y \leq p_y + \frac{w}{2}\} \end{aligned} \tag{6}$$

where w is a parameter defining the window size. Further, let $\mathcal{S}_p$ denote the segment in which p resides. We then derive p's subsegment $\mathcal{L}_p$ by intersection: $\mathcal{L}_p = \mathcal{W}_p \cap \mathcal{S}_p$. Since we build a subsegment at each pixel, this leads to a large number of overlapping subsegments (figure 1a).

Our soft segmentation term encourages the segmentation assumption to be fulfilled in each subsegment. To implement this assumption, we give 0 costs if there is only a single surface within a subsegment and a user-defined penalty λ_{seg}, otherwise. Formally, we define the term as

$$E_{seg}(F) = \sum_{p \in \mathcal{I}} \begin{cases} 0 & \forall q \in \mathcal{L}_p : f_q = f_p \\ \lambda_{seg} & \text{otherwise.} \end{cases} \tag{7}$$

Note that setting $\lambda_{seg} := \infty$ leads to a strict enforcement of the segmentation constraint, as implemented by most other segmentation-based methods. For different settings, the term only encourages surface discontinuities to be aligned with segment borders and hence allows deviations. In this context, it is important to understand that the amount of penalization is proportional to the amount of deviation from the segment shapes. This is enabled by our construction of subsegments and explained in figures 1b and 1c.

Curvature Term This prior regularizes the shape of B-spline surfaces by enforcing low curvature. We define the curvature prior as

$$E_{curv}(F) = \sum_{p \in \mathcal{I}} \lambda_{curv} \tau(p, f_p) \tag{8}$$

where λ_{curv} is a constant penalty and the function $\tau(p, f_p)$ denotes the second order derivative (curvature) of surface f_p computed at pixel p. Note that since we have a continuous surface model, we can analytically compute the *exact* second order derivative (or any other point-wise surface property). Also note that our prior is implemented as a simple unary term in the graph. All of this is in contrast to [28] where the second order derivative is only an *approximation* and hence affected by discretization artefacts and introduces non-submodular triple cliques in the graph.

MDL Term The term imposes a penalty on the occurrence of a surface. It therefore aims at minimizing the number of surfaces, which implements the MDL principle. Formally, the prior is defined as

$$E_{mdl}(F) = \sum_{f \in \mathcal{F}} \lambda_{mdl} T[f \in F] \tag{9}$$

where λ_{mdl} is a constant penalty.

4. Optimization

To optimize our energy, eq. (1), we use the fusion-move approach [16] which has recently become quite popular, *e.g.* [28, 12], probably due to its ability of handling arbitrary continuous labelings in an efficient discrete optimization framework. Fusion-move is an iterative procedure where a current surface assignment F is "fused" with a new *proposal* surface assignment F'. The fusion is done by optimizing an auxiliary binary energy, where label 0 (at pixel p) means that surface label f_p will be taken and label 1 means surface label f'_p. In our case, the binary fusion-energy is a pairwise energy, which is, in general, non-submodular. Hence we optimize this NP-hard problem with the QPBO-I method [20]. In order to apply the fusion-move approach, there are two open questions: (1) how to transform our energy, eq. (1), to a binary pairwise fusion-energy and (2) how to generate proposals F'. These are discussed below. In future work, we plan to also add the gradient descent fusion-move strategy of [12] which may improve performance.

4.1. Constructing the Fusion-Energy

In the following, we go through individual terms of our energy function, eq. (1), and express those in the binary, pairwise fusion-energy.

The construction of the pairwise energy of the data term, eq. (2), follows [27, 28]. Roughly spoken, it works by adding an infinite pairwise term to prevent a pixel from being visible if there is another pixel that occludes it. The only difference to [27, 28] is that we do not add such pairwise terms if these pixels lie on the same surface. The smoothness constraint, eq. (5), is a pairwise term, and the curvature prior, eq. (8), a unary term. The construction of the MDL prior, eq. (9), in the form of a pair-wise energy can be found in [10] for the α-expansion move approach. The generalization to fusion moves is straight forward.

The most interesting part is the soft segmentation term, eq. (7), which is a higher-order potential of large clique size. While, in general, such potentials are very difficult to handle, we can take advantage of the sparseness property of the soft segmentation term. In particular, [14] and [19] propose transformation schemes with which our higher-order potential can be transformed to a pairwise energy using a small number of additional auxiliary nodes. (In practice, a maximum of 4 auxiliary nodes per higher-order potential is required.) In detail, consider one higher-order clique which has 4 nodes and two proposals $\bar{F}$ and $\bar{F}'$ which have to be fused, *i.e.* binary label 0 corresponds to $\bar{F}$ and 1 to $\bar{F}'$. We distinguish between two cases. Firstly, if all surface labels in either $\bar{F}$ and/or $\bar{F}'$ are identical, then the binary fusion energy has non-maximum costs for binary labelings $\{0,0,0,0\}$ and/or $\{1,1,1,1\}$. This is a $\mathcal{P}^n$ Potts model and hence the construction of [14] can be used, which leads to a submodular pairwise energy. Secondly, assume the case where *e.g.* $\bar{F} = \{5,5,5,3\}$ and $\bar{F}' = \{6,6,5,5\}$. Here the binary labeling $\{0,0,1,1\}$ has non-maximum costs.[8] Fortunately, [19] showed a transformation[9] which leads to a non-submodular pairwise energy.

In the context of non-submodularity, all of our pairwise terms, except for the curvature prior, may be non-submodular. Despite this drawback, we found experimentally that the number of unlabelled nodes of QPBO is relatively low (approximately 4% on average).

4.2. Proposal Generation

Here, we present different types of proposals, which is inspired by previous work [15, 16, 28]. In the first step, we fuse all proposals of the form *Segmentation + Model Fitting* and *Fronto-Parallel*. In practice, we already obtain a good-quality solution at this stage and try to refine it in the following. (Note that all of the subsequent proposal types take the current solution as an input and modify it in some way.) We loop through the proposal sequence: (1) *Single Surface*, (2) *Refit*, (3) *K-Means*. In our implementation, we use three iterations after which the algorithm terminates. Additionally, we propose a *Surface Dilation* proposal at random points in time.[10]

Segmentation + Model Fitting Proposals This proposal type is almost identical to the *SegPl* proposals of [28]. We apply mean-shift color segmentation [5] on the reference image and compute a disparity map using a fast but imprecise stereo matcher [3]. A first proposal is generated by fitting a plane to each segment using the fitting procedure described in [2]. We then compute a second proposal by fitting a B-spline to each segment. To obtain a large number of different proposals and to consequently reduce the risk of missing correct surfaces, we compute various segmentations and disparity maps by changing the parameters of the segmentation and stereo algorithms. In our implementation, we compute 8 different mean-shift segmentations and 5 disparity maps (by varying the smoothness setting of [3]). By combining each segmentation with each disparity map, we derive 40 plane and 40 B-spline proposals.

Fronto-Parallel Proposals A proposal of this type only contains a single surface, namely a fronto-parallel plane. We generate one proposal for each allowed disparity. These proposals allow us to fall-back to a fronto-parallel plane in case that a correct slanted plane or B-spline has been missed in the *Segmentation + Model Fitting* proposals (However, this has rarely been the case in our experiments).

[8] It is important to note that in these cases many binary labelings, *e.g.* $\{0,0,0,1\}$ can be assigned a maximum cost, *i.e.* λ_{seg}, without changing the cost of the original energy, since the surface label of the third pixel is 5 in both surface labelings $\bar{F}$ and $\bar{F}'$.

[9] We use the Type-I construction.

[10] Approximately every fifth proposal is of this type.

Single Surface Proposals This proposal type aims at refining the current assignment of pixels to surfaces. We start from the current solution and select large surfaces, *i.e.* which cover at least 5% of the pixels in the reference image. For each large surface f, we generate one proposal in which all pixels are assigned to f. Note, we do not expand small surfaces, since their number is typically large. This can severely influence the algorithm's run time. *Single Surface* proposals are vital for optimizing our MDL prior.

Refit Proposals This proposal type aims at refining the current surfaces. For each surface f that is part of the current solution, we extract all pixels assigned to f. We then refit the surface over these pixels, either using a plane or a B-spline. In the fitting procedure, we apply all of the 5 precomputed disparity maps (the ones obtained using [3] as discussed above) and the current disparity map. In total, this leads to 6 different plane and 6 B-spline proposals.

K-Means Proposals Here, we first generate the disparity map according to the current surface assignments. The disparity map is then segmented using the K-Means algorithm where the number of cluster centers is chosen randomly. For each segment of homogeneous disparity, we either fit a plane or a B-spline using one of the 5 precomputed disparity maps or the current disparity map. The decision whether to fit a plane or B-spline and which disparity map to use are random. Here is an example to explain the intuition behind a *K-Means* proposal. A cone in the Cones set (Middlebury set) can be represented by several slightly different planes that all lead to very similar disparities in the corresponding disparity map. Obviously, a better choice is to use a single B-spline surface to model the whole cone. This is exactly what a *K-Means* proposal will construct: It segments the whole cone based on its disparity and fits a single surface.

Surface Dilation Proposals Proposals of this type are specifically designed for minimizing the value of the soft segmentation term. To obtain a proposal, we take the current solution and "expand" a randomly chosen number of surfaces. Expansion of a surface f works as follows. We extract the region consisting of all pixels assigned to f and perform morphological dilation on it. In the proposal, all pixels of the dilated region are then assigned to f.

5. Results

We use the Middlebury benchmark [21] to evaluate our method. The algorithm's parameters are set to the constant values of $\lambda_{occ} := 0.001$, $\sigma_d := 2.5$, $\lambda_{smooth} := 0.03$, $\lambda_{seg} := 0.1$, $w = 5$, $\lambda_{curv} := 0.001$ and $\lambda_{mdl} := 20$. These parameters have been found empirically to optimize performance. The results on the Middlebury set (figure 2) show that our method is successful in: (1) correctly capturing the disparity discontinuities (due to our soft segmentation term and occlusion handling), (2) modeling complex surfaces as low curvature B-splines (due to our curvature prior) and (3) keeping the number of surfaces low (due to our MDL prior). Our current Middlebury online ranking is the *sixth* rank out of 74 methods. For the Teddy test set, our method even achieves the *first* rank on all error measures.

We now evaluate the performance of the individual terms of our energy in eq. (1). We have therefore tested our method with different parameters using the Middlebury online table and plot corresponding results in table 1. In the table, the entry *All Terms On* provides quantitative results using the parameters above. This entry serves as a reference, *i.e.* when we say that performance changes this should be understood in comparison to *All Terms On*. In the following, we will always modify exactly one parameter and all other parameters remain constant, as defined above. We go through the table from bottom to top.

As a first experiment, we turn off our soft segmentation term by setting $\lambda_{seg} := 0$. Note that (as depicted from the entry *Soft Seg. Off*) this leads to a large drop in performance by 29 ranks. This is not surprising, because segmentation is known to give a big advantage on the Middlebury set. We now turn off our MDL prior by setting $\lambda_{mdl} := 0$. The result is that the number of surfaces grows considerably (figure 3), which has negative influence on the quantitative results (entry *MDL Off*). Also setting $\lambda_{curv} := 0$ (to switch off the curvature prior) leads to performance degradations (entry *Curvature Off*). The resulting overfitting problem is demonstrated for the Venus set in figure 4. We evaluate the performance of our improved occlusion handling strategy by implementing the standard asymmetric occlusion model, *i.e.* we allow pixels of the same surface to occlude each other. This leads to very little difference in performance (*Standard Asym. Occ.*). However, our occlusion handling is more correct with respect to slanted surfaces (see figure 5).

To demonstrate the advantage of our soft segmentation term over hard segmentation methods, we switch on all parameters and set them as defined above. We use the Map test pair, which has traditionally been a pitfall for segmentation-based algorithms in the old Middlebury benchmark, but has unfortunately been removed in the new one. Figure 6 shows that due to our soft segmentation constraint, we can handle this image set well.

6. Conclusions

This paper has proposed a high-quality stereo algorithm. Our major contributions are a soft segmentation term and an MDL prior. We have argued that a surface-based representation has considerable advantages over a disparity-based one. A limitation is that our algorithm does currently not handle the input images in a symmetric way. We have decided for the asymmetric treatment for computational reasons, *i.e.* it takes approximately an hour to compute a disparity map and a symmetric model would most likely increase the computation time by a factor of two.

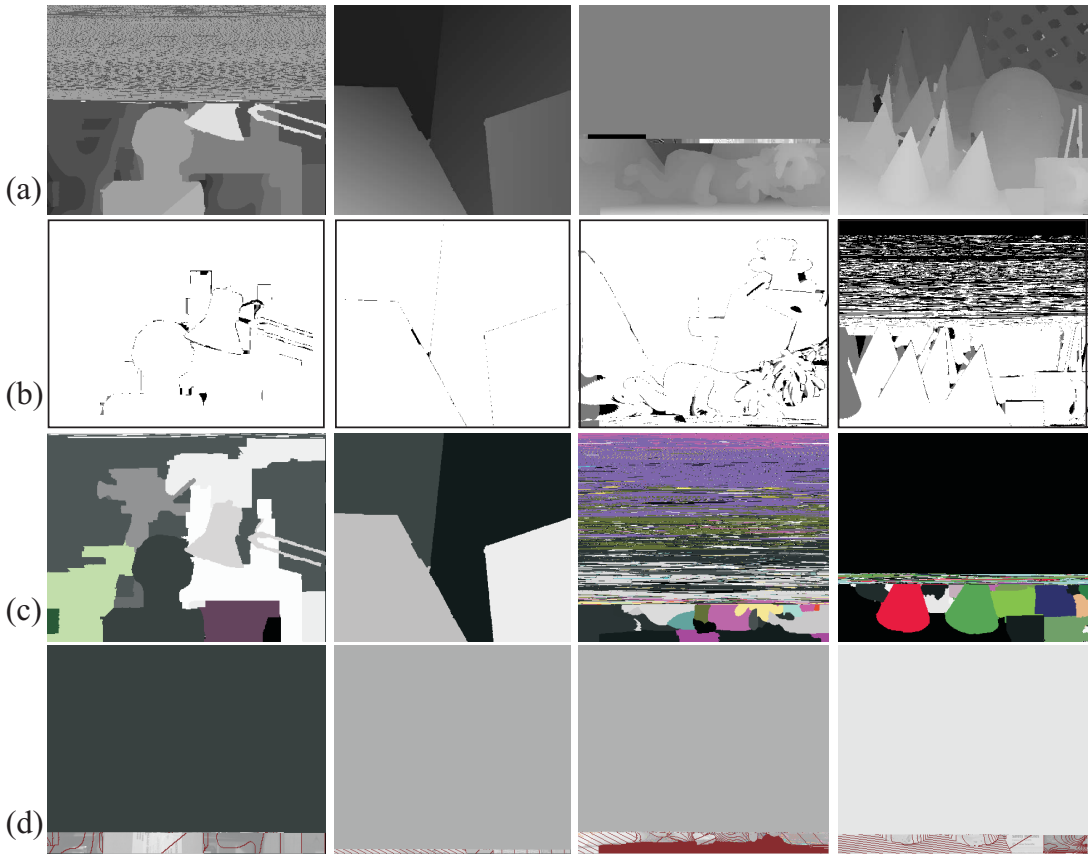


Figure 2. Results on the Middlebury test sets. (a) Disparity maps computed by our method using constant parameters. (b) Disparity errors > 1 pixel computed by comparison against the ground truth. (Black and gray pixels show errors in unoccluded and occluded image areas, respectively.) (c) Assignment of pixels to surfaces. Pixels of the same color lie on the same surface according to our solution. Colored surfaces represent B-splines and grey-valued surfaces correspond to planes. (d) Contour lines overlayed on the left input image.

Algorithm	Rank	Avg. Error	Tsukuba			Venus			Teddy			Cones		
			nocc	all	disc	nocc	all	disc	nocc	all	disc	nocc	all	disc
All Terms On	6	4.06	1.28_{20}	1.65_{12}	6.78_{23}	0.19_{13}	0.28_{8}	2.61_{20}	$\mathbf{3.12}_{1}$	$\mathbf{5.1}_{1}$	$\mathbf{8.65}_{1}$	2.89_{7}	7.95_{8}	8.26_{13}
Standard Asym. Occ.	6	4.65	1.16_{12}	1.44_{8}	6.20_{16}	0.21_{15}	0.30_{8}	2.87_{22}	5.33_{11}	7.06_{4}	12.40_{6}	2.87_{7}	7.87_{7}	8.11_{10}
Curvature Off	12	5.17	1.26_{18}	1.74_{14}	6.73_{21}	1.54_{53}	1.71_{45}	6.47_{43}	3.91_{3}	8.79_{8}	11.40_{4}	2.74_{4}	8.07_{9}	7.71_{6}
MDL Off	13	5.26	1.47_{28}	2.02_{28}	7.38_{32}	0.55_{34}	0.77_{27}	7.10_{46}	5.21_{11}	6.93_{3}	12.56_{6}	2.8_{7}	8.67_{15}	7.74_{6}
Soft Seg. Off	35	7.06	1.59_{31}	2.53_{34}	7.99_{36}	1.42_{53}	1.86_{46}	12.3_{55}	6.81_{22}	12.1_{21}	13.10_{10}	4.38_{38}	10.10_{33}	10.5_{32}

Table 1. Middlebury results for using our method in conjunction with different parameter settings. Values in the table represent error percentages measured in different image regions. We use the table to evaluate the individual terms of our energy. (More information is given in the text.) In the Middlebury ranking, our method currently takes the sixth rank (see entry *All Terms On*). For the Teddy set, our method is the top performer for all error measures (bold numbers).

References

[1] S. Birchfield and C. Tomasi. Multiway cut for stereo and motion with slanted surfaces. In *ICCV*, 1999.

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

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

[4] M. Bleyer, M. Gelautz, C. Rother, and C. Rhemann. A stereo approach that handles the matting problem via image warping. In *CVPR*, pages 501–508, 2009.

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

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

[7] O. Duchenne, J. Audibert, R. Keriven, J. Ponce, and F. Segonne. Segmentation by transduction. In *CVPR*, 2008.

[8] D. Hearn and M. Baker. *Computer Graphics with OpenGL (3rd edition)*. pp. 448–452, 2004.

[9] H. Hirschmüller. Stereo processing by semiglobal matching and mutual information. *PAMI*, 30(2):328–341, 2008.

[10] D. Hoiem, C. Rother, and J. M. Winn. 3d layoutcrf for multi-view object class recognition and segmentation. In *CVPR*, 2007.

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

[12] H. Ishikawa. Higher-order gradient descent by fusion-move graph cut. In *ICCV*, 2009.

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

[14] P. Kohli, M. Kumar, and P. Torr. P3 & beyond: Solving energies with higher order cliques. In *CVPR*, 2007.

[15] V. Lempitsky, S. Roth, and C. Rother. Fusionflow: Discrete-continuous optimization for optical flow estimation. In *CVPR*, 2008.

[16] V. Lempitsky, C. Rother, and A. Blake. Logcut - efficient graph cut optimization for markov random fields. In *ICCV*, 2007.

[17] M. Lin and C. Tomasi. Surfaces with occlusions from layered stereo. In *CVPR*, pages 710–717, 2003.

[18] A. S. Ogale and Y. Aloimonos. Stereo correspondence with slanted surfaces: critical implications of horizontal slant. In *CVPR*, pages 568–573, 2004.

[19] C. Rother, P. Kohli, W. Feng, and J. Jia. Minimizing sparse higher order energy functions of discrete variables. In *CVPR*, pages 1382–1389, 2009.

[20] C. Rother, V. Kolmogorov, V. Lempitsky, and M. Szummer. Optimizing binary mrfs via extended roof duality. In *CVPR*, 2007.

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

[22] B. Smith, L. Zhang, and H. Jin. Stereo matching with non-parametric smoothness priors in feature space. In *CVPR*, pages 485–492, 2009.

[23] J. Sun, Y. Li, S. Kang, and H. Shum. Symmetric stereo matching for occlusion handling. In *CVPR*, volume 25, pages 399–406, 2005.

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

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

[26] Y. Wei and L. Quan. Asymmetrical occlusion handling using graph cut for multi-view stereo. In *CVPR*, volume 2, pages 902–909, 2005.

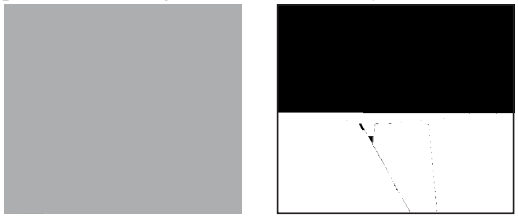
[27] O. Woodford, I. Reid, P. Torr, and A. Fitzgibbon. On new view synthesis using multiview stereo. In *BMVC*, volume 2, pages 1120–1129, 2007.

[28] O. Woodford, P. Torr, I. Reid, and A. Fitzgibbon. Global stereo reconstruction under second order smoothness priors. In *CVPR*, pages 1–8, 2008.

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

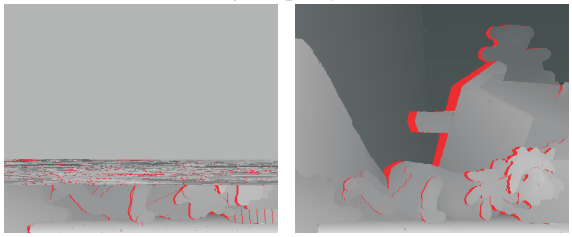
(a) (b) (c)

Figure 3. MDL prior. (a) Crop of the Cones image. (b) Results without using our MDL prior. Surface labels are shown left and disparity results are shown right. The background is modeled by a large set of different surfaces. (c) Results using our MDL prior. Large parts of the background are modeled by the same surface.



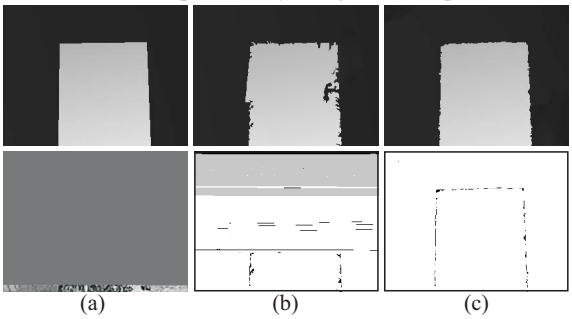
(a) (b)

Figure 4. The effect of turning the curvature prior off. (a) Contour lines taken from the computed disparity map and overlayed on the left image. (This image should be compared against figure 2d). Due to low texture and no penalization of curvature, the spline in the background (top left part of the image) erroneously adapts to the data. This leads to a large disparity error shown in (b).



(a) (b)

Figure 5. Standard versus improved asymmetric occlusion model. (a) Result of the standard asymmetric model. Occlusions (colored red) are wrongly detected on slanted surfaces. (b) Our occlusion model. We avoid this problem by using the concept of surfaces.



(a) (b) (c)

Figure 6. Soft versus hard segmentation. (a) Ground truth disparities and left image of the map data set. (b) We transform our algorithm into a hard segmentation method by setting $\lambda_{seg} := \infty$. The resulting disparity map is shown on top and disparity errors at the bottom. Since segmentation fails on the Map reference image, there occur large disparity errors. (c) Our soft segmentation method can successfully recover from erroneous segmentation. (We use the same parameters as for figure 2.)