Object Stereo — Joint Stereo Matching and Object Segmentation

Michael Bleyer[1]* Carsten Rother[2] Pushmeet Kohli[2] Daniel Scharstein[3]† Sudipta Sinha[4]
[1]Vienna University of Technology, Vienna, Austria
[2]Microsoft Research Cambridge, Cambridge, UK
[3]Middlebury College, Middlebury, USA
[4]Microsoft Research Redmond, Redmond, USA

Abstract

This paper presents a method for joint stereo matching and object segmentation. In our approach a 3D scene is represented as a collection of visually distinct and spatially coherent objects. Each object is characterized by three different aspects: a color model, a 3D plane that approximates the object's disparity distribution, and a novel 3D connectivity property. Inspired by Markov Random Field models of image segmentation, we employ object-level color models as a soft constraint, which can aid depth estimation in powerful ways. In particular, our method is able to recover the depth of regions that are fully occluded in one input view, which to our knowledge is new for stereo matching. Our model is formulated as an energy function that is optimized via fusion moves. We show high-quality disparity and object segmentation results on challenging image pairs as well as standard benchmarks. We believe our work not only demonstrates a novel synergy between the areas of image segmentation and stereo matching, but may also inspire new work in the domain of automatic and interactive object-level scene manipulation.

1. Introduction

In the last two decades much high-quality research has been conducted in the areas of image segmentation and stereo matching. There is some overlap in these efforts, as many stereo methods—in fact, nearly all of the top-ranked methods in the Middlebury benchmark [20]—use image segmentation in some way. However, existing stereo methods typically use low-level segmentation methods to over-segment the image into superpixels. In contrast, in this work we push the idea of combining segmentation and stereo matching to the next level—the object-level.

To achieve this, we extend recent ideas of object-level segmentation in 2D images to 3D scenes. In particular we build upon the body of work that uses higher-order Markov Random Field models for image segmentation, which originates from the seminal work of graph cut-based, interactive image segmentation [19, 5].

We model a 3D scene as a collection of 3D objects. We assume that (1) each object is compact in 3D, (2) each object is connected in 3D, and (3) all visible parts of an object share a similar appearance. In addition, we (4) favor scene interpretations with a few large objects over those with many small objects. Finally, we assume standard stereo photo consistency, which means we expect matching objects to have similar colors across the two input views. Figure 1 shows a sample result (object grouping and disparities) produced by our method.

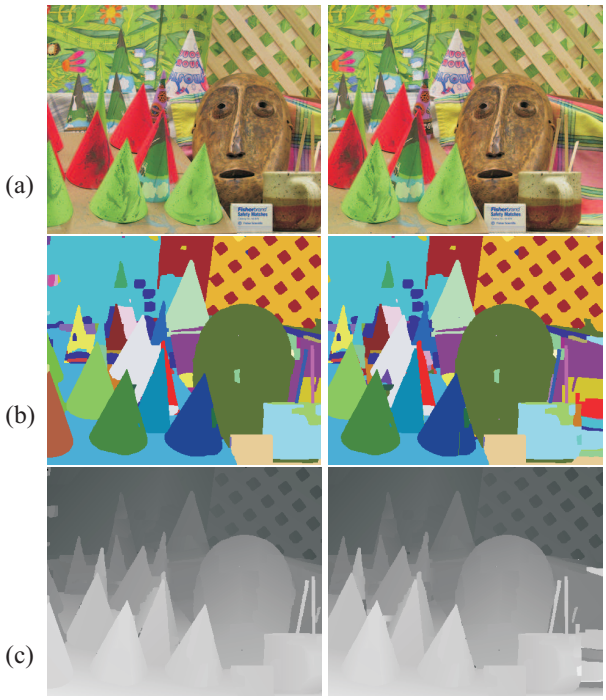


Figure 1. Our approach for joint segmentation and stereo matching. (a) The Cones input stereo pair [20]. (b) Extracted objects. Pixels of the same color belong to the same object. (c) Computed disparity map. Note that our method can recover surfaces with complex geometry, and assign disconnected surface patches to the same object. For Cones, our method achieves the first rank in the Middlebury ranking [20].

Before we formally define our scene model, let us emphasize the key advantages of the above prior assumptions in the context of stereo matching.

The first prior, compactness, is used in virtually all work on 2D image segmentation and stereo matching. Here we encode it in several ways: we assume that (1) objects are coherent, i.e., most pairs of neighboring pixels in one view

*Michael Bleyer received financial support from the Vienna Science and Technology Fund (WWTF) under project ICT08-019.
†Daniel Scharstein was supported by NSF grant IIS-0917109.

belong to the same object; (2) depth variations within an object are smooth; and (3) objects have a bias towards being planar in 3D. Note that the idea of a planar bias is different from the assumption that all objects are planar, which many previous approaches have made.

The second prior, 3D connectivity of objects, has not, to the best of our knowledge, been used in the context of stereo matching. It states that disconnected 2D regions in an image may belong to the same object only if they are separated by an occluding object with smaller depth (i.e., closer to the camera). Recent work on 2D interactive segmentation employs similar constraints with impressive results [24]. In the 2D setting, however, user input is needed to confirm that an object is indeed 2D connected. In contrast to this, 3D connectivity is virtually always true, hence we can utilize it in a fully automatic system. The success of our technique in the presence of difficult occlusions can be observed in the upper right region of Figure 1b, where the background surface visible through the holes in the wooden lattice is accurately detected as a single object despite being disconnected in 2D. At the same time, the two green cones in the front of the scene cannot be grouped into a single object as there is no occluding object in front of them.

The third prior, similar appearance, is the standard self-similarity term used in image segmentation [19]. It is inspired by the fact that each object in a scene has a compact distribution of colors. In this work we use color as the only appearance cue, but other features such as texture could be used as well. The fourth prior, encouraging scene interpretations with few objects, prevents single pixels from being explained as individual objects and has been successfully used in object segmentation [8, 14].

The above assumptions translate into two important properties of our method. The first and obvious one is that our color models introduce a color segmentation into the stereo matching process. Untextured regions with homogeneous colors are well described by a single color model and get hence assigned to the same object. Together with the planar bias, this property allows our algorithm to extend disparities into untextured regions and to precisely capture disparity discontinuities.

The second property goes beyond that and is to our knowledge not present in existing stereo methods. It concerns the problem of assigning disparities to small disconnected background regions in the presence of complex occlusions. For instance, consider again the small green background regions visible through the holes in the wooden lattice in Figure 1. For existing stereo methods, which consider each such region in isolation, assigning the correct disparity is difficult or even impossible due to the potential lack of texture and partial (or perhaps complete) occlusion in the other view. In both cases, the smoothness term would favor assigning the foreground disparity. For a human observer, on the other hand, the foreground/background assignment is easy based on surface colors. Our algorithm, employing object and color models as well as occlusion reasoning, is similarly able to assign the correct disparities. This is true even for the surface patches along the right image edge, which are fully occluded in the other view. In the same way, our method can also handle background regions that are completely untextured and thus contain no disparity cues.

Let us briefly consider possible applications of our work. Aside from being able to accurately reconstruct difficult stereo scenes, our work may also enable and inspire new work in the domain of automatic and interactive object-level scene manipulation. As point-and-shoot stereo cameras are entering the consumer market, there is a pressing need to advance object-level segmentation from 2D to 3D to enable better image manipulation and editing techniques. A 3D object-level segmentation such as provided by our method would also be useful for interactive extraction of foreground regions, as well as for 2D/3D inpainting. Another example is image re-targeting where it has recently been shown that objects with depth give improved results [17].

2. Related Work

As mentioned, many stereo methods perform a color segmentation of one of the images in a pre-processing step and use the resulting segments either as a hard [23, 2] or soft [21, 26, 4] constraint.

Early methods employing hard constraints (e.g., [23]) assume that each color segment can be modeled with a single disparity plane, which fails if segments straddle depth boundaries. Thus, an oversegmentation into very small segments ("superpixels") is often used. In contrast, our method jointly computes segmentation and depth, aiming to recover large segments corresponding to entire objects, and using a symmetric process that yields a consistent segmentation of both images.

More recent methods, including ours, use segmentation as a soft constraint, meaning they prefer solutions consistent with a given color segmentation, but also allow for deviation — typically at the price of higher costs in the energy model. In the simplest case, a soft segmentation method is derived by adjusting the pair-wise smoothness penalty with the output of a color segmentation algorithm [26]. However, due to the MRF shrinking bias, this often does not preserve the segmentation well. Sun et al. [21] bias the disparity map towards the disparity result of a hard segmentation method by adjusting the data term, which is problematic if the hard segmentation is inaccurate.

The soft segmentation method of [4] uses higher-order cliques to make the stereo result consistent with a precomputed segmentation, but optimizing these higher-order cliques is difficult and time consuming. In contrast, our color models require only an additional unary data term,

which does not affect the run time.[1] Like [4] we employ an MDL term (but apply it at the object rather than the surface level) and perform optimization using fusion moves [16].

We are aware of only one paper [22] that employs color models in stereo matching. In this work, an oversegmentation of one input image is refined using single Gaussians to model color within each small segment. However, the color models are not used during matching, and thus the method cannot handle the difficult occlusion cases described above.

From the perspective of object segmentation, [10] presents a method for bilayer segmentation of stereo input into foreground and background regions. The authors also use color models. However, the most distinct difference to our work is that we segment the scene into an arbitrary number of objects and not just two. Furthermore, in contrast to our work, the two objects have no influence on the depth estimation other than leaving the disparity change across the two objects unconstrained.

Finally, there is an interesting connection to the recent paper by Ladicky et al. [15] on joint stereo matching and object recognition. The main difference to our work is that their approach requires training a classifier for predefined object classes in an offline learning phase, whereas our method extracts object segments during runtime in an unsupervised manner. While we cannot expect to compete with the authors' object-level results, our approach has the important advantage that it does not need to know the object classes in advance and hence is considerably more general.

3. Our Model

We now describe our joint object and disparity scene model.

Scene Representation As stated above, we represent the scene as a collection of objects. A crucial point is how to describe the depth of an object. We make the assumption that an object's depth can roughly be estimated by a 3D plane and call this plane *object plane*. Obviously, this assumption is an oversimplification and will not hold true for many real-world objects. For example, consider the complex 3D shape of the mask in the Cones test set of Figure 1. Our solution is to add an additional level of detail on top of the object planes in order to model depth variations within an object. This is inspired by the plane-plus-parallax model [13].

To implement this level of detail, we compute a separate disparity map. It is important to note that this disparity map is not computed independently from the object planes. Instead, we compute a parallax value at each pixel p within an object o_p. This parallax is obtained by subtracting p's disparity according to o_p's object plane from its disparity according to the disparity map. The idea is to put constraints on the parallax values, i.e., we enforce that parallax values have a compact distribution within object o_p. This distribution is stored in the so-called *parallax model* of object o_p. The parallax model provides the probability of the occurrence of a specific parallax in object o_p and our model tries to avoid parallaxes that have low probabilities.

We represent the disparity map that provides the additional level of detail using a *surface-based representation*. More precisely, the disparity map is a collection of 3D planes. We call these planes *depth planes*. Note that these depth planes are considerably smaller than objects in their spatial extents and can consequently capture the desired depth details within an object.

Notation Let $\mathcal{I}$ be the coordinates of all pixels of left and right images. Further, let $\mathcal{O}$ denote the set of all possible objects and $\mathcal{F}$ the set of all possible depth planes. An object $o \in \mathcal{O}$ contains the following parameters: (1) a color model, (2) a parallax model and (3) an object plane, where the latter two parameters are used to regularize the disparity map F, as discussed above. We search for two mappings (1) $F : \mathcal{I} \rightarrow \mathcal{F}$ that assigns each pixel to a depth plane, and (2) $O : \mathcal{I} \rightarrow \mathcal{O}$ that assigns each pixel to an object.

Note that F is an implicit disparity map. If a pixel p is assigned to a depth plane f_p, we can compute disparity by $d_p = f_p[a] \cdot p_x + f_p[b] \cdot p_y + f_p[c]$ where a, b and c denote plane parameters and p_x, p_y are p's image coordinates.

We now define an energy function that evaluates the quality of F and O, which implicitly defines a probability distribution over object and depth plane labellings. We then try to minimize the energy to obtain a "good" approximation to the Maximum a Posteriori (MAP) solution of the model. Formally, we define the energy as:

$$E(F,O) = E_{pc}(F,O) + E_{oc}(O) + E_{dc}(F,O) + E_{col}(O) + E_{par}(F,O) + E_{mdl}(O) + E_{con}(F,O) \quad (1)$$

where different terms correspond to the different likelihood and prior terms of our model. Note, the parameters for each object, e.g., its color model, are also part of the energy function but are omitted for ease of notation and will be explained in detail when appropriate. The individual energy terms are defined next.

Photo Consistency Term E_{pc} The photo consistency term measures the pixel dissimilarity of corresponding points and accounts for occlusion handling. Moreover, it ensures that corresponding pixels are assigned to the same depth plane and object. We first give the definition of our data term and then provide an explanation in the paragraphs

[1] The recent work of [25] has shown that our global color model term can be transformed into (not tractable) higher-order cliques. Furthermore, it has shown that a simple iterative procedure (where color models are data terms) does perform equally well as a complex inference procedure of the higher-order model, which justifies our simple, iterative procedure.

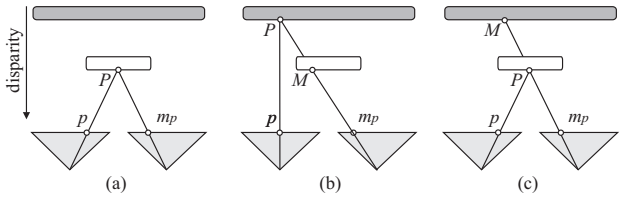


Figure 2. The data term. More information is given in the text. (a) The 3D point P is visible from both cameras. (b) P is occluded by M. (c) It is impossible for the right camera to see M, because its viewing ray is blocked by P.

below. The term is defined as

$$E_{pc}(F,O) = \sum_{p\in\mathcal{I}} \begin{cases} \rho(p,m_p) : f_p = f_{m_p} \wedge o_p = o_{m_p} \\ \lambda_{occ} \quad : \text{otherwise if } d_p < d_{m_p} \\ \infty \quad : \text{otherwise.} \end{cases} \tag{2}$$

Here, m_p denotes the matching point of p in the other view. If p lies in the left view m_p is computed as $p - d_p$ and as $p + d_p$ if p resides in the right view. The constant λ_{occ} is a user-defined penalty for occlusion. The function $\rho()$ computes the dissimilarity between two corresponding pixels. In our implementation, we compute the Birchfield/Tomasi measure [1] and then truncate the resulting value by $\lambda_{occ} - 1$. This truncation ensures that the costs of an occlusion are always higher than that of a visible match, and encourages a smaller number of occluded pixels in the lowest energy (or MAP) solution. Our treatment of occluded pixels is similar to [12], the main difference being the surface-based representation of our disparity map F, which prevents us from detecting wrong occlusions at slanted surfaces [4].

The photo consistency term is able to distinguish between different scenes shown in Figure 2. In the first case (Figure 2a), the pixel p and its matching point m_p both lie on the same depth plane. Hence, they generate the same 3D point P and the data term measures photo consistency. In the second case (Figure 2b), p and m_p lie on different depth planes, which leads to two different 3D points P and M. Furthermore, p has lower disparity than m_p. This case corresponds to an occlusion, since the 3D point M blocks the viewing ray of the right camera so that the right camera cannot see P. Consequently, our data term imposes the occlusion penalty λ_{occ}. Finally, in the third case (Figure 2c), p has higher disparity than m_p. Given the assumption that surfaces are opaque, this case is impossible. The viewing ray of the right camera would have to "see through" P in order to see M. To avoid this case, we assign an infinite penalty to this configuration.

Object-Coherency Term E_{oc} The object-coherence prior encourages neighboring pixels in the image to take the same object label. It takes the form of a standard Potts model prior used in image segmentation [19] and is defined as

$$E_{oc}(F,O) = \sum_{<p,q>\in\mathcal{N}} \lambda_{ocoh} * T[o_p \neq o_q] \tag{3}$$

where λ_{ocoh} is a penalty. T is the indicator function that returns 1 if its argument is true and 0 otherwise.

Depth Plane-Coherency Term E_{dc} This encodes the prior that depth plane assignments within an object shall be spatially coherent. The depth plane-coherency prior is defined as $E_{dc}(F,O) =$

$$\sum_{<p,q>\in\mathcal{N}} \begin{cases} \frac{\lambda_{dcoh}}{2} : o_p = o_q \wedge f_p \neq f_q \wedge |d_p - d_q| \leq 1 \\ \lambda_{dcoh} : o_p = o_q \wedge f_p \neq f_q \wedge |d_p - d_q| > 1 \\ 0 \quad : \text{otherwise} \end{cases} \tag{4}$$

where $\mathcal{N}$ represents the set of all spatial neighbors (4-connectivity) in left as well as right images and λ_{dcoh} denotes a constant penalty.

The depth plane-coherency prior does not penalize a depth plane transition if neighboring pixels are assigned to different objects, since it is reasonable to assume that object discontinuities go along with discontinuities in depth. If there is a depth plane transition within an object, the prior checks whether this transition is smooth in disparity. This means we impose a small penalty if the corresponding jump in disparity is smaller than a pixel. Otherwise, if the jump exceeds the threshold of 1, we impose a larger penalty. Note that disparity discontinuities (larger jumps) inside an object can occur because of self-occlusion, but are less likely than a smooth disparity transition and derive a larger penalty.

Object-Color Term E_{col} Each object contains a color model. This color model is implemented as a Gaussian Mixture Model (GMM) with five mixture components.[2] The GMM gives the probability that a pixel lies inside the object according to its color value. We evaluate the color costs at each pixel using the GrabCut data term [19]

$$E_{col}(O) = \sum_{p\in\mathcal{I}} \lambda_{color} * -\log(\pi(c_p, o_p)) \tag{5}$$

where c_p denotes the color of pixel p and λ_{color} is a penalty for color inconsistency. The function $\pi(c,o)$ returns the probability of color c in the GMM of object o.[3]

The object-color term enforces that an object has a compact set of colors. This compactness is maximized if an object's GMM captures very similar colors with low variance among them. Hence, E_{col} is minimized if every pixel is a separate object. In our energy model, we have two terms that work against a degenerate solution, i.e., the object-coherency prior and (more importantly) the object-MDL term. This avoids an undesired clutter in the object assignments O.

[2] We use the GMM implementation of OpenCV 2.1.

[3] Note we avoid the degenerate case of infinite probability π by assuming a minimum variance for all Gaussians.

Object-Parallax Term E_{par} Our parallax term regularizes the disparities of an object with respect to its object plane. It computes the plane parallax as follows. Let us denote the disparity at pixel p according to o_p's object plane by $d_p^{o_p}$. The parallax is then computed as $d_p - d_p^{o_p}$. The key idea is that distribution of the parallax is likely to be compact.[4] We implement the parallax distribution using a non-parametric histogram.[5] We define the object-parallax term as

$$E_{par}(F, O) = \sum_{p \in \mathcal{I}} \lambda_{parallax} * (1 - \theta(d_p - d_p^{o_p}, o_p)) \quad (6)$$

where $\lambda_{parallax}$ is a penalty for parallax inconsistency. The function $\theta(par, o)$ returns the probability of parallax par in object o's histogram.

Note that analogously to the color term, the parallax term introduces a bias, i.e., it prefers flat objects over rounded ones. For example, if we set $\lambda_{parallax}$ to a high value, the depth planes of an object become identical to the object planes. We believe that this is a good bias, because flat surfaces may be predominant in natural images, especially in man-made environments.

Object-MDL Term E_{mdl} The term puts a penalty on the occurrence of an object and hence prefers solutions that have a small number of objects (MDL principle). We define the object-MDL term as

$$E_{mdl}(O) = \sum_{o \in \mathcal{O}} \lambda_{mdl} * T[\exists p \in \mathcal{I} : o_p = o] \quad (7)$$

where λ_{mdl} is a constant penalty.

3D Connectivity Prior E_{con} The connectivity prior is a global potential in the model. It operates on the object assignments of all pixels O and enforces that objects are always connected in 3D. The potential assigns a cost of 0 to all solutions in which all objects that appear in the labeling are connected. Solutions that do not satisfy this constraint are given infinite cost. In that sense, the connectivity term is more of a constraint than a prior on the solution.

An object is considered connected if there exists a path that connects all pixels with the same object label, such that on the path are either (1) pixels that belong to the same object or (2) pixels that belong to different objects but have higher disparity relative to the disparity of the object plane. Here, case (2) represents a path that is occluded by a foreground surface, which is a valid explanation for 3D connectivity. More formally,

$$E_{con}(F, O) = \sum_{(p,q) \in \mathcal{I}^l, \mathcal{I}^r : o_p = o_q} \begin{cases} 0 : \text{if } C(p,q) = \text{true} \\ \infty : \text{otherwise} \end{cases}, \quad (8)$$

where $\mathcal{I}^l, \mathcal{I}^r$ are left and right input images, respectively. The function C checks for connectivity of any two pixels p and q and is defined as

$$C(p, q) = \exists \Pi : (\prod_{s \in \Pi_q^p} (o_s = o_p) \vee (d_s^{o_p} \leq d_s)), \quad (9)$$

where Π_q^p is any path traversing the 4-connected neighborhood system $\mathcal{N}$, with start-point p and end-point q. $d_s^{o_p}$ represents the disparity of pixel s for object plane of o_p.

The computation of the connectivity prior (8) can be performed using depth or breadth first search in a graph with worst case runtime complexity that is polynomial in the number of edges in the image. The constructed graph contains one node for every pixel s that satisfies the condition:

$$(o_s = o_p) \vee (d_s^{o_p} \leq d_s). \quad (10)$$

The graph contains all edges between the nodes that are present in the original neighborhood system $\mathcal{N}$.

In the case when most objects in the image are convex, a more efficient check can be implemented by just considering the line joining pixels p and q instead of all possible paths. For computational tractability we use the following strategy (see Figure 3). We randomly sample a large set of pairs of points p and q, where p and q belong to different connected components of the object. We then draw a line between p and q and check if all pixels on the line satisfy condition (10).

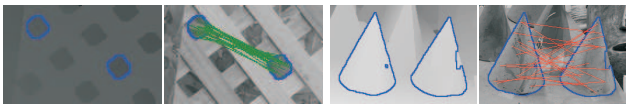


Figure 3. Checking 3D connectivity of two regions. We draw random lines between the regions and check whether these lines are fully occluded according to the computed disparity map. Green lines pass this test, while red ones fail. Since most lines are occluded in the lattice example (left images), we label the two background regions as being 3D connected. The two foreground cones (right images) fail our 3D connectivity test.

4. Inference via Energy Minimization

We minimize the energy defined in the previous section using the fusion move algorithm [26, 16]. The basic roadmap for the inference is as follows. We start with an initial solution S that consists of a disparity map F and an object map O. We then obtain a proposal S' from a proposal generator discussed in Section 4.1. S and S' are fused to produce a new solution S^*. In the fusion move, we generate a new solution by "selecting" some depth planes and

[4] We could have additionally enforced that the distribution of the parallax has a particular form, e.g., singled-peaked distribution with mode at 0. However, we did not find that to be necessary.

[5] We use a histogram instead of GMM, since it is a simple non-parametric model which worked well in this case.

objects from S and others from S'. The challenge is to compute the "optimal" fusion result S^*, i.e., the one that leads to the largest decrease of energy among all possible fusion moves (see Section 4.2). Once a fused solution S^* has been computed, we set $S := S^*$ and fuse the next proposal S'' obtained from the proposal generator. We also perform a refinement step where parameters of the present objects are refit.

4.1. Proposal Generator

We now describe how the different proposals for the fusion move algorithm are generated. The proposal generator first returns *initial proposals*. Once all initial proposals are fused, we run a *refit proposal* and finally fuse all *expansion proposals*. Refit and expansion proposals are iterated three times. We now describe these proposal types.

Initial Proposals We first compute a disparity map F in a way that is equivalent to the *SegPl* proposals of [26, 4]. One of the two input images is selected to serve as a reference frame. We compute an initial disparity map using the fast stereo matcher of [3] as well as a color segmentation of the reference image via mean-shift segmentation [6]. F is now derived by fitting a plane to each color segment using the initial disparity map. To derive O, we take the color segmentation result and group segments that have similar depths according to F.[6] Segments that fall into the same group define the spatial extent of an object o. Object o's parameters are estimated as follows. We initialize o's color model by using all pixels of o as samples for a GMM. The GMM parameters are then computed by OpenCV's EM algorithm. The object plane is derived by fitting a plane according to the disparity map F. To initialize the parallax model, we compute the difference between o's disparities in F and the disparities according to o's object plane and store the resulting parallaxes in a histogram. Due to our symmetrical formulation, we still need to propagate F and O to the second view. This works by image warping, i.e., we generate the second view by warping the plane and object indices according to the disparity map F of the reference view. After warping, we fill the occluded regions by replicating plane and object indices along horizontal scanlines. To derive a large variety of initial proposals we combine different initial disparity maps with different mean-shift segmentations and also use different parameter settings in the depth segmentation algorithm. In total, we have approximately 30 initial proposals.

Refit Proposals A refit proposal $\langle F', O' \rangle$ is derived by refitting the object parameters of the current solution $\langle F, O \rangle$. More precisely, for each object o present in O we compute a new color model, object plane and parallax model according to o's spatial extent in O and the disparity map F. This operation defines O', while F' is set to F.

Expansion Proposals In this type of proposal, we select one depth plane f present in F and one object o present in O where F and O again represent the current solution. The proposal solution $\langle F', O' \rangle$ is then derived by setting all pixels of F' to f and all pixels of O' to o. To keep the number of expansions tractable, we require that there exist at least 500 pixels, i.e., a small fraction of the image dimensions, whose depth plane assignment in F is f and whose object assignment in O is o.

4.2. Computing the Optimal Fusion

To compute the optimal fusion of two proposal solutions, we construct a quadratic pseudo-boolean (QPB) function to represent the energy of the fused solutions. This function may contain non-submodular terms and is NP-hard to minimize in general. We use QPBO-F [11] to derive a solution to the fusion move problem that is guaranteed to have equal or lower energy than our current solution.[7]

For the fusion move, our goal is to compute a boolean variable x_p for each pixel p, where $x_p = 0$ means that p takes the depth plane *and* object label of proposal 1 and $x_p = 1$ means that the labels of proposal 2 are taken. The photo consistency term of equation (2) introduces two pairwise terms at each pixel p. The first one connects p with its matching point m_p according to proposal 1. The second one establishes a connection to the matching point m'_p induced by proposal 2. The costs of these pairwise terms are derived by evaluating the three cases defined in equation (2). The object-coherency prior of equation (3) and the depth plane-coherency term of equation (4) both introduce pair-wise terms between spatially neighboring pixels. The object-color term (equation (5)) and the object-parallax term (equation (6)) are trivial to express, because each of them converts to a unary term in the pseudo-boolean energy. The object-MDL prior in equation (7) defines higher-order cliques over all pixels assigned to the same object in proposals 1 or 2. Here, we follow [14, 7] and express the MDL prior as a $\mathcal{P}^n$ Potts model whose transformation to a pair-wise energy is explained in [9].[8]

[6] Our depth segmentation algorithm tries to find a new mapping F^* of segments to planes that consists of fewer planes, but still represents a good approximation of the disparity map F via energy minimization. The data term of our energy function measures the absolute difference between the disparity of the new mapping F^* and the original disparity map F. The smoothness term penalizes neighboring pixels that are assigned to different depth planes. We optimize this energy by performing an α-expansion for each depth plane f present in F, which is efficient, because we can do it on the segment level. Segments that have the same depth plane assignment in F^* form a single depth segment. We can obtain different depth segmentations by varying the smoothness penalty of the energy function.

[7] QPBO only guarantees to find part of a global optimal labeling and can leave pixels unlabeled. Hence we are not guaranteed to find the optimal fusion move. However, we empirically observed that the percentage of unlabeled pixels is very close or equal to 0 in most cases.

[8] Due to the use of fusion moves, there is no guarantee that the MDL term introduces a $\mathcal{P}^n$ Potts type higher-order clique. The prior can take the form of a general sparse higher-order clique, which can be optimized

Maintaining the 3D connectivity constraint of equation (8) during a fusion move is very hard. In fact, it can be easily shown that the problem of finding the optimal fusion move is NP-hard in the presence of the connectivity term. We therefore remove the connectivity term from the energy when computing the fusion move. One strategy for optimization is "Move & Check". After the fused solution has been found, we can evaluate it with respect to the connectivity term. If some regions of the object fail the connectivity check, we either assign them to different objects, or roll back the move. Because of efficiency reasons, we do the check step after all the moves have been made. We found empirically that this strategy works well.

5. Results

We test our algorithm on standard image pairs and newly-recorded real-world scenes. The algorithm's parameters are set as follows: $\lambda_{occ} = 25$, $\lambda_{ocoh} = 25$, $\lambda_{dcoh} = 25$, $\lambda_{color} = 4$, $\lambda_{parallax} = 2$ and $\lambda_{mdl} = 100000$.[9] These parameters have been estimated empirically in order to optimize quality of disparity results.

We use the commonly-accepted Middlebury benchmark [20] to evaluate our method. In this benchmark, our method currently takes rank eight among approximately 90 methods. We have found that our algorithm works particularly well on the Cones test sets. When sorting the Middlebury table according to Cones, our algorithm is the top performer. Results for the four Middlebury images are shown in Figures 1, 4 and 5. For quantitative results, the reader is referred to the Middlebury online table.

Let us look at the Teddy results in Figure 4. As seen in Figure 4b, our algorithm accomplishes to segment the scene into a small number of large objects based on their color, depth and 3D connectivity. A particularly interesting case is marked by the yellow rectangle. There is a small isolated white region (see arrow) that is only visible in the right image. Common algorithms would fill in disparity for this region via spatial disparity propagation, i.e., they extrapolate the disparity of neighboring visible pixels. Note that in the case depicted in Figure 4, this form of propagation has to fail, because for this region, none of its spatial neighbors carries the correct (background) disparity. In contrast, our algorithm combines this isolated region and the region above the flower pot to a single object due to their colors being similar. Together the regions form an object that corresponds to the wall of the toy house. Since the isolated white region is now a member of the object wall, its disparity is biased towards the wall's object plane. Therefore, we can reconstruct the correct depth. This grouping also makes sense in terms of 3D connectivity. The two regions can be connected in 3D, because there is an occluder (the flower pot) between them.

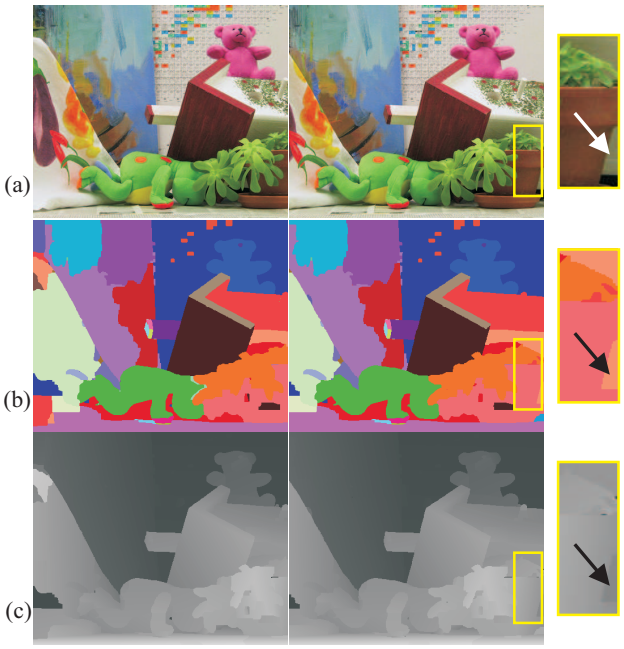


Figure 4. Results on the Teddy images. (a) Input images. (b) Extracted objects. (c) Disparity results. Our algorithm propagates reasonable depth information to completely occluded regions using color information (see region marked by the yellow rectangle).

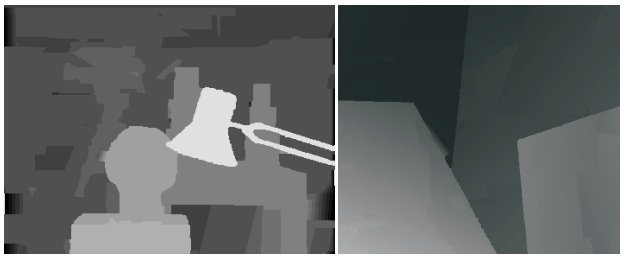


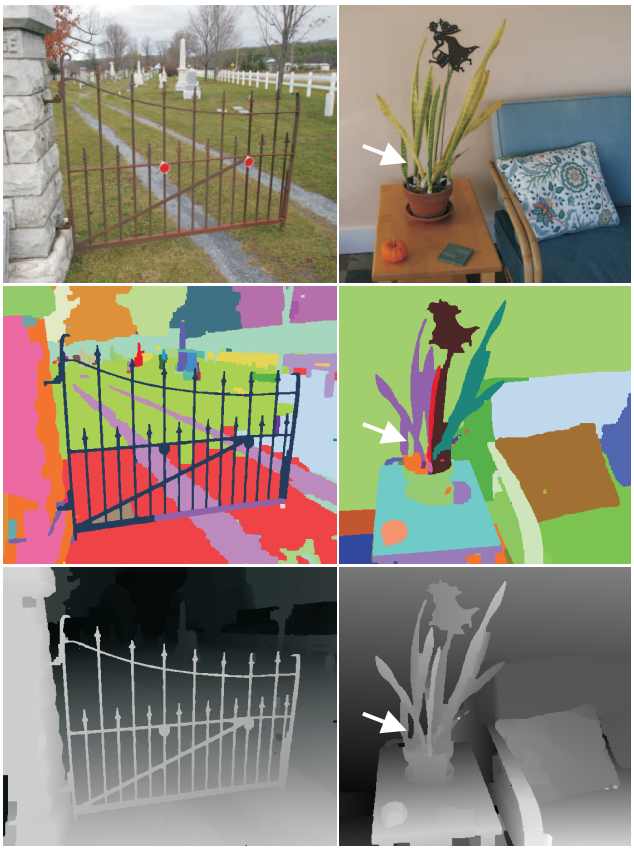
Figure 5. Disparity results for the Tsukuba and Venus sets.

Apart from using standard images, it is important to test our method on real-world stereo pairs. In the Cemetery test pair of Figure 6a, our algorithm can correctly reconstruct the thin structure of the gate. It also accomplishes segmentation of the image into meaningful objects, e.g., see the two road lanes. The effect of our 3D connectivity term is depicted by looking at the tomb stones that are similar in color and depth and hence would form a "good" object. However, they are not connected in 2D and there is also no occluder present, which would represent an explanation for the tomb stones being connected in 3D. Hence the 3D connectivity term splits them into separate objects.

Finally, the Fairy test set of Figure 6b illustrates another case that we believe to be challenging for competing algorithms. Here, the challenge is the lack of texture at the background wall. In particular, the small region marked by the

with the construction introduced in [18]. However, this case does not occur when using our sequence of proposals explained in Section 4.1.

[9] We keep this parameters constant except for Figure 6b, where we set $\lambda_{occ} = 15$.



(a) (b)

Figure 6. Results on newly-recorded real-world scenes. (a) The Cemetery image pair. Our method can successfully reconstruct the thin structure of the gate. (b) The Fairy test set. Our method accomplishes to reconstruct the background, although there is very little texture present. This also works for the small isolated region (white arrow), because our algorithm looks at its color.

arrow poses a problem. The spatial smoothness term erroneously motivates to assign this region to the foreground disparity and the data term is not able to balance this out due to no texture being present. Our algorithm looks at the color of this small region. It finds that the color matches that of the background and biases the disparity towards the object plane of the background object.

6. Conclusions

We have proposed a combined algorithm for stereo matching and object segmentation. Our model represents the scene as a small collection of objects. Objects are assumed to be approximately planar and incorporate a color model. The object level enables our algorithm to utilize color segmentation as a soft constraint and to handle difficult occlusion cases, which cannot be accomplished by competing stereo methods. Furthermore, we have introduced a 3D connectivity constraint that enforces consistency of object assignments with stereo geometry.

Further work should concentrate on improving the algorithm's run time. Currently, our algorithm is slow, i.e., it takes approximately 20 minutes to obtain results on images such as the ones from the Middlebury set. Furthermore, the optimization of the 3D connectivity constraint represents an open topic.

References

[1] S. Birchfield and C. Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *TPAMI*, 20(4):401–406, 1998.

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

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

[4] M. Bleyer, C. Rother, and P. Kohli. Surface stereo with soft segmentation. In *CVPR*, 2010.

[5] Y. Boykov and M. Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In *ICCV*, 2001.

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

[7] A. Delong, A. Osokin, H. Isack, and Y. Boykov. Fast approximate energy minimization with label costs. In *CVPR*, 2010.

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

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

[10] V. Kolmogorov, A. Criminisi, A. Blake, G. Cross, and C. Rother. Bi-layer segmentation of binocular stereo video. *CVPR*, 2005.

[11] V. Kolmogorov and C. Rother. Minimizing non-submodular functions with graph cuts - a review. *PAMI*, 29(7):1274–1279, 2007.

[12] V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. In *ECCV*, volume 3, pages 82–96, 2002.

[13] R. Kumar, P. Anandan, and K. Hanna. Direct recovery of shape from multiple views: A parallax based approach. In *ICPR*, 1994.

[14] L. Ladicky, C. Russell, P. Kohli, and P. H. Torr. Graph cut based inference with co-occurrence statistics. In *ECCV*, 2010.

[15] L. Ladicky, P. Sturgess, C. Russell, S. Sengupta, Y. Bastanlar, W. Clocksin, and P. Torr. Joint optimisation for object class segmentation and dense stereo reconstruction. In *BMVC*, 2010.

[16] V. Lempitsky, C. Rother, and A. Blake. Logcut - efficient graph cut optimization for Markov Random Fields. In *ICCV*, 2007.

[17] A. Mansfield, P. Gehler, L. V. Gool, and C. Rother. Scene carving: Scene consistent image retargeting. In *ECCV*, 2010.

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

[19] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23:309–314, 2004.

[20] 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/.

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

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

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

[24] S. Vicente, V. Kolmogorov, and C. Rother. Graph cut based image segmentation with connectivity priors. In *CVPR*, 2008.

[25] S. Vicente, V. Kolmogorov, and C. Rother. Joint optimization of segmentation and appearance models. In *ICCV*, 2009.

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