

# Layered Scene Decomposition via the Occlusion-CRF

Chen Liu<sup>1</sup>   Pushmeet Kohli<sup>2</sup>   Yasutaka Furukawa<sup>1</sup>

<sup>1</sup>Washington University in St. Louis

<sup>2</sup>Microsoft Research

## Abstract

*This paper addresses the challenging problem of perceiving the hidden or occluded geometry of the scene depicted in any given RGBD image. Unlike other image labeling problems such as image segmentation where each pixel needs to be assigned a single label, layered decomposition requires us to assign multiple labels to pixels. We propose a novel “Occlusion-CRF” model that allows for the integration of sophisticated priors to regularize the solution space and enables the automatic inference of the layer decomposition. We use a generalization of the Fusion Move algorithm to perform Maximum a Posterior (MAP) inference on the model that can handle the large label sets needed to represent multiple surface assignments to each pixel. We have evaluated the proposed model and the inference algorithm on many RGBD images of cluttered indoor scenes. Our experiments show that not only is our model able to explain occlusions but it also enables automatic inpainting of occluded/invisible surfaces.*

## 1. Introduction

The ability of humans to perceive and infer the geometry of their environment goes beyond what is directly visible. Given a scene with desks on a floor, we see desks in the foreground, and naturally infer the floor and the walls as a complete surface behind. However, this information is not represented by commonly used data structures such as a depth map or for that matter a RGBD image which only explains visible surfaces.

Occlusions lie at the heart of this problem, and have been a challenge for reconstruction algorithms since the dawn of Computer Vision. For instance, partial occlusion of objects makes it challenging for us to perform 3D reconstruction, and cause rendering artifacts such as holes or texture-stretching. This has inspired the proposal of many problem-specific techniques such as anisotropic diffusion [15], symmetric image matching [20] and segmenta-

tion based stereo [2].

One of the main challenges of handling occlusions emanates from the fact that to accurately model geometry in the physical world, we need to go beyond the 2D image representation. One approach to alleviate the occlusion problem is to lift the domain from 2D to 3D, that is, from a depthmap to a 3D voxel grid [24, 8]. However, the surface geometry is inherently 2D, and the 3D voxel representation does not efficiently use its modeling capacity. Injecting semantic information and representing a scene as a room layout and objects [7, 28] (or a block world for outdoors [8]) is another effective solution to resolve the occlusion problem. However, these approaches focus on scene understanding rather than precise geometric description, severely limiting their potential high-end applications in Computer Graphics and Robotics.

This paper introduces a novel layered and segmented depthmap representation, which can naturally handle occlusions and inpaint occluded surfaces. We use a generalization of the Fusion Move algorithm to perform Maximum a Posterior (MAP) inference on the model, that can handle the large label sets needed to represent multiple surface assignments to each pixel. The optimization procedure works by repeatedly proposing a sub-space defined by multiple plausible solutions and searches for the best configuration within it using tree reweighted message passing (TRW-S) [10]. Our experiments show that our Fusion Space algorithm for performing inference is computationally efficient and finds lower energy states compared to competing methods such as Fusion Move and general message passing algorithms. Notice that this new representation and optimization scheme are orthogonal to the active semantic reconstruction research, and can be readily available for any other method to use. The technical contributions of this paper are two fold: 1) A novel layered and segmented depthmap representation that can naturally handle occlusions; and 2) A novel Fusion Space optimization approach that is more efficient and effective than the current state-of-the-art.

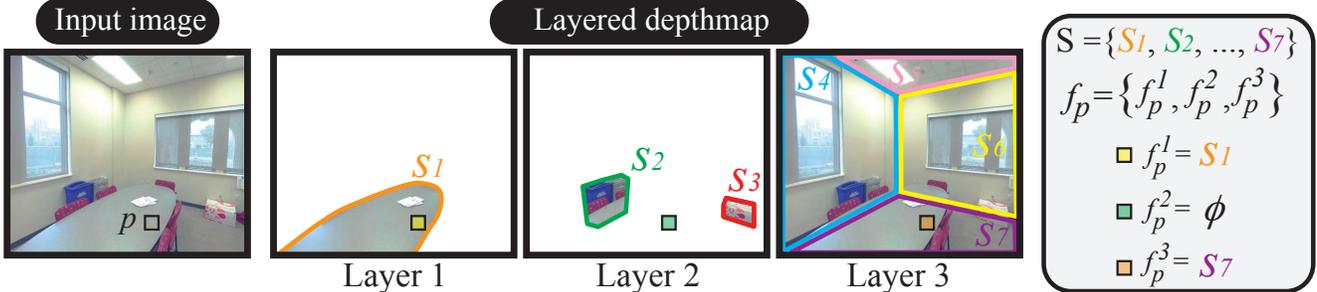


Figure 1. Occlusion-CRF model decomposes an input RGBD image into layers of segmented depthmaps. The model can naturally handle depth discontinuities and represent occluded/invisible surfaces. The key innovation in our model is that a single variable  $f_p$  encodes the states (assigned surfaces) of all the layers. This encoding enables us to represent complicated constraints and interactions by standard unary or binary potentials. A set of surface segments ( $S$ ) are adaptively generated during the optimization. Each segment is either a planar or b-spline surface as in surface stereo [2].

## 2. Related work

Layered representations for visual data have a long history in computer vision [25]. However, most of this work looked at image sequences of moving objects where all parts of the scene were visible in some image. For instance, Wexler *et al.* [26] presented a method of completing the unseen parts of the scene in any given frame of a video sequence by copying content from neighboring frames. In a similar spirit, Shade *et al.* [17] proposed a novel representation, *layered depth image (LDI)*, in which multiple depths may exist for a single line of sight. Zheng *et al.* [29] proposed a layered analogue of panoramas called *layered depth panorama (LDP)* that is a multi-perspective cylindrical disparity space. The multi-perspective nature allows them to represent scene geometry that are invisible in the current view. However, the inference of LDP mentioned in [29] requires multiple images and cannot convert an RGBD image into the representation.

A 3D voxel can naturally model occluded surfaces and is, in a sense, an ultimate representation [9]. However, the model suffers from severe staircasing artifacts, since the 3D representation lacks in its ability to reason and enforce surface smoothness such as planes or b-spline surfaces. Recent work on parsing indoor scenes is able to reason about occluded regions in a scene but they have to rely on either priors on the geometry of the environment like indoor scenes [12, 16, 27, 28] or object category information [21].

More recently, the problem of hallucinating the occluded parts of the scene has attracted a lot of research [6, 7, 19]. While these methods produce good completion results, their modeling capacity is limited because of their reasoning over one or two layers. Approaches like [2] try to overcome this restriction but they are not able to infer extent of the surfaces. In contrast, our model is able to perform joint inference over occluding layers in the scene, allowing us to decompose the scene into layers where surface extents are well defined.

## 3. Occlusion-CRF model

The inspiration of our occlusion-CRF model comes from the *surface stereo algorithm* [2] by Bleyer *et al.*, which models an image as a segmented depthmap with piecewise planar or b-spline surfaces. Our representation is essentially a stack of segmented depthmaps. The layered representation enables us to model occluded surfaces naturally without sharp depth discontinuities with the use of an *empty* label (See Fig. 1). The key innovation of our layered scene representation is that a single variable encodes the states of all the layers per pixel, as opposed to associating one variable per pixel in each layer. The advantage of this encoding is that 1) the visibility constraint (i.e., the first non-empty layer must be consistent with the input depth) can be pre-computed into unary terms; and 2) the interaction of multiple layers can be represented by standard pairwise terms. These constraints and interactions would require complex higher-order relations otherwise. Note that this is different with simply concatenating variables together ([5]) in the sense that we indeed infer one variable instead of a vector of variables for each pixel. Our idea shares similar spirit with [2, 3, 13, 14] that using variable cliques to model higher order relations.

Given an RGBD image, we seek to decompose the scene into  $L$  layers of segmented depthmaps, where each segment is either a planar surface, a b-spline surface, or empty ( $L = 4$  in our experiments). Surface segment candidates  $\mathcal{S}$  are dynamically generated during optimization as in *surface stereo* (See Sect. 5).  $f_p^l \in \mathcal{S}$  denotes the surface ID assigned to pixel  $p$  at the  $l$ <sup>th</sup> layer ( $f_p^l$  can be empty). A tuple of  $L$  surface IDs encodes both the visible and occluded surfaces at each pixel:  $f_p = \{f_p^1, f_p^2, \dots, f_p^L\}$ .

Layered scene decomposition can be formulated as an energy minimization problem over  $\mathcal{F} = \{f_p | p \in \mathcal{I}\}$ , where  $\mathcal{I}$  denotes the image domain and  $\mathcal{F}$  is a set of variables. We enforce the background (i.e.,  $L$ <sup>th</sup>) layer (usually contains the room structure) to be non-empty, which also ensures that

at least one non-empty surface is assigned to every pixel.

#### 4. Energy function

Our ‘‘Occlusion-CRF’’ model integrates sophisticated priors to regularize the solution space and enables the automatic inference of the layer representation. In particular, our energy  $E(\mathcal{F})$  consists of six terms:

$$E(\mathcal{F}) = E_{data}(\mathcal{F}) + E_{smooth}(\mathcal{F}) + E_{MDL}(\mathcal{F}) + E_{curv}(\mathcal{F}) + E_{Convex}(\mathcal{F}) + E_{parallax}(\mathcal{F}).$$

This energy 1) respects input depth values (data term), 2) pushes depth discontinuities across layers, enabling layered analysis (smoothness term), 3) prefers convex surface segments with minimal boundaries, effectively inpainting occluded surfaces (smoothness and convex terms), and 4) prefers fewer segments, suppressing noise (MDL term). The curvature term accounts for the different degrees of freedom of the two surface types (i.e., planar or b-spline). The parallax term realizes long-range interactions without densifying the connections too much.

**Data term:** The data term has four components:

$$E_{data}(\mathcal{F}) = \sum_{p \in \mathcal{I}} \lambda_{depth} E_{depth}(f_p) + \lambda_{norm} E_{norm}(f_p) + \lambda_{color} E_{color}(f_p) + E_{order}(f_p).$$

$E_{depth}$  and  $E_{norm}$  measure their deviations from the input depth and normal, respectively. The input normal is obtained by a local plane fitting per pixel.  $E_{color}$  measures how well the color model associated with the segment explains the pixel color. The energy definitions are standard and the details are in the supplementary document. An important point is that these terms are evaluated only **at the first non-empty layer** for each pixel.  $E_{order}$  assigns a large penalty when the depth ordering conflicts with the layer ordering. More specifically, let  $d(f_p^l)$  be the depth of a surface  $f_p^l$ , then  $E_{order}$  becomes  $10^6$  if  $d(f_p^l) > d(f_p^{l'}) + 0.03\text{m}$  for some  $l < l'$ , and otherwise 0.

**Smoothness term:** The smoothness energy is the sum of pairwise penalties  $E_{smooth}(f_p, f_q)$  over neighboring pixels in a 8-neighborhood system. The penalty is summed over layers:

$$E_{smooth}(f_p, f_q) = \omega \left( \sum_{l=1}^L \lambda_{s_1} S_1(f_p^l, f_q^l) + \sum_{l=1}^L \sum_{m=1}^L \lambda_{s_2} S_2(f_p^l, f_q^m) \right). \quad (1)$$

The innovative smoothness term is the key to successful layer decomposition. We want depth discontinuities to be

explained by layers with empty region in the foreground, instead of sharp geometric gap in a single layer. Therefore, the cost of adjacent ‘‘empty’’ and ‘‘non-empty’’ labels should be cheaper than the cost of two adjacent ‘‘non-empty’’ labels which indicate large depth change.  $S_1$  is set to a small constant 0.05 if  $(f_p^l = \phi, f_q^l \neq \phi)$  or  $(f_p^l \neq \phi, f_q^l = \phi)$ .  $S_1$  is 0 when  $f_p^l = f_q^l$ , that is, when both are  $\phi$  or the same surface segment. When they are different surface segments,  $S_1$  works as a truncated linear function of the depth difference plus a small constant 0.0001 penalizing the label change. The truncation is at 0.4m.  $\lambda_{s_1} = 10^4$ .

The second cost  $S_2$  is a standard anisotropic diffusion term:  $\exp(-\|c_p - c_q\|^2 / \beta^2)$ , where  $|c_p - c_q|$  is the color difference in the HSV space without the V channel.  $\beta$  is set to the root mean square of color differences over every pair of neighboring pixels in the image. This term is added instead of multiplied to the main smoothness term, because the purpose of this term is not to allow sharp depth discontinuities, and we want to keep the effects minimal. This cost is added only when  $f_p^l$  and  $f_q^m$  are the first (i.e., closest) non-empty surfaces at  $p$  and  $q$ , respectively. This prevents us from associating the color information to non-visible surfaces in backward layers.  $\omega$  encodes the strength of the pixel connection and is  $1/\sqrt{2}$  for diagonal pairs and 1 for horizontal/vertical pairs.  $\lambda_{s_2} = 500$ .

**MDL term:** This is a multi-layer extension of the MDL prior in *surface stereo* [2]. The same surface may appear in multiple layers, which should be penalized proportionally. We count the occurrence of surface IDs for each layer independently.  $\lambda_{MDL} = 20000$  is the penalty for each occurrence.

**Curvature term:** The term assigns an additional constant penalty (100) per pixel if a b-spline surface is used over a planar surface [2].

**Convex term** We want a surface to ‘‘extrapolate’’ well (e.g., a wall inpainting a geometry behind all objects). However, we do not want a surface with a complex shape or consists of many connected components. This term seeks to make each segment nearly convex and a single connected component. This is a complicated constraint to enforce, and we find that the following heuristic works well in practice (See Fig. 2). Suppose a pair of neighboring pixels  $p$  and  $q$  have different surfaces  $f_p^l$  and  $f_q^l$  in the same layer. Let us consider an image region (i.e., pixels) that were used to generate a surface  $f_p^l$  for  $p$ . This pair pays a standard smoothness penalty but will also pay a convex penalty  $\tau_{convex} = 5000$  if  $q$  is closer to this image region than  $p$ . Intuitively, this term penalizes the concave parts of an segment and also the appearance of isolated connected components.

**Parallax term:** We observed in our experiments that the above five terms define a ‘‘correct’’ model (the lowest en-

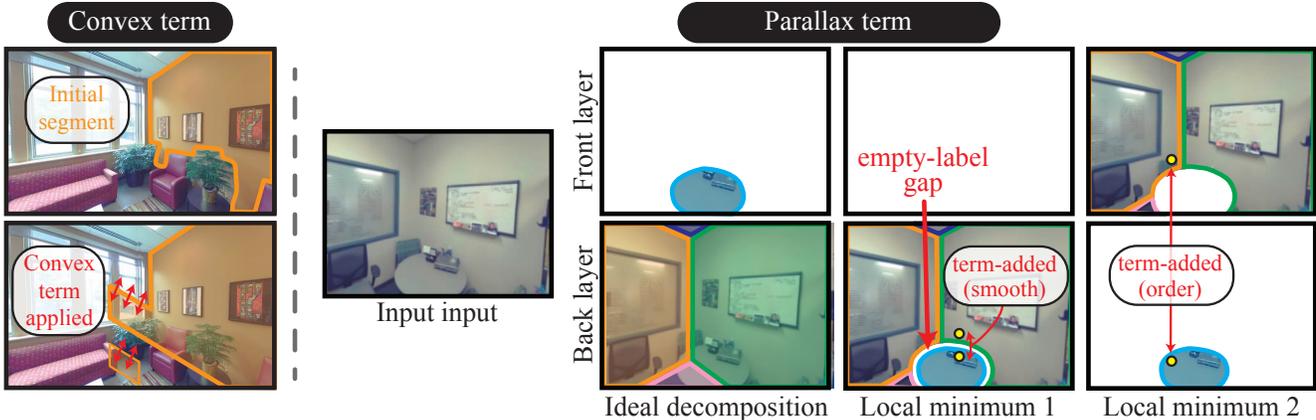


Figure 2. Left: A convex term prevents excessive extrapolation by penalizing a surface to have concave shapes and multiple connected components. Right: Without a parallax term, our optimization would often get trapped in local minima. The first typical local minima avoid high smoothness penalties by placing empty labels between discontinuous surfaces. The second typical local minima cannot reason proper layer order, because the ordering constraint is enforced only at the same pixel location. Parallax terms are added between pixels apart more than 1 pixel in the same layer (as a smoothness penalty in the first case) or over different layers (as a ordering penalty in the second case).

ergy is what we want), but introduce many deep local minima. Figure 2 illustrates two typical types. The problem is that the model enforces only local constraint (at best neighboring pixels) and lacks long-range interactions. The parallax terms are added to pairs of pixels that are more than one pixel apart in the same layer or in different layers.

To avoid densifying the connections, we identify pairs of pixels belonging to the *extended neighborhood* as follows. We look at the current layered depthmap from slightly different viewpoints by shifting the camera center by 0.1m towards the left, right, top, or bottom. For each new viewpoint, we collect pairs of pixels (in original image domain) that project to the same pixel (in the new view). For such pair of pixels, the smoothness cost  $E_{smooth}$  and the ordering penalty  $E_{order}$  are added (multiplied with a weight  $\lambda_{parallax} = 0.2$ ):

$$\sum_{(p,q) \in \mathcal{N}'} \lambda_{parallax} (E_{smooth}(f_p, f_q) + E_{order}(f_p, f_q)).$$

$\mathcal{N}'$  is the extended neighborhood system. Note that the ordering penalty  $E_{order}$  is extended to two pixels here (with abuse of notation).  $E_{order}(f_p, f_q)$  becomes  $10^6$  if the depth of  $f_p^l$  is larger than the depth of  $f_q^{l'}$  for some  $l < l'$ , and 0 otherwise.

## 5. Fusion space method

Occlusion-CRF poses a challenging optimization problem, as the label space is exponential to the number of layers. Although the number of layers is relatively small (4), the number of surface candidates may become 30, making their combinations at an order of hundreds of thousands.

Message passing algorithms such as TRW-S [10] store messages over the entire solution space, and is not feasible for our problem.<sup>1</sup> Fusion Move (FM) has been successfully used to solve such challenging problems [2, 3, 11] by repeatedly making a solution proposal and solving a binary problem. However, in each step, a variable sees a very limited solution space (two labels), and our experiments show that FM method is not effective for our problem either (See Sect. 6). Range Move (RM) allows multiple proposal labels in a single inference step [22, 23]. However, RM is not applicable to our problem, because it requires numeric labels with orders and cannot handle non-submodular energies.

We propose a new optimization approach, named *Fusion Space* (FS), to infer an occlusion-CRF model from an RGBD image. FS repeatedly proposes a restricted solution space for each variable, and solves a multi-labeling problem to update the solution by TRW-S [10]. The restricted space must contain the current solution to guarantee monotonic convergence. We have seven types of proposals and try them one by one after a random permutation. We repeat this process three times. The exception is the first two proposals in the first iteration. The first proposal must be the *surface adding proposal* to generate surface labels, as initially no surface labels exist. The second proposal must be a *background hull proposal*, which effectively recovers a background architectural structure. Now, we explain the details

<sup>1</sup>In addition, experiments on a small toy example revealed that TRW-S solver [10] suffers from very bad convergence in our problem setting. We suspect that this is due to the existence of many empty labels in the solution, whose energies are far from submodular (i.e., small penalty at the combination of empty and non-empty labels). Existing depth reconstruction algorithms handle the existence of empty labels effectively by message passing algorithms [4]. However, “empty” labels are rare and are treated as outliers with relatively large penalties in these methods.

of the proposals, that is, how to restrict or specify possible surface IDs for each pixel.

**Surface adding proposal:** This proposal adds new surface segments based on pixels that are not well explained by the current solution. We first identify a set of visible (first non-empty) pixels whose current depths or normals deviate from the inputs by more than 0.03m or  $30^\circ$  degrees, respectively. We iteratively use RANSAC to fit a plane to these pixels in each layer and remove the corresponding inliers from the set. For each connected region consisting of these pixels, whose size is less than 5% of an image, we also fit a b-spline surface, because curved surfaces are usually small objects. We obtain a set of planes and b-spline surfaces, and allow their usage in the foremost empty layer (the first layer if all layers are non-empty). We grow each surface as long as the pixels are inliers of the surface based on the same tolerance as above. We dilate the surface twice at the end.

**Background hull proposal:** A scene or a part of a scene often consists of a very smooth background and multiple objects in the front. For example, desks and chairs are in front of walls and objects are in front of a desk. This proposal places a set of smooth surface segments as a background, and pushes the remaining objects to frontal layers. We call the background surface a *background hull*, as the remaining geometry must be in front of it.

For each connected component surrounded by empty labels in one layer, we seek to form a background hull by a combination of at most three vertical and at most two horizontal surfaces with a tolerance of  $20^\circ$  degrees, where the up direction is the y-axis in our RGBD images. For every possible combination of surfaces, we form a background full (algorithmic details are in the supplementary document), and evaluate its goodness by “the number of inlier pixels based on the depth and normal deviations” (same tolerance as before) minus ten times “the number of pixels behind the hull by a tolerance of 0.03m”. We pick the background hull with the maximum score. Pixels on the layer of the connected component can take the current label or the one in the background hull. To allow objects to be pushed forward, the current label is also allowed at the same pixel in frontal layers.

**Surface refitting proposal:** As segments evolve over iterations, we need to refit surfaces based on new sets of pixels they explain. We use the same algorithm as in the surface adding proposal to fit a planar surface or a b-spline surface to generate new surface labels, and grow them. We update the model over the union of the current and the new surface labels.

**Layer swap proposal:** This proposal seeks to move around segments to different layers. For each segment in a layer, we allow the surface to move to the same pixels in different

layers. We dilate each surface twice to allow small shape changes. We exclude the background layer from this consideration, as this proposal is intended to handle isolated objects in a scene. The backward merging proposal (details in the supplementary material) achieves similar effects for the background layer.

We have three more proposal generation schemes, namely single surface expansion proposal, backward merging proposal, and structure expansion proposal. Their designs are similar to the ones mentioned above, and we refer readers to the supplementary material for their details.

## 6. Experimental results

We have evaluated the proposed approach both on NYU Depth Dataset V2 [18] and RGBD images that we acquired by ourselves. For the NYU dataset, we often observe large depth measurement errors as well as mis-alignment between RGB and depth data. While such data is suitable for evaluating the robustness of an algorithm, it makes it difficult to analyze the results. Therefore, we have used high-end laser range sensors (Faro 3D Focus) to acquire high-resolution RGBD images. The scanner produces a 40M pixel panoramic RGBD image, and we have produced standard RGBD images in the perspective projection. Our implementation is in C++, and a PC with a 3.6GHz CPU and 16GB of RAM has been used. The same set of parameters is used for all the examples (See the algorithm sections). Note that, although there are many parameters in our model, our algorithm is not sensitive to the choice of most parameters. The full experimental results are given in the supplementary document, and we here focus on 6 examples.

Figures 3 and 4 show an input image, a depth image, and the inferred multi-layer representation both as a 3D rendering and 2D images. The texture of inpainted/occluded geometry is set to a black color, while we will apply texture inpainting algorithm for rendering applications [1]. Architectural structures are estimated properly in the background layer for all the examples. The first smooth hull proposal recovers the structure in many cases, while the backward merging proposals adds refinement in some cases. Notice that the layer structure, often in the form of objects, tables, and walls/floor, are properly estimated in many examples.

We compared our algorithm against the standard Fusion Move (FM) algorithm. Our energy is not submodular and graph-cuts based technique cannot be used. Due to the extensive label space, standard message passing algorithms, such as loopy belief propagation or TRW-S, are not options for our problem either. To apply FM, we modify each of our proposal to limit the solution to a single new label (See the supplementary document). Figure 6 illustrates how the model energy and its lower-bound reported by TRW-S [10] change over iterations for FM and FS. We execute enough iterations for both FM and FS to observe the convergence.

	Our approach					Fusion move				
	Energy	Run. Time(min)	# surfaces	#layers	resolution	Energy	Run. Time(min)	# surfaces	#layers	resolution
<b>ours_1</b>	2.04E+06	11.3	20	4	200 x 150	4.09E+06	6.5	8	4	200 x 150
<b>ours_2</b>	2.65E+06	26.4	31	4	200 x 150	4.24E+06	12.3	24	4	200 x 150
<b>ours_3</b>	6.38E+06	36.9	18	4	200 x 150	9.10E+06	23.1	15	4	200 x 150
<b>ours_4</b>	3.01E+06	20.1	45	4	200 x 150	5.72E+06	9.6	15	4	200 x 150
<b>NYU_1</b>	4.48E+06	100.1	27	4	196 x 149	9.37E+06	14.0	23	4	196 x 149
<b>NYU_2</b>	3.43E+06	83.0	11	4	196 x 149	8.95E+06	7.9	8	4	196 x 149

Table 1. Comparison between Fusion Space and Fusion Move. Note that even though the running time of Fusion Move approach is shorter, the energy stays at a higher state.

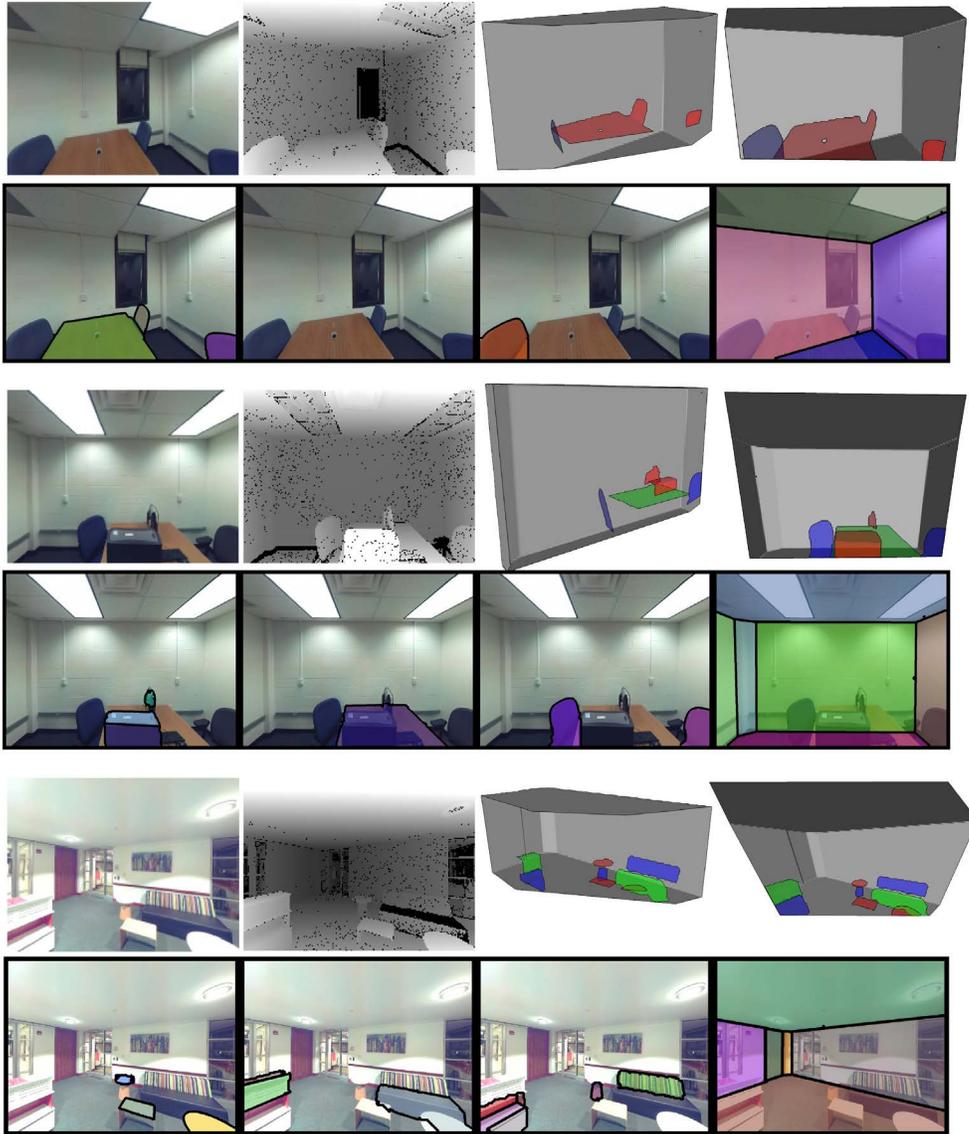


Figure 3. An input image, a depth image, and the inferred multi-layer representation as a 3D rendering is shown at the top row. The multi-layer representation as 2D images is shown at the bottom row. Each surface segment is represented as a translucent color mask on the image and its boundary is marked as black. From top to bottom, ours\_1, ours\_2, and ours\_3 datasets.

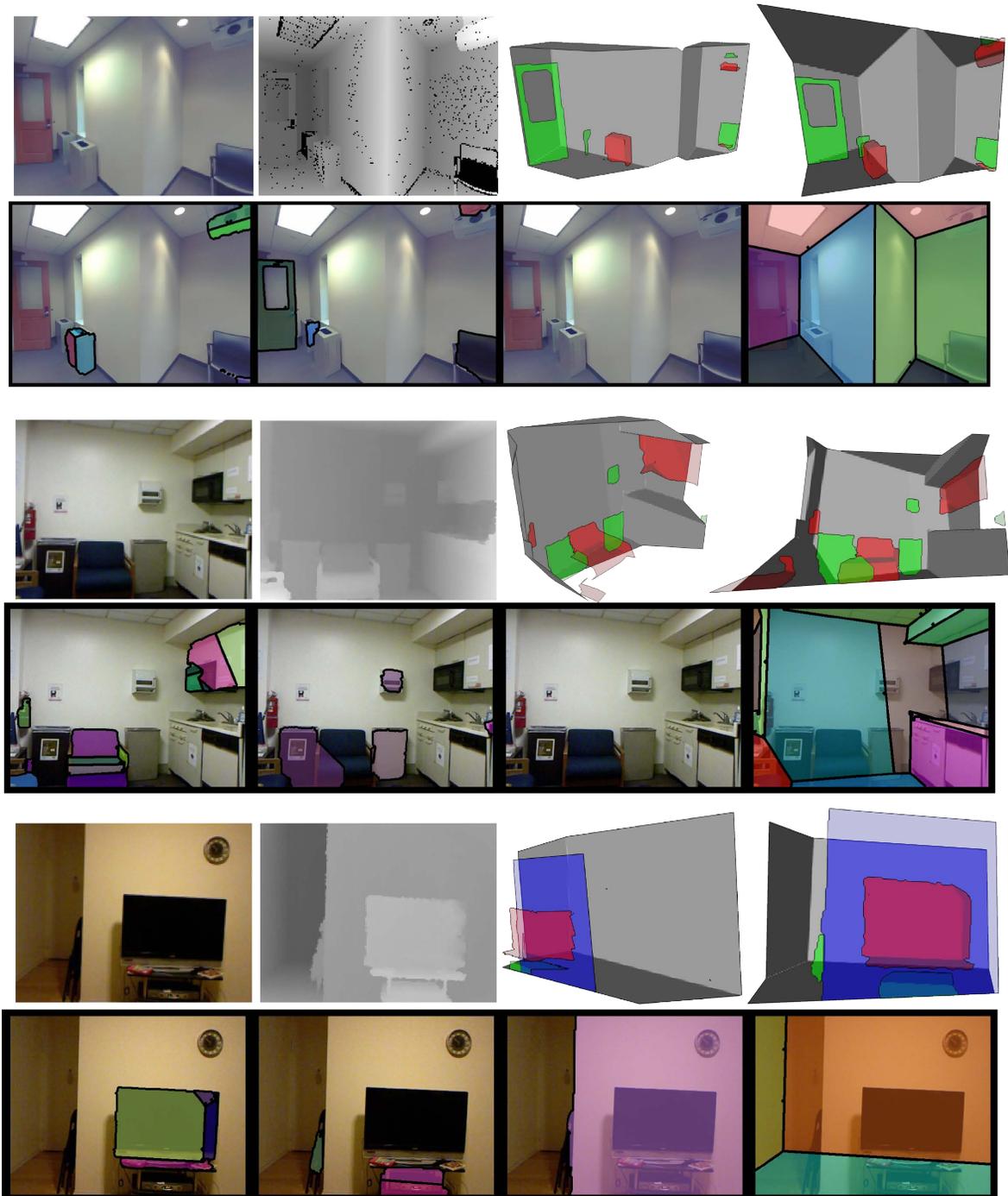


Figure 4. Results continued. From top to bottom ours\_4, NYU\_1, and NYU\_2 datasets.

Notice that the x-axis of the plot is the iterations not the running-time, and a single FS proposal is more expensive than a single FM proposal. However, as Table 1 shows, our method (FS) can achieve much lower energy state without significant slow-down. One key observation is that the gap from the lower-bound is very small (usually less than 10%)

in our method, despite the fact that we are solving more difficult problem per proposal. We have carefully designed the proposals and also set the parameters, so that each proposal can make big jumps yet the optimization is still efficient. Lastly, many proposals cannot make any progress. Even though the current model is in the solution space, TRW-S

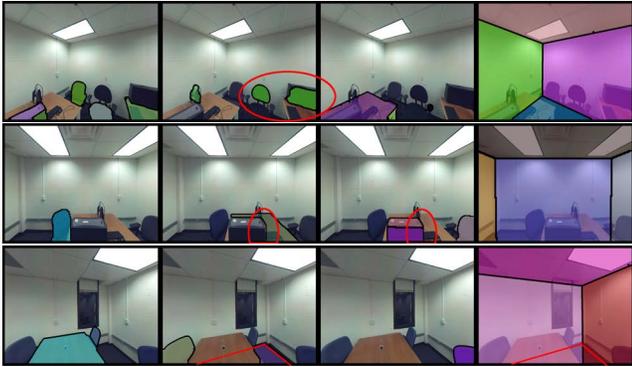


Figure 5. From top to bottom: 1) Without the convex term, a segment in the foreground has a complicated shape and multiple connected components arise; 2) Without the parallax term, a proper layer order cannot be enforced (e.g., an object behind a table); 3) Our variable encoding allows us to enforce anisotropic terms only at visible pixels. We ignored the visibility information and added the anisotropic terms to also invisible surface boundaries. The floor is not merged with the background, because the boundary between the floor and the wall run in the middle of the desk chair, leading to large penalties.

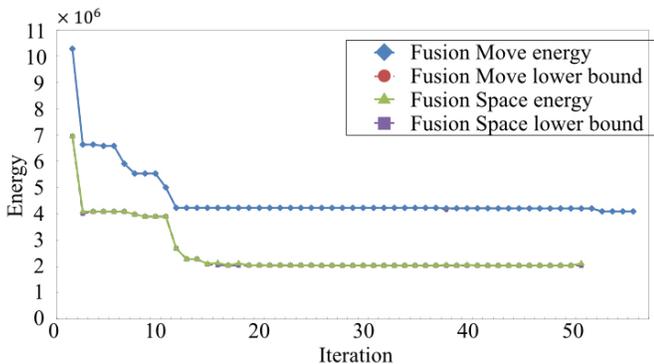


Figure 6. A plot of energies over iterations for ours\_1 dataset. Four curves show the Fusion space energy, the Fusion space lower bound, the Fusion Move energy, and the Fusion Move lower bound, respectively. Our approach achieves much lower energy. The lower bound is almost identical to the energy itself in both cases, proving the effectiveness of our Fusion Space method, in particular, solution-space generation schemes.

may return a state with a higher energy. In this case, we simply keep the previous solution.

Figure 5 evaluates the effectiveness of various terms in our algorithm, in particular, convex, parallax, and anisotropic terms. Note that our results with all the terms are shown in Figs. 3 and 4.

Lastly, we have experimented two Graphics applications based on our model. The first application is image-based rendering, in particular, parallax rendering. A standard depthmap-based rendering either causes holes or texture-stretches at occlusions. Our model is free from these two rendering artifacts, and can properly render initially oc-

cluded surfaces whose texture have been inpainted by a standard texture-inpainting algorithm [1]. The second application is the automatic removal of objects in an image, which is as simple as just dropping a few layers from the front in displaying our layered depthmap model.

## 7. Conclusions and Future Directions

The paper presents a novel Occlusion-CRF model, which addresses the challenging problem of perceiving the hidden or occluded geometry. The model represents a scene as multiple layers of segmented depthmaps, which can naturally explain occlusions and allow inpainting of invisible surfaces. A variant of a fusion-move algorithm is used to manage a large label space that is exponential to the number of layers. To the best of our knowledge, this is the first attempt to propose a generic geometric representation that fundamentally addresses the occlusion problem.

While our algorithm successfully reconstructs relatively large foreground objects in frontal layers, our current algorithm fails to deal with complex clutters in some cases as it relies on depth values to infer surface segments and layers. The issue is exacerbated by the fact that depth sensors have much lower resolution than the RGB sensors. An interesting future work is to more effectively utilize images via image segmentation or recognition techniques. Speed-up is also an important future work. An interesting direction is to solve multiple fusion-space moves simultaneously, then merge multiple solutions by yet another TRW-S inference. Another limitation of our current algorithm is that we use a fixed number of layers which might be redundant or insufficient to represent the scene. A future direction is to incorporate the automatic inference of the number of layers into our optimization framework.

We hope that this paper will stimulate a new line of research in a quest of matching the capabilities of the human vision system. The code and the datasets of the project will be distributed for the community.

## 8. Acknowledgement

This research is partially supported by National Science Foundation under grant IIS 1540012 and partially supported by Google Faculty Award.

## References

- [1] C. Barnes, E. Shechtman, A. Finkelstein, and D. Goldman. Patchmatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics-TOG*, 28(3):24, 2009. 5, 8
- [2] M. Bleyer, C. Rother, and P. Kohli. Surface stereo with soft segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1570–1577. IEEE, 2010. 1, 2, 3, 4
- [3] M. Bleyer, C. Rother, P. Kohli, D. Scharstein, and S. Sinha. Object stereojoint stereo matching and object segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3081–3088. IEEE, 2011. 2, 4
- [4] N. Campbell, G. Vogiatzis, C. Hernández, and R. Cipolla. Using multiple hypotheses to improve depth-maps for multi-view stereo. *Computer Vision-ECCV 2008*, pages 766–779, 2008. 4
- [5] A. Delong and Y. Boykov. Globally optimal segmentation of multi-region objects. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 285–292. IEEE, 2009. 2
- [6] R. Guo and D. Hoiem. Beyond the line of sight: Labeling the underlying surfaces. In *ECCV*, pages 761–774, 2012. 2
- [7] R. Guo, C. Zou, and D. Hoiem. Predicting complete 3d models of indoor scenes. *CoRR*, abs/1504.02437, 2015. 1, 2
- [8] A. Gupta, A. A. Efros, and M. Hebert. Blocks world revisited: Image understanding using qualitative geometry and mechanics. In *Computer Vision-ECCV 2010*, pages 482–496. Springer, 2010. 1
- [9] B.-s. Kim, P. Kohli, and S. Savarese. 3d scene understanding by voxel-crf. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1425–1432. IEEE, 2013. 2
- [10] V. Kolmogorov. Convergent tree-reweighted message passing for energy minimization. *PAMI*, 28(10):1568 – 1583, 2006. 1, 4, 5
- [11] V. Lempitsky, C. Rother, S. Roth, and A. Blake. Fusion moves for markov random field optimization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32(8):1392–1405, 2010. 4
- [12] D. Lin, S. Fidler, and R. Urtasun. Holistic scene understanding for 3d object detection with rgbd cameras. In *ICCV*, pages 1417–1424, 2013. 2
- [13] C. Olsson and Y. Boykov. Curvature-based regularization for surface approximation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1576–1583. IEEE, 2012. 2
- [14] C. Olsson, J. Ulén, and Y. Boykov. In defense of 3d-label stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1730–1737, 2013. 2
- [15] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(7):629–639, 1990. 1
- [16] A. G. Schwing, S. Fidler, M. Pollefeys, and R. Urtasun. Box in the box: Joint 3d layout and object reasoning from single images. In *ICCV*, pages 353–360, 2013. 2
- [17] J. Shade, S. Gortler, L.-w. He, and R. Szeliski. Layered depth images. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, pages 231–242. ACM, 1998. 2
- [18] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *Computer Vision-ECCV 2012*, pages 746–760. Springer, 2012. 5
- [19] N. Silberman, L. Shapira, R. Gal, and P. Kohli. A contour completion model for augmenting surface reconstructions. In *ECCV*, pages 488–503, 2014. 2
- [20] J. Sun, Y. Li, S. B. Kang, and H.-Y. Shum. Symmetric stereo matching for occlusion handling. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 399–406. IEEE, 2005. 1
- [21] J. Tighe, M. Niethammer, and S. Lazebnik. Scene parsing with object instances and occlusion ordering. In *CVPR*, pages 3748–3755, 2014. 2
- [22] O. Veksler. Graph cut based optimization for mrfs with truncated convex priors. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007. 4
- [23] O. Veksler. Multi-label moves for mrfs with truncated convex priors. In *Energy minimization methods in computer vision and pattern recognition*, pages 1–13. Springer, 2009. 4
- [24] G. Vogiatzis, P. H. Torr, and R. Cipolla. Multi-view stereo via volumetric graph-cuts. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 391–398. IEEE, 2005. 1
- [25] J. Y. Wang and E. H. Adelson. Representing moving images with layers. *Image Processing, IEEE Transactions on*, 3(5):625–638, 1994. 2
- [26] Y. Wexler, E. Shechtman, and M. Irani. Space-time completion of video. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(3):463–476, 2007. 2
- [27] J. Zhang, C. Kan, A. G. Schwing, and R. Urtasun. Estimating the 3d layout of indoor scenes and its clutter from depth sensors. In *ICCV*, pages 1273–1280, 2013. 2
- [28] Y. Zhang, S. Song, P. Tan, and J. Xiao. Panocontext: A whole-room 3d context model for panoramic scene understanding. In *Computer Vision-ECCV 2014*, pages 668–686. Springer, 2014. 1, 2
- [29] K. C. Zheng, S. B. Kang, M. F. Cohen, and R. Szeliski. Layered depth panoramas. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007. 2