EUROGRAPHICS 2015 / O. Sorkine-Hornung and M. Wimmer
(Guest Editors)

Volume 34 (2015), Number 2

Hallucinating Stereoscopy from a Single Image

Qiong Zeng, Wenzheng Chen, Huan Wang, Changhe Tu, Daniel Cohen-Or[†], Dani Lischinski[‡], Baoquan Chen

Shandong University, China [†]Tel-Aviv University, Israel [‡]The Hebrew University of Jerusalem, Israel

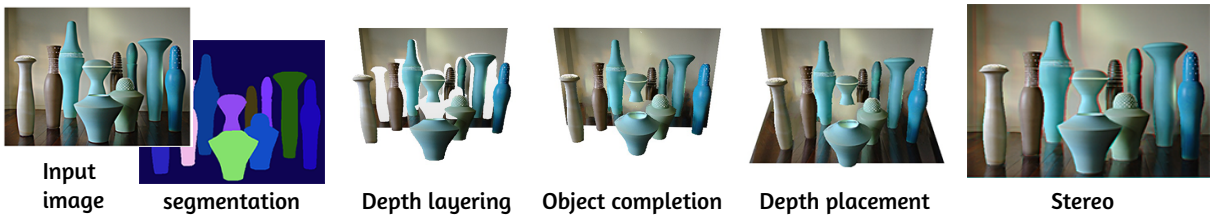


Figure 1: *Given a segmented image as input we produce a stereo pair (or a motion parallax animation) by hallucinating plausible 3D geometry for the scene. First, segments are depth-sorted using simple depth and occlusion cues. Next, the geometry and texture of each object is completed using symmetry and convexity priors. Finally, we infer a depth placement for each object. We urge the reader to view the companion video before reading the paper, and to examine our anaglyphs in full size using red/cyan glasses. All of the results in this paper are available in the supplementary material.*

Abstract

We introduce a novel method for enabling stereoscopic viewing of a scene from a single pre-segmented image. Rather than attempting full 3D reconstruction or accurate depth map recovery, we hallucinate *a rough approximation of the scene's 3D model using a number of simple depth and occlusion cues and shape priors. We begin by depth-sorting the segments, each of which is assumed to represent a separate object in the scene, resulting in a collection of depth layers. The shapes and textures of the partially occluded segments are then completed using symmetry and convexity priors. Next, each completed segment is converted to a union of generalized cylinders yielding a rough 3D model for each object. Finally, the object depths are refined using an iterative ground fitting process. The hallucinated 3D model of the scene may then be used to generate a stereoscopic image pair, or to produce images from novel viewpoints within a small neighborhood of the original view. Despite the simplicity of our approach, we show that it compares favorably with state-of-the-art depth ordering methods. A user study was conducted showing that our method produces more convincing stereoscopic images than existing semi-interactive and automatic single image depth recovery methods.*

Categories and Subject Descriptors (according to ACM CCS): I.3.7 [Computer Graphics]: Picture/Image Generation—viewing algorithms; I.4.8 [Image Processing and Computer Vision]: Scene Analysis—depth cues;

1. Introduction

Inferring a 3D model of a scene or an object from a set of 2D images is one of the long-standing grand challenges of computer vision. The problem is even more challenging (in fact, it is ill-posed) when the input consists of only a single photograph. However, there are applications for which a full 3D model is unnecessary, and even a rough approximation of the scene's depth map may suffice.

One such application is stereoscopic viewing of a scene. While stereo vision greatly enhances the 3D experience and perception of a scene by human observers, there is evidence that accurate disparities between the left and the right eye images are not essential for evoking a compelling sensation of depth; for example, people often prefer somewhat exaggerated disparities [IdH98]. Motivated by this observation, in this work we aim at *hallucinating* a rough 3D scene model from a single segmented image, which can then be used to generate a convincing stereoscopic pair of images, or even a continuous motion parallax animation.

Our work is related to much previous research on image-based modeling and 3D scene reconstruction, which we discuss in more detail in the next section. Most of these previous works require multiple images and/or considerable user assistance. Although several automatic single-image methods have recently been proposed (e.g., [HSEH07, SSN09, KLK14]), these methods require a large relevant training set, seem to mostly work well on scenes with deep perspective, such as natural or urban outdoor scenes, and do not handle nearby foreground objects well. However, the most compelling stereo effects are typically obtained in scenes that feature prominent nearby foreground objects, and often have much smaller overall depth range, such as "still-life" or "product shot" type scenes, as shown in Figure 1. Our method targets such scenes, requiring neither training, nor large image databases with additional information.

The problem we are faced with is a challenging one. Although a small sideways motion of the viewpoint does not typically result in a drastic change in the image, in the presence of mutually occluding objects relatively close to the observer, even a small change in viewpoint causes noticeable parallax, which should be reproduced in a convincing manner. It is thus necessary to come up with a plausible depth ordering, and to hallucinate plausible geometry for each foreground object. Furthermore, parallax typically disoccludes portions of the scene, which necessitates to complete both the geometry and the texture in the exposed regions.

Accurate segmentation is crucial for the ability to produce a convincing stereo effect. Segmentation has been an active area of research since the dawn of computer vision, but we have yet to see a foolproof automatic segmentation or foreground extraction method. Indeed, we observed that it is often the failure to correctly resolve foreground object contours that is responsible for the lack of convincing stereopsis in depth maps produced by existing fully automatic 3D reconstruction methods. Thus, in this work we opt to start with a manually segmented image, where each object that we wish to set apart from the background is represented as a single segment. From this point on, however, our method proceeds fully automatically.

Starting from a segmented image, our method utilizes a number of simple depth and occlusion cues in order to assign each segment into one of several distinct depth-ordered layers. This is achieved by formulating and solving a multi-labeling problem using graph cuts. Next, we use heuristics based on the assumption that objects tend to be *convex* and *symmetric* (or comprise convex and symmetric parts) in order to complete partially occluded segments and augment them with an approximate 3D geometry. Specifically, for each segment, we look for a vertical axis that maximizes the convexity of the shape obtained by reflection about this axis. Having found such an axis we apply texture completion where needed and "inflate" the segment into a union of 3D generalized cylinders with texture (somewhat resembling the inflation mechanism employed by Igarashi *et al.* [IMT99]).

Finally, it is necessary to assign specific depths to the different objects, and to do so in a consistent manner. Here, similarly to previous work (e.g., [HSEH07, RT09]) we assume that most objects in the scene are standing on the ground, or on some other supporting plane (e.g., table top). An iterative optimization process is employed to fit a supporting plane to a set of estimated object-ground contact points and to find consistent object positions on this plane, yielding the final hallucinated scene model.

Despite the simplicity of our method, we have been able to generate a variety of surprisingly compelling stereo pairs and parallax animations, which manage to convey a palpable sense of depth and 3D, as shown in Section 7 and the supplementary video and materials. We report a favorable quantitative comparison with two state-of-the-art depth ordering methods. We have also carried out a user study whose results show that our method produces more convincing stereoscopic images than existing semi-interactive and automatic single-image depth recovery methods.

2. Related Work

Image-based rendering: Many methods have been proposed for generating novel views of an object or a scene without explicitly constructing a 3D model (e.g., [CW93, MB95, LH96] and numerous follow-ups, see [Oli02]). Unlike the current work, most image-based rendering approaches utilize multiple images with or without explicit correspondences among them.

Image-based 3D modeling: Much work has also been done on methods that create high quality textured 3D models from photographs. An early example is the pioneering Façade system [DTM96] for creating an architectural model, typically from multiple photographs of a building, with many follow ups in research and commercial products [Oli02].

Some methods have been proposed specifically for novel view synthesis from a single still image. The "Tour Into the Picture" system [HAA97] texture-maps the image onto a simple user-drawn mesh, enabling, in some cases, a compelling 3D navigation experience. Subsequent works (e.g., [Kan98, OCDD01, ZDPSS01]) extend this idea by providing more sophisticated user-guided 3D modeling techniques. In all these techniques, the user actively participates in the 3D modeling process. The results can be very impressive, but the modeling times could be on the order of hours [OCDD01].

Recent examples of image-based modeling from a single image include Töppe *et al.* [TOCR11], who use Cheeger sets to fit a 3D model to a silhouette based on a few user scribbles. Similarly, 3-Sweep [CZS*13] offers an intuitive UI for

fitting generalized cylinders to objects in a single image. In both of the above methods, the user must explicitly model every single object, and they are not designed for handling multiple objects with occlusions. In contrast, our approach automatically hallucinates a simple 3D geometry for each image segment, accounting for possible occlusions between them, and also assigns each objects with a depth placement in the scene.

Depth ordering: Several methods have been proposed in the computer vision literature that attempt to infer the relative depth ordering of objects in a single image. For example, Amer *et al.* [ART10] estimate a globally consistent 2.1D sketch from T-junctions using constrained quadratic optimization. Palou and Salembier [PS13] infer the depth order from T-junctions and highly convex contours using a binary space partitioning tree. Jia *et al.* [JGCC12] use supervised learning to combine features based on boundary and junction characteristics in order to infer depth ordering. They employ an MRF formulation to encourage a globally consistent ordering, and quantitatively show improved performance over the state-of-the-art. We also rely on T-junctions as local occlusion cues, but propose a novel MRF formulation that employs a data term, based on each segment's estimated ground contact. This enables us to estimate a global ordering even between segments which would belong to different connected components in Jia's MRF graph. Section 7 presents a quantitative comparison between our depth ordering and those of [JGCC12, PS13].

Automatic 3D from a single image: A fully automatic approach is proposed by Hoiem *et al.* [HEH05], which uses machine learning techniques to construct a simple "pop-up" 3D model, consisting of a ground and a number of vertical planes. Khan *et al.* [KRFB06] reconstruct an approximate depth map for image-based material editing of a single object. The diorama work by Assa and Wolf [AW07] utilizes various depth cues to automatically model the scene as piecewise-smooth depth map surface, with slits that capture depth discontinuities. Hoiem *et al.* [HSEH07] use supervised machine learning to recover occlusion boundaries and depth ordering of free-standing structures in a single still image. Saxena *et al.* [SSN09] also use supervised machine learning to infer a set of plane parameters for each small homogeneous patch in an image, yielding a textured polygonal 3D model. Liu *et al.* [LGK10] estimate depth after first predicting per-pixel semantic labels.

Most of the aforementioned single-image techniques are designed for outdoor scenes, which typically exhibit significant depth and perspective, as well as significant variations in the color, texture, and orientation of the surfaces in the scene. In contrast, in this work we aim at scenes with more local arrangements of objects, often without a deep perspective, but with prominent foreground objects, and often with similarities in color and texture. For example, this is often the case with "still-life" or "product shot" photographs, such as the one shown in Figure 1. Our approach does not involve machine learning, and thus does not require large relevant training datasets, which may be difficult to construct for such scenes.

Karsch *et al.* [KLK14] automatically estimate depth maps for still images and videos using non-parametric depth transfer, using an RGBD database of videos with Kinect-captured depth for training. They search the database for candidate images that are similar in appearance to the input image, and align them to the input using SIFT Flow [LYT11]. An optimization procedure is then used to interpolate and smooth the warped candidate depth values. The effectiveness of their approach greatly depends on the presence of good matching candidates in the RGBD database. In contrast, our approach does not require any RGBD data. We include this method in our user study in Section 7.

User-assisted 3D recovery: Russell and Torralba [RT09] describe a method for 3D scene recovery from user annotations, where the user draws the outline and provides a semantic label for each object in the image. Object relationships are then derived by integrating cues from object labels across a database of labeled images. Similarly to our approach, they assume that the scene is comprised of free-standing objects on a ground plane, but recover only a planar representation for each object. For good results, the horizon must also be drawn by the user. They demonstrate some convincing results, but it remains unclear how well their method would perform on scenes with objects that are not well represented in the labeled database. We include this method in our user study in Section 7.

Stereo from video sequences: With the increasing popularity of stereoscopic cinema, several systems were proposed for user-assisted conversion of monocular video sequences into stereoscopic ones. For example, [GWCO09] and [WLF*11] describe systems where user scribbles in a few frames are propagated to define a dense disparity map for the entire sequence. Ward *et al.* [WKB11] and Gao *et al.* [GLYG12] describe two other systems that combine user input with Structure from Motion techniques in order to augment a video sequence with depth. Yu *et al.* [YLR*11] extract a 3D representation from videos of a static scene with moving objects, using assumptions similar to those of our approach.

Our work bears some similarity to the recent work by Liu *et al.* [LMY*13] on "stereoscopizing" cel animations. They infer approximate ordering of layers by exploiting T-junction cues in individual frames and then set up a graphcut problem on the graph of relations between pairs of layers, the solution to which attempts to maintain the temporal consistency of the ordering throughout the animation. In contrast, our approach operates on a more challenging case: a single natural image, where occlusion cues are less clear, and without being able to leverage temporal coherence. Instead, our approach uses a different MRF formulation that attempts to

maximize global consistency between T-junction cues and local depth cues. Our approach also generates a more diverse geometric model consisting of generalized cylinders and planes, in contrast to fronto-parallel planar layers.

3. Overview

The goal of our method is to hallucinate a rough approximation of 3D geometry given a single still image of a scene along with its segmentation. We assume that the segmentation has each geometrically and semantically distinct object in a separate single segment.

The geometry hallucination process starts by using simple depth and occlusion cues to organize the segments in a collection of depth-ordered layers. This is achieved by solving an optimization problem using graph cuts, where each segment's position in the image provides the data term, and T-junctions between adjacent segments provide the pairwise smoothness term. This optimization provides an initial depth ordering, detecting partial occlusions, and distinguishing between object silhouettes and occlusion contours.

Armed with the above information, we proceed to complete both the 2D shape and the texture of each segment using convexity and symmetry priors. For each segment, we seek a vertical axis, such that reflecting the segment about this axis maximizes the convexity of the resulting 2D shape. Having completed the 2D shape, we inpaint any missing texture. Next, we "inflate" each completed 2D segment into 3D, by representing it as a union of generalized cylinders.

At this point we have a hallucinated textured 3D model for each object, but in order to produce a plausible stereo effect it is necessary to assign specific depths to the different objects, and to do so in a consistent manner. Here we take advantage of the fact that most objects in the scene typically rest on the ground, or on some other supporting plane (e.g., table top). An iterative optimization process is employed to fit the supporting plane to a set of estimated object-ground contact points and to find consistent object positions on this plane, yielding the final hallucinated scene model.

The process outlined above is depicted in Figure 1, and its three main phases (depth layering, object completion, and ground fitting) are described in the next sections.

4. Depth Layering

Given an image and its accompanying segmentation, our goal is now to infer a partial depth ordering of the segments (distinct objects) in the image. More formally, let $s_1,\ldots,s_N$ denote the N segments in an image, and let $1,\ldots,L$ denote L *depth layers* ($N \leq L$), ordered according to their distance from the camera. Assigning a segment s_i to layer k means that the corresponding object may be occluded by objects in layers $1,\ldots,k-1$ and may in turn occlude objects in layers $k+1,\ldots,L$. Multiple segments may be assigned to the same

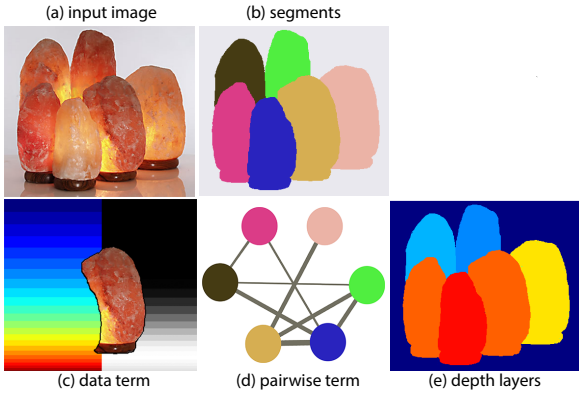


Figure 2: *Depth layering: given an input image (a) and its segmentation (b), we assign each segment to one of a set of depth ordered layers. (c) The set of 24 layers is visualized using colors, while the gray levels on the right show the likelihood of the superimposed segment to belong to each layer. (d) The adjacency graph where the thickness of each edge corresponds to the magnitude of the occlusion cue $|T_{ij}-T_{ji}|$. (e) The resulting layer assignment.*

layer, meaning that we have no evidence that one of the corresponding objects is closer or farther from the camera than the other(s). Thus, the process of establishing a partial depth ordering is formulated as a labeling problem, which we solve by minimizing an energy function using multi-label graph cuts [BVZ01].

Specifically, we seek a layer assignment ℓ that minimizes the following energy function:

$$E(\ell) = \sum_{i=1}^{N} D_i(\ell(s_i)) + \lambda \sum_{(i,j)} S_{ij}(\ell(s_i),\ell(s_j)). \qquad (1)$$

The function consists of a sum of unary data terms D_i, and pairwise smoothness terms S_{ij}, balanced by the parameter λ (we set $\lambda = 1$).

Data term. In scenes with free-standing objects, the bottom part of the object often corresponds to the object's point of contact with the ground, thus providing a simple, yet effective depth cue. Therefore, our data term D_i uses the vertical position of the segment's bottom in order to estimate the likelihood of it belonging to each of the depth layers. Specifically, given the target number of layers L, we split the image into L horizontal bands $b_1,\ldots,b_L$, and estimate the likelihood of segment s_i to belong to layer k using a Gaussian centered at the band that contains the bottom of the segment $y_{min}(s_i)$:

$$D_i(k) = 1 - \exp\left(-\frac{(y_k - y_{min}(s_i))^2}{\sigma^2}\right) \qquad (2)$$

Here y_k is the vertical coordinate at the center of band b_k. In our implementation all of the vertical coordinates y are normalized to $[0,1]$, and $\sigma = 0.2$. We found that using this

Gaussian falloff is more robust than simply assigning all of the likelihood to the band containing the bottom. Furthermore, note that every layer is assigned a non-zero likelihood, and not only those layers corresponding to bands containing a portion of the segment.

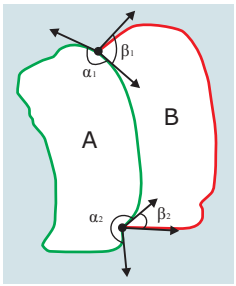
For our application it is important to achieve good depth resolution and therefore we use a large number of layers, $L = \max(20, 4N)$, with thinner bands at the bottom and thicker ones towards the top of the image (the thicknesses of the layers form an arithmetic series, whose sum is the height of the image), in order to ensure finer depth resolution in the frontmost parts of the scene. The data term is illustrated in Figure 2(c). This data term is one of the differences between our approach and previous MRF-based approaches, such as [JGCC12, LMY*13].

The smoothness term S_{ij} accounts for occlusion cues between each pair of adjacent segments (s_i, s_j). Penalty is incurred if two segments are assigned to the same layer whenever there is an indication that one of them is occluding the other. Also, there is a penalty in the case of a contradiction between the assigned layers and the occlusion cue, i.e., if segment j is deemed to occlude segment i, but is assigned to a deeper layer (with a higher number):

$$S_{ij}(k,l) = \begin{cases} \gamma(1 - e^{-(T_{ij}-T_{ji})^2}) & \text{if } (k = l) \\ \delta(1 - e^{-(T_{ij}-T_{ji})^2}) & \text{if } (k-l)(T_{ij}-T_{ji}) > 0 \\ 0 & \text{otherwise,} \end{cases} \quad (3)$$

where $\gamma = 0.1$ and $\delta = 1.2$ in our current implementation.

Here T_{ij} quantifies the cues that segment s_i occludes s_j, and conversely for T_{ji}. In our current implementation we employ only one kind of occlusion cue: the relationship between the angles around any T-junctions that might exist between the two segments. Since our method is given a segmentation of the image as input, T-junctions are easily identified as the points where three differently segments meet. We sample a number of points on each of the three edges approaching the junction, and fit a line to each edge. We then compute the angles between these three lines. As illustrated in the inset figure, considering a T-junction involving two segments, A and B, larger angles α_i provide a stronger cue for A occluding B. Since there may be several such junctions between a pair of adjacent segments, we set T_{AB} to the average of the angles α_i, and T_{BA} to the average of the angles β_i.



Despite the simplicity of the depth and occlusion cues utilized above, we found that minimizing eq. (1) succeeded in recovering a feasible depth ordering of the segments in most of our examples. The second column in Figure 8 shows several depth layering results produced by this method.

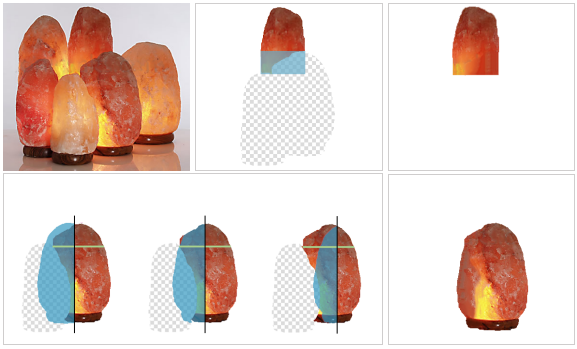


Figure 3: *Segment completion. Top row: input image, segment with multi-sided occlusion, and its completion result. Bottom row: segment with single-sided occlusion, and its completion result. We search for a vertical reflection axis that produces the most convex shape (the middle one).*

5. Object Completion and Modeling

Having estimated the depth ordering of the segments, we can now determine which parts of each segment's boundary correspond to occlusion contours, as opposed to object silhouettes. We use this information to hallucinate the occluded parts of each segment, based on the assumption that objects tend to be convex and symmetrical.

We distinguish between two cases: (i) one-sided occlusion, where occlusion contours exist only on the right half or only on the left half of the segment's boundary, and (ii) multi-sided occlusion. The former case is handled by searching for a vertical axis through the segment, such that when the segment is reflected about this axis, the union of the reflected shape with the original segment yields a shape that is as convex as possible. This idea is illustrated in the second row of Figure 3. The convexity of each candidate shape is measured using the convexity rank of Asafi *et al.* [AGCO13]. For multi-sided occlusion, we complete the missing part by overlaying the segment with a rectangle containing the occluded contours of the segment (top row of Figure 3).

After completing the shapes of the partially occluded segments, we use content-aware image completion [BSFG09] to complete the texture inside the occluded parts. Texture completion is also used to fill the holes in the remaining background layer.

Having a completed 2D shape for each segment, the next step is to produce a 3D model, composed from a union of generalized cylinders. This is done by sweeping the 2D shape vertically (scanline order): the midpoint of each horizontal segment of width w inside the shape is assumed to be located on the axis of a skewed generalized cylinder whose

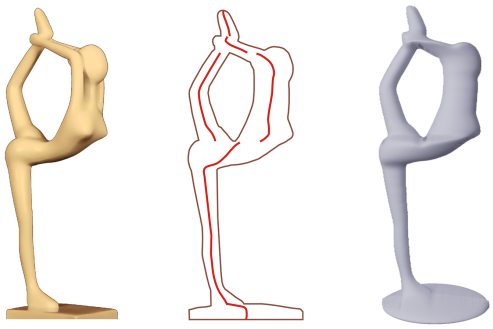


Figure 4: *A segment of a statue is converted to a 3D model using a collection of skewed generalized cylinders. The axes of the different cylinders are shown in the middle image.*

Figure 5: *Two anaglyph images of a teddy bear. Left: using a collection of generalized cylinders. Right: using a billboard. Note the differences in the perceived shape of the bear's head. Please magnify and use red/cyan glasses for viewing these images. This and other examples are included in the supplementary material.*

radius at this point is set to $w/2$. Given the collection of axial points and the radius at each point, we generate the 3D skewed generalized cylinders and apply Laplacian smoothing [Tau95] to the resulting surface. We prevent the smoothing from modifying the positions of the contour points. This process is illustrated in Figure 4.

In our experiments we found that although even using flat (billboard) geometry to represent each object generally yields satisfactory results, using a 3D surface of revolution, as we do, creates a slightly stronger sensation of 3D. The difference is most easily noticed for large foreground objects with convex parts, such as the teddy bear shown in Figure 5. However, large flat objects in the foreground may appear less correct as a result.

6. Depth Placement and Ground Fitting

Having estimated the depth ordering of the objects and the 3D shape of each object, our goal is now to establish the actual position of each object in the scene. We begin with an initial placement where the position of each object is determined by the depth layer that it belongs to and by the positions of the objects in the layers in front of it. We start with the frontmost layer and align the fronts of the objects in it to the same depth (arbitrarily set to zero). Proceeding from front to back, we then place the objects in the next layer at the same depth, chosen as close as possible behind the center of the largest object in the previous layer, but without letting their 3D models interpenetrate, as shown in the middle column of Figure 6.

In many cases, the objects in the scene may be assumed to rest on a common planar ground surface. Attempting to recover this ground plane and ensuring that the objects' placement is consistent with the recovered ground geometry can greatly improve the relative positions of the objects in the scene, and yield a more realistic looking stereo effect.

We perform ground plane fitting in an iterative manner. Starting with the initial object placement, determined as described earlier, we obtain a set of estimated contact points between the objects and the ground. The lowest point of each segment is designated as a contact point, unless the contour is classified as occluded at that point. We recover the normal $\mathbf{n}$ to the ground plane by using least squares fitting:

$$\mathbf{n} = \arg\min_{\mathbf{n}} \sum_{i=1}^{k} (\mathbf{n} \cdot (x_i - a))^2 + \lambda n_x^2, \tag{4}$$

where $x_1, \ldots, x_k$ are the estimated contact points, while a is the average of the contact points. Eq. (4) is a standard technique for plane fitting [SE03], to which we have added a regularization term λn_x^2. By minimizing the horizontal n_x component of the normal we obtain a level ground plane and avoid implausible slopes that might otherwise arise due to inaccurately estimated contact points. The ground plane equation is then defined that has the normal $\mathbf{n}$ and passes through the point a.

Having estimated the plane, we compute a displacement d_i for the depth component (z-coordinate) of each contact point, so as to make the points comply with the ground plane equation. This is done by optimizing

$$\{d_i\} = \arg\min_{\{d_i\}} \sum_{i=1}^{k} (\mathbf{n} \cdot (x_i - a) + n_z d_i)^2, \tag{5}$$

where n_z is the z-component of the unit normal $\mathbf{n}$ to the ground plane. Although each variable d_i appears in a separate term, we must nevertheless optimize them simultaneously, because our optimization is subject to the ordering constraints imposed by the order of the depth layers. The above two-step procedure is iterated until convergence.

We also have to estimate the depth of those objects which do not have a contact point with the ground. This is done by averaging the depths of the adjacent objects in layers in front of and behind the object in question and using the resulting average depth to position the object between them.

Finally we create a vertical background plane facing the camera, and position it immediately behind the farthest object in the scene. All of the image pixels that belong to none of the segmented objects are texture mapped either onto the ground plane or onto the vertical background plane based on their position with respect to the intersection line between these two planes.

Figure 6 demonstrates the significant effect that ground fitting has on the depth placement of objects in one of our test scenes. In general, we found that the difference in the resulting stereo images can be quite significant. With ground fitting the depths typically appear more correct, and objects appear better attached to the ground, rather than floating above it. These differences are demonstrated in Figure 7.

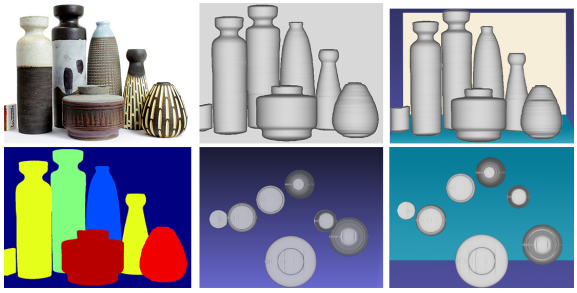


Figure 6: *Ground fitting. Left column: input image and depth layers (hotter colors denote closer layers); Middle column: front and top view of the scene model before ground fitting; Right column: front and top views after ground fitting.*

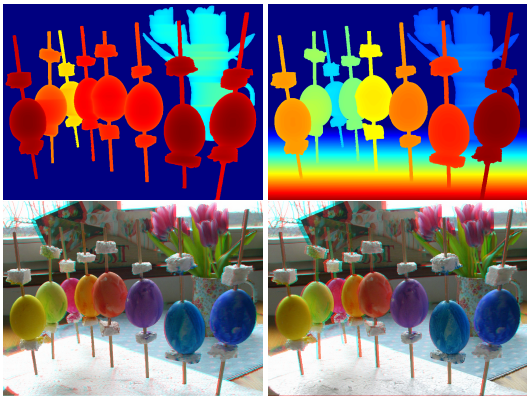


Figure 7: *Two hallucinated depth maps and corresponding anaglyph images. The left column was generated without the ground fitting step; the right column with ground fitting. Without ground fitting the standing objects appear to be floating above their supporting surface. Please magnify and use red/cyan glasses for viewing these images. Several additional examples are included in the supplementary material.*

7. Results

In this section we present and discuss some of the results generated by our method, and compare our results with those produced by several relevant previous works for depth ordering and depth map estimation.

We have implemented our stereo hallucination method in C++. Our current implementation has not been optimized for speed, and the average processing time is 3.3 minutes per image. About 75 percent of the time is spent in the texture completion stage. The time depends on the number and shape of the segments, and the total area where texture has to be completed. The segmentations that our method requires as input were generated using a simple interactive tool based on GrabCut [RKB04]. It took us around 140 seconds, on average, to prepare the segmentation of each image. We obtain stereo pairs from our hallucinated models by placing a vertical rotation axis in the middle of the model and rotating the model around it by a small angle (typically 7 degrees).

All of the results in this section, as well as many additional ones, are also included in the supplementary materials. In the paper we include anaglyph stereo images, which should be viewed at full resolution using red/cyan glasses in order to best experience the stereopsis. In the supplementary material and video we also include for each result a short parallax animation, where the scene is rotated back-and-forth around its vertical axis.

Figure 8 shows three example results along with the by-products of our method: the depth-ordered layers and the resulting depth map after the final depth placement and ground fitting. Despite the extremely simple heuristics used by our layer assignment energy function (1), it may be seen that the overall assignment is roughly correct in all of these examples (red corresponds to nearest, and blue to the farthest layers), although there are some local errors in the layer assignment: the right kid in the second row is assigned to the nearest layer, and in some cases several segments are assigned to the same layer, where in fact some are farther away than others. The ground plane fitting produces a somewhat incorrect depth map in the bottom row, where the ground is significantly non-planar. However, the resulting anaglyph still succeeds in conveying a fairly plausible sensation of 3D.

Figure 9 shows four additional examples (the segmentations, depth layers, and depth maps may be found in the supplementary material). We found the ground fitting to be particularly effective in the two examples shown in the right column: without it the foreground subjects appear floating in front of the background. Although the parallax video sequences sometimes reveal various inaccuracies in our 3D hallucinations, such inaccuracies are generally much more difficult to notice in the still anaglyphs.

Depth ordering accuracy. Since correct depth ordering of segments is important for the quality of our results, we performed an experiment in order to quantify the accuracy of

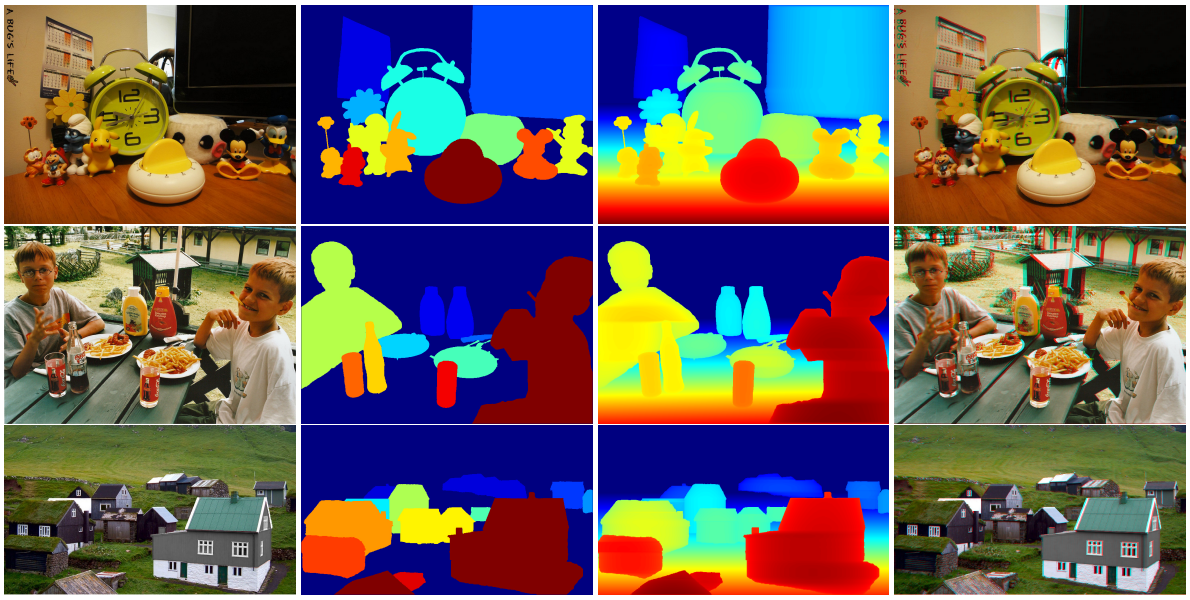


Figure 8: *From left to right: Input image, depth layers, final depth map, anaglyph image. In depth visualizations, hotter is closer and cooler is farther. The reader is strongly encouraged to view these and other results in the supplementary materials, where the anaglyph images may be viewed in full size (using red/cyan glasses). Also included are parallax video sequences, which may be viewed without glasses.*

Figure 9: *Some additional stereo hallucination results: input images and the resulting anaglyphs. The segmentations, depth layers, depth maps, and parallax sequences are included in the supplementary materials.*

this stage in our pipeline, and compare it to two state-of-the-art depth ordering methods [JGCC12, PS13]. Two datasets were used in this experiment: (1) the *d-order* dataset provided by Jia *et al.* includes 1087 images with manual segmentation and ground-truth depth ordering of segments; (2) our own dataset, with 52 near-view images collected from the Internet, which we have manually segmented and depth-ordered (included in the supplementary materials). Since Jia's method requires supervised training, it was trained separately on each of the datasets (using half of the images,

Table 1: *Depth order accuracy comparison (in percents).*

	d-order adj. pairs	d-order all-pairs	near-view adj. pairs	near-view all-pairs
Jia2012	89.26	39.58	79.84	29.88
Palou2013	52.80	50.66	43.85	43.56
Our method	89.56	83.88	82.66	84.60

chosen at random). Table 1 summarizes the results of this experiment.

Accuracy is measured as the percentage of segment pairs that were correctly ordered by each algorithm. In the first and third columns, we report the accuracy only for those segment pairs that are adjacent to each other in the image (for each of the two datasets), while the second and fourth columns report the accuracy when considering all of the segment pairs in each image. We were not able to achieve good results with Jia's method when it was trained using 26 (half) of the images in our near-view dataset. Thus, all of the results for Jia's method were achieved after training with 541 images from the d-order dataset.

It may be seen that on the d-order dataset our method achieves the same level of adjacent pair accuracy as Jia's method, despite being simpler, and requiring no training. Both methods achieve much better accuracy than Palou's method. However, Jia's method is unable to order segments that belong to different connected components in the image. Thus, when all pairs of segments are considered (second column) Jia's accuracy is drastically reduced, and it is significantly outperformed by both ours and Palou's method, neither of which is subject to this limitation.

On the near-view dataset, our method has the highest adjacent pair accuracy among the three. As mentioned earlier, Jia's results on this dataset were achieved after training with the larger, but less similar, d-order training set. It is possible that a much larger training set of near-view images might have resulted in a better accuracy. Again, the same drastic drop in accuracy is observed when considering order between all pairs of segments in the image.

Stereo-viewing experience. We carried out a user study in order to evaluate the effectiveness of our stereo anaglyphs and parallax animations, and compare with a number of alternatives. We have assembled a set of 80 test images consisting of 30 outdoor images with range data, randomly selected from the Make3D dataset [SSN09], 30 indoor images from the NYU-V2 depth dataset [SHKF12], and 20 images from our near-view dataset (the latter 20 images have no ground truth depth map). Anaglyphs and parallax animations were generated from the ground truth depth (when available), and from depth maps generated by the methods of Russell and Torralba [RT09] and Karsch *et al.* [KLK14]. To provide the input for [RT09], we marked the outlines of all of the segments that were given as input to our method, as well as the horizon line, and annotated each object with a suitable semantic label. Karsch's method was trained using suitable image databases (400 images from the Make3D dataset for the outdoor images, and 719 images from NYU-V2 for the remaining indoor and near-view images).

There were 100 participants in our user study (42 females and 58 males, between the ages of 18 to 32), who have first been tested to ensure that they are able to perceive stereo using anaglyph glasses. Each participant was first shown 16 anaglyph pairs of 16 scenes chosen at random. One anaglyph was always produced by our method, while the other by one of the other alternatives. The anaglyphs in each pair were displayed side-by-side with random ordering. The participants were asked to select the anaglyph that resulted in a more convincing stereo effect. Next, each participant was shown 8 pairs of parallax animations without anaglyph glasses, and was asked to indicate which animation better conveys the 3D structure of the scene.

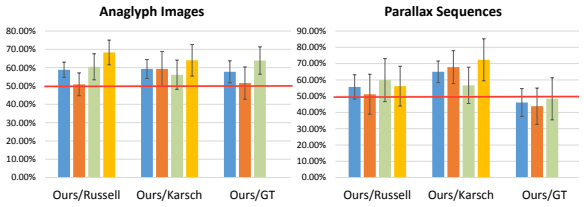


Figure 10: *User study results. Each group of bars shows the average percentage of users that preferred our method's results, with 95% confidence intervals. Blue: average over all images; Orange: average over 30 NYU-V2 images; Green: average over 30 Make3D images; Yellow: average over our set of 20 images. Bars above the red line correspond to cases where our result was preferred by the majority of the users.*

The results of the study, plotted in Figure 10, show that the majority of users prefer the results produced by our method over the results of the two other methods. Our method was preferred in about 59 percent of the anaglyph comparisons with either method, and in 56 and 65 percent of the parallax animation comparisons (compared to [RT09] and [KLK14], respectively). Interestingly, our anaglyph results were also preferred in 58 percent of the comparisons with anaglyphs generated from ground truth depth. This may be attributed to the fact that our results tend to produce a somewhat exaggerated stereo effect, compared to ground truth depth. In the parallax animations most users preferred the ground truth results, but by a rather modest margin (54 percent).

When breaking down the results according to the different image categories, we can see that our method is on par with Russell's on the indoor NYU-V2 scenes (these scenes contain many planar surfaces, well suited for Russell's piecewise planar representation), but outperforms it on the outdoor Make3D scenes. This is despite Russel's method re-

quiring additional user input (horizon lines, semantic labels) and leveraging a large database of labeled images. Our method outperforms Karsch's on both of these scenes (with a larger margin in indoor scenes), without requiring a large RGBD database. For our near-view scenes, our method is preferred by the largest margin over both other methods. The supplementary materials include the full data of the user study, including the images and the distribution of votes for each image.

Figure 11 shows the anaglyphs generated using the different methods above for three scenes for which our method's result received the most votes, as well as for three scenes, where our method received the least number of votes.

The supplementary material also includes a qualitative comparison between our method and three older automatic methods for recovering the 3D structure of a scene from a single image: photo pop-up [HEH05], occlusion recovery [HSEH07], and Make3D [SSN09]. We applied the implementation provided by the original authors on a few outdoor photos, as well as a few of our near-view scenes. Since these methods are completely automatic, for fairness we compared them to our method using two different segmentations: a fully automatic segmentation using [AMFM11], and a manual segmentation. While our results are better with manual segmentation inputs, both our manual and our automatic results are superior to the results of these three methods.

7.1. Assumptions and Limitations

The foremost limitation of our method is that it requires a good manual segmentation. For example, any segmentation where two objects that are separated in depth are labeled as a single segment, or when a single object is split into two or more segments, might lead to visible artifacts.

Our method makes a number of assumptions about the scene. Objects in the scene are assumed to be standing on a level and planar ground; the best results are obtained when each such object has a narrow ground contact region, visible in the image. Although our occluded segment completion heuristics are based on the assumption that objects are convex and symmetric, satisfactory results are obtained also on objects that violate these assumptions. The background geometry is approximated using a fronto-parallel plane, an assumption that is sometimes violated, e.g., by oblique walls in indoor scenes.

The top row of Figure 12 demonstrates a failure case, where the pots and pans are suspended from the ceiling, rather than standing on the ground. The estimated object order is wrong in this case, leading to visible artifacts in the resulting hallucinated stereo. The depth of objects floating in mid air, or those whose contact with the ground is completely occluded, may also be estimated incorrectly.

Our method is best suited for objects with a narrow base.

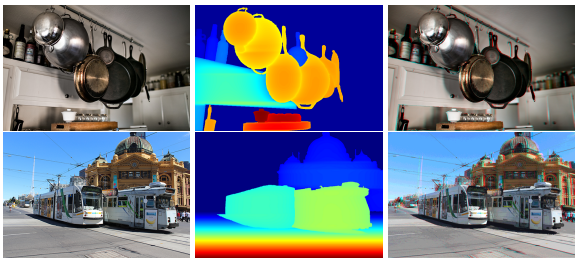


Figure 12: *Limitations. Top row: the depth layers ordering and the resulting depth map are wrong for this scene, where several objects are hanging from the ceiling. As a result, the stereo effect in the resulting anaglyph (top right image) are wrong. Interestingly, applying our method to a vertically flipped version of this image, succeeds in fitting a ground plane to the points of contact with the ceiling, and then flipping the result back produces a better result. Bottom row: the streetcars have a long ground contact base; the depth is determined by the front contact with the ground and the backs of the streetcars appear floating above the ground. This is mostly visible in the parallax animation, but difficult to see in the anaglyph.*

Very long/wide objects whose contact with the ground spans a large range of depths (such as a streetcar standing at an angle to the camera, as shown in the bottom row of Figure 12) will still be modeled using generalized cylinders positioned according to the front contact point, which may lead to the appearance of the object floating above the ground despite our ground fitting.

8. Conclusion

We have presented a new method for generating stereo pairs and parallax from a single image and its segmentation. We have demonstrated that even simple depth and occlusion cues, along with simple shape and symmetry priors may be used to hallucinate a rough 3D scene model, which is nevertheless often sufficient for a rather compelling stereo effect.

We leave it to future work to address the limitations discussed earlier, and hope that as automatic segmentation algorithms continue to improve, and as more sophisticated cues and priors are incorporated into our approach, it will find its way into consumer applications, enabling more users to enjoy an enhanced viewing experience.

Acknowledgements

We thank the anonymous reviewers for their valuable comments. This work was supported by NSFC (National Natural Science Foundation of China, No. 2015CB352500, 61332015 and 61272242), the Israel Science Foundation and the Intel Collaborative Research Institute for Computational Intelligence (ICRI-CI).

Figure 11: *Anaglyphs generated by different methods for six examples from our user study. In each row we show from left to right: the input image, the Russell and Torralba [2009] result, the Karsch et al. [2014] result, and the ground truth depth result, when available. The top three rows correspond to the examples in each of the three image sets where our method was preferred most strongly. The bottom three rows shows examples where our method was least frequently preferred.*

References

[AGCO13] ASAFI S., GOREN A., COHEN-OR D.: Weak convex decomposition by lines-of-sight. *Computer Graphics Forum 32*, 5 (2013), 23–31. 5

[AMFM11] ARBELÁEZ P., MAIRE M., FOWLKES C., MALIK J.: Contour detection and hierarchical image segmentation. *IEEE Trans. PAMI 33*, 5 (Apr. 2011), 898–916. 10

[ART10] AMER M. R., RAICH R., TODOROVIC S.: Monocular Extraction of 2.1D Sketch. In *Proc. ICIP*. 2010, pp. 3437–3440. 3

[AW07] ASSA J., WOLF L.: Diorama construction from a single image. *Computer Graphics Forum 26*, 3 (2007), 599—608. 3

[BSFG09] BARNES C., SHECHTMAN E., FINKELSTEIN A., GOLDMAN D.: PatchMatch: a randomized correspondence algorithm for structural image editing. *ACM Trans. Graph. 28*, 3 (2009), 24. 5

[BVZ01] BOYKOV Y., VEKSLER O., ZABIH R.: Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence 23*, 11 (2001), 1222–1239. 4

[CW93] CHEN S. E., WILLIAMS L.: View interpolation for image synthesis. In *SIGGRAPH'93* (1993), ACM, pp. 279–288. 2

[CZS*13] CHEN T., ZHU Z., SHAMIR A., HU S.-M., COHEN-OR D.: 3-Sweep: Extracting editable objects from a single photo. *ACM Trans. Graph. 32*, 6 (Nov. 2013), 195:1–195:10. 2

[DTM96] DEBEVEC P. E., TAYLOR C. J., MALIK J.: Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *Proc. SIGGRAPH '96* (1996), ACM, pp. 11–20. 2

[GLYG12] GAO J., LIAO M., YANG R., GONG M.: Video stereolization: Combining motion analysis with user interaction. *IEEE Trans. Vis. Comp. Graphics 18*, 7 (2012), 1079–1088. 3

[GWCO09] GUTTMANN M., WOLF L., COHEN-OR D.: Semi-automatic stereo extraction from video footage. In *Proc. ICCV* (2009), pp. 136–142. 3

[HAA97] HORRY Y., ANJYO K.-I., ARAI K.: Tour into the picture: Using a spidery mesh interface to make animation from a single image. In *SIGGRAPH'97* (1997), ACM, pp. 225–232. 2

[HEH05] HOIEM D., EFROS A. A., HEBERT M.: Automatic photo pop-up. *ACM Trans. Graph. 24*, 3 (July 2005), 577–584. 3, 10

[HSEH07] HOIEM D., STEIN A. N., EFROS A. A., HEBERT M.: Recovering occlusion boundaries from a single image. In *Proc. ICCV* (2007), pp. 1–8. 2, 3, 10

[IdH98] IJSSELSTEIJN W. A., DE RIDDER H., HAMBERG R.: Perceptual factors in stereoscopic displays: the effect of stereoscopic filming parameters on perceived quality and reported eye-strain. In *Proceedings of SPIE: Human vision and electronic imaging III* (1998), Rogowitz B. E., Pappas T. N., (Eds.), vol. 3299, SPIE, pp. 282–291. 1

[IMT99] IGARASHI T., MATSUOKA S., TANAKA H.: Teddy: A sketching interface for 3d freeform design. In *Proc. SIGGRAPH* (1999), pp. 409–416. 2

[JGCC12] JIA Z., GALLAGHER A., CHANG Y.-J., CHEN T.: A learning-based framework for depth ordering. In *Proc. CVPR* (2012), pp. 294–301. 3, 5, 8

[Kan98] KANG S. B.: *Depth painting for image-based rendering applications*. Tech. rep., Compaq Cambridge Research Lab, 1998. 2

[KLK14] KARSCH K., LIU C., KANG S.: DepthTransfer: Depth Extraction from Video Using Non-parametric Sampling. *IEEE Trans. PAMI*, to appear (2014). 2, 3, 9

[KRFB06] KHAN E. A., REINHARD E., FLEMING R. W., BÜLTHOFF H. H.: Image-based material editing. *ACM Trans. Graph. 25*, 3 (July 2006), 654–663. 3

[LGK10] LIU B., GOULD S., KOLLER D.: Single image depth estimation from predicted semantic labels. In *Proc. CVPR* (June 2010), pp. 1253–1260. 3

[LH96] LEVOY M., HANRAHAN P.: Light field rendering. In *Proc. SIGGRAPH' 96* (1996), ACM, pp. 31–42. 2

[LMY*13] LIU X., MAO X., YANG X., ZHANG L., WONG T.-T.: Stereoscopizing cel animations. *ACM Trans. Graph. 32*, 6 (Nov. 2013), 223:1–223:10. 3, 5

[LYT11] LIU C., YUEN J., TORRALBA A.: SIFT Flow: Dense correspondence across scenes and its applications. *IEEE Trans. PAMI 33*, 5 (2011), 978–994. 3

[MB95] MCMILLAN L., BISHOP G.: Plenoptic modeling: An image-based rendering system. In *Proc. SIGGRAPH '95* (1995), ACM, pp. 39–46. 2

[OCDD01] OH B. M., CHEN M., DORSEY J., DURAND F.: Image-based modeling and photo editing. In *Proc. SIGGRAPH '01* (2001), ACM, pp. 433–442. 2

[Oli02] OLIVEIRA M. M.: Image-based modeling and rendering techniques: A survey. *RITA - Revista de Informática Teórica e Aplicada IX*, 2 (October 2002), 37–66. 2

[PS13] PALOU G., SALEMBIER P.: Monocular depth ordering using T-junctions and convexity occlusion cues. *IEEE Trans. Image Proc. 22*, 5 (2013), 1926–1939. 3, 8

[RKB04] ROTHER C., KOLMOGOROV V., BLAKE A.: "GrabCut": Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph. 23*, 3 (Aug. 2004), 309–314. 7

[RT09] RUSSELL B. C., TORRALBA A.: Building a database of 3D scenes from user annotations. In *Proc. CVPR*. June 2009, pp. 2711–2718. 2, 3, 9

[SE03] SCHNEIDER P. J., EBERLY D. H.: *Geometric Tools for Computer Graphics*. Elsevier Science (USA), 2003. 6

[SHKF12] SILBERMAN N., HOIEM D., KOHLI P., FERGUS R.: Indoor segmentation and support inference from rgbd images. In *Proc. ECCV* (2012). 9

[SSN09] SAXENA A., SUN M., NG A. Y.: Make3D: Learning 3D scene structure from a single still image. *IEEE Trans. PAMI 31*, 5 (2009), 824–840. 2, 3, 9, 10

[Tau95] TAUBIN G.: A signal processing approach to fair surface design. In *Proc. SIGGRAPH '95* (1995), ACM, pp. 351–358. 6

[TOCR11] TÖPPE E., OSWALD M. R., CREMERS D., ROTHER C.: Image-based 3d modeling via Cheeger sets. In *Computer Vision – ACCV 2010*, vol. 6492 of *Lecture Notes in Computer Science*. Springer, 2011, pp. 53–64. 2

[WKB11] WARD B., KANG S. B., BENNETT E.: Depth director: A system for adding depth to movies. *IEEE Computer Graphics and Applications 31*, 1 (2011), 36–48. 3

[WLF*11] WANG O., LANG M., FREI M., HORNUNG A., SMOLIC A., GROSS M.: StereoBrush: interactive 2D to 3D conversion using discontinuous warps. In *SBIM*. 2011. 3

[YLR*11] YU F., LIU J., REN Y., SUN J., GAO Y., LIU W.: Depth generation method for 2D to 3D conversion. In *Proc. 3DTV-CON* (May 2011), pp. 1–4. 3

[ZDPSS01] ZHANG L., DUGAS-PHOCION G., SAMSON J.-S., SEITZ S. M.: Single view modeling of free-form scenes. In *Proc. CVPR* (2001), IEEE, pp. 990–997. 2