Automatic 3D Indoor Scene Modeling from Single Panorama

Yang Yang[1] Shi Jin[2] Ruiyang Liu[2] Sing Bing Kang[4] Jingyi Yu[2,3]

[1]University of Delaware
yyangwin@udel.edu

[2]ShanghaiTech University
{jinshi,liury}@shanghaitech.edu.cn

[3]Plex-VR Inc.
jingyi.yu@plex-vr.com

[4]Microsoft Research
sbkang@microsoft.com

Abstract

We describe a system that automatically extracts 3D geometry of an indoor scene from a single 2D panorama. Our system recovers the spatial layout by finding the floor, walls, and ceiling; it also recovers shapes of typical indoor objects such as furniture. Using sampled perspective subviews, we extract geometric cues (lines, vanishing points, orientation map, and surface normals) and semantic cues (saliency and object detection information). These cues are used for ground plane estimation and occlusion reasoning. The global spatial layout is inferred through a constraint graph on line segments and planar superpixels. The recovered layout is then used to guide shape estimation of the remaining objects using their normal information. Experiments on synthetic and real datasets show that our approach is state-of-the-art in both accuracy and efficiency. Our system can handle cluttered scenes with complex geometry that are challenging to existing techniques.

1. Introduction

The recent surge in interest in immersive mixed reality has resulted in the widespread availability of both commercial and consumer 360° camera systems. The consumer-grade cameras include Ricoh Theta and Samsung Gear 360 while the higher-end ones include Nokia Ozo and Jaunt One. Immersive panoramas are used in social apps such as Facebook Spaces. Also, consumers can now contribute to Google Street View by uploading panoramas. These developments highlight the importance of panoramic content creation.

Panoramic or 360° images and videos, however, can only provide limited immersive experience, due to lack of stereo parallax. In this paper, we address the problem of automatically adding depth to a single 360° panorama. This has the benefit of being able to synthesize stereo views, thus allowing the user to experience 3D immersive visualization of the scene using a VR headset. We currently restrict the scope to indoor scenes.

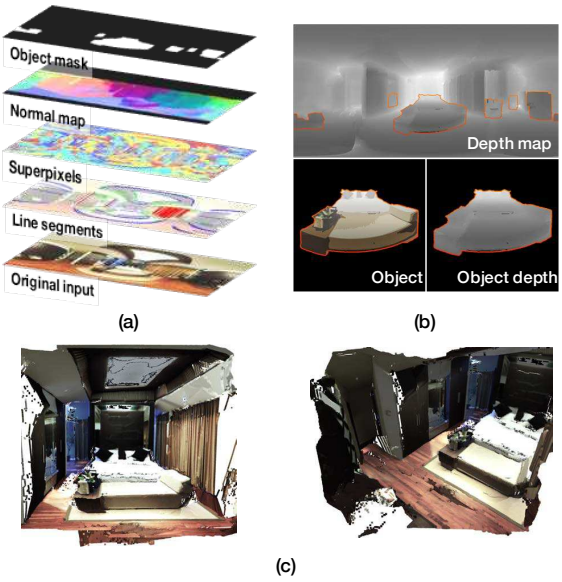

Figure 1. Our approach. (a) Geometric cues (such as line segments, surface normal) and semantic cues (object masks) are used to automatically convert 2D panoramas to 3D ones. (b) Recovered 3D depth map and closeup views of an object. (c) Two different virtual views.

We present a novel efficient technique to infer 3D structure based on a single 2D panorama. More specifically, we estimate the spatial layout (consisting of floor, wall, and ceiling) using the Manhattan world assumption, and recover depths of typical indoor objects (e.g., furniture). Fig. 1 illustrates our proposed technique, with results for a representative input panorama.

What sets our work apart from prior work on adding depth to panoramas is that in addition to inferring the global layout, we use semantic cues and depth propagation to recover object depth. This is critical to producing plausible

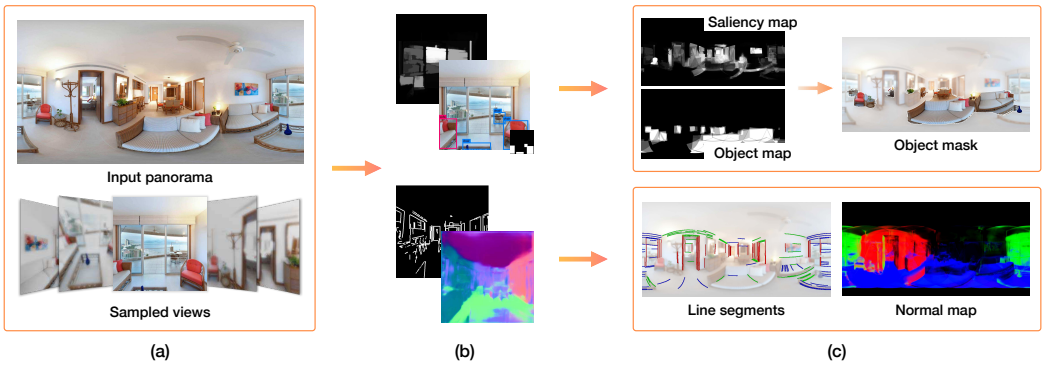

Figure 2. Cue extraction pipeline. (a) We sample the input panorama as local perspective sub-views. (b) We estimate saliency, detect objects (top) and extract lines, per-pixel normals (bottom) for each sub-view image. (c) The scene is partitioned into objects and background using object masks. Lines and planes are used to recover background shape while normal information is used to recover object shape.

depth for the *entire* panorama, which could be of a cluttered indoor scene.

2. Related Work

Much work has been done in extracting depth information from single image. Given the ill-posedness of the problem, approaches with different priors have been used, including low-level vision of shape-from-X, direct use of geometric features, and use of scene semantics. In this section, we briefly review representative techniques.

Shape-from-X. There are many approaches that use shape-from-X [25, 6, 19, 3, 2, 13, 12], which have various levels of success in indoor scene recovery. Most approaches, however, rely on textures to estimate camera parameters. The lack of parallax in single panoramas also makes the application of shape-from-X unstable.

Geometry-based techniques. Geometry-based single view methods [15, 17, 9, 24, 23, 28] typically make use of low-level geometric information such as lines, vanishing points, and geometric priors (walls, ceilings, etc) for reconstruction. Compared with regular perspective images, panoramas, especially indoor panoramas, contain global spatial layouts that link all geometric primitives. A recent solution [30] exploits such spatial arrangements but can fail on cluttered scenes due to occlusions that cause incomplete geometry.

Semantic-based techniques. Semantic methods classify the image into different geometric regions based on appearance cues (e.g., color and texture). Hoiem et al. [10, 11] construct the surface layout by labeling the image using geometric classes. The room layout and objects have also been represented as boxes. For example, Hedau et al. [9, 8] handle clutter by modeling the room space with a parametric 3D "box" and iteratively refitting the box. Schwing et al. [23] jointly infer the room layout with the objects present in the scene. Choi et al. [5] build a 3D Geometric Phrase Model that combines semantic and geometric relationships between objects.

Data-driven techniques. More recently, data-driven approaches have shown certain levels of success. Conceptually, the problem can be reduced to single-image depth inference. [21, 7, 22, 27, 16, 20, 4] learn from extensive 3D data to generate a per-pixel depth map. Karsch et al. [14] use a non-parametric learning framework to recover depth from a single image, and extended that framework to videos. While deep networks can be fine-tuned, the lack of explicit low-level geometric relationships makes it difficult to produce visually compelling results. Moreover, existing networks are trained on perspective images with a limited field-of-view without exploiting panoramic properties that reveal global geometric arrangements.

Depth from panoramas. In contrast with non-panoramic approaches mentioned above, our method explores the global geometric information represented by panoramas. Previous explorations in panoramic reconstruction include [32, 12, 30]. Zhang et al. [32] exploited the contextual constraints of panorama images to overcome the limitation of small field-of-view for object detection. Ikehata et al. [12] addressed a new SfM approach using indoor panoramic image streams as inputs to reconstruct the indoor scenes. They fused single-view and multi-view reconstruction techniques together via geometric relationship detection.

The work of Yang et al. [30] is the closest to ours since they proposed a technique to also recover the shape of 3D rooms from a full-view indoor panorama. However, it fails to recover the object structure in cluttered scenes due to a lack of higher degree of freedom (DOF) clues. By contrast, our algorithm not only considers the geometric cues but also takes semantic cues into account, which primarily improves the algorithm capability for highly cluttered scenes.

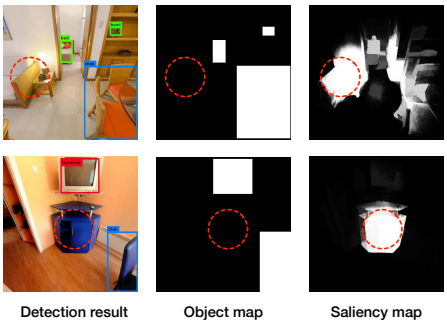

Figure 3. Examples of how the saliency map and object detection information complement each other, with highlights within dashed red circles. Top row: The sofa is only partially visible, and so not detected as an object. Bottom row: The blue cabinet is not detected even though it is fully visible. Both show up as salient regions.

3. Overview of Algorithm

Our system is illustrated in Fig. 2. We first sample from the panorama image to generate local perspective sub-views Fig. 2(a). More specifically, we sample 18 views equally around the panorama, with each view spanning a FOV of 90°. Saliency and object detection information is first extracted per sub-view, as shown in Fig. 2(b). We also collect geometric information, e.g. lines and per-pixel normals, for each sub-view. All the resulting information are mapped to the panorama and analyzed to produce the panoramic object mask, line segments, and normal map shown in Fig. 2(c).

The object mask is used to partition the panorama into layout (background) and object (foreground). The background depth is computed first. Using a piecewise linear assumption, we project the panorama to a sphere. We then partition the sphere into superpixels based on context information, thereby reducing panoramic reconstruction to estimating the depth of each superpixel. We enforce geometric constraints among superpixels and lines and tackle the nonlinear optimization problem by decomposing it into two linear subproblems and solving them in an alternating fashion iteratively. We further construct constraints among superpixels in object regions and reduce their DOF by initializing superpixel normals with the rough surface normal map. Finally, object depth is recovered by propagating depth from the ground portion of the background.

We shall now describe the details of each part, starting with the preprocessing step of cue extraction.

4. Preprocessing Step: Cue Extraction

For each perspective sub-view, we extract geometric cues (namely, lines, geometry context, orientation map, and approximate surface normal) as well as semantic cues (namely, saliency and object detection map).

Line segment and vanishing point extraction. We apply the line segment detection (LSD) algorithm [26] on the sub-views in a similar manner as [32]. Since straight line segments in perspective images correspond to geodesics on a unit sphere in the panorama, we estimate the vanishing points using a geodesic voting method. Subsequently, the line segments are grouped based on the vanishing directions. Fig. 2(c) shows the colored line segments with three Manhattan directions.

Geometric context and orientation map generation. We follow the same strategy in [32] to extract the geometric context (GC) [9] and orientation map (OM) [15], respectively. We use GC for ground region estimation and OM when solving the energy function for extraction of object depth in Sec. 7.

Surface normal estimation. We use the normal estimation network of Bansal et al. [1] to extract the higher DOF pixel-wise surface normal from the sub-views. Fig. 2(b) shows the rough normal map for sampled view, and Fig. 2(c) shows the fused panoramic normal map.

Saliency and object detection. To better isolate the objects from room layouts, we fuse the results of saliency [31] and object detection algorithms [18]. The object detection algorithm works well when the entire object is visible. The first column of Fig. 3 shows that the chairs and television have been correctly detected. However, it fails in the cases where the object is occluded or the object is unusual. Saliency detection helps in these situations because partially visible or unusual objects typically show up as being salient. In the first row of Fig. 3, the sofa (circled) is not detected, while the saliency map includes it. The second row illustrates the same occurrence with the blue cabinet (also circled).

5. Object Mask Generation

To separate the layout (background) from object (foreground) for separate analysis, we extract the object mask for the panorama. Recall that object detection information and saliency map are recovered in each perspective sub-view, as shown in Fig. 2(b). Both types of information are warped back to the panorama (in spherical coordinates). Where the information overlap, we average, yielding the final panoramic results as shown in Fig. 2(c). The object mask in the panorama is then computed using saliency and object detection information that have been transferred and merged from the sub-views.

The object mask is computed by optimizing this unary energy function:

$$E_M = w_o E_o + w_s E_s, \tag{1}$$

where E_M denotes the object mask, E_o, E_s denotes the object detection output and saliency map, respectively. w_o and w_s are weights for two energy terms. We solve this function with the binary-label graph-cut. Fig. 2(c) shows an example object mask that highlights objects in the scene.

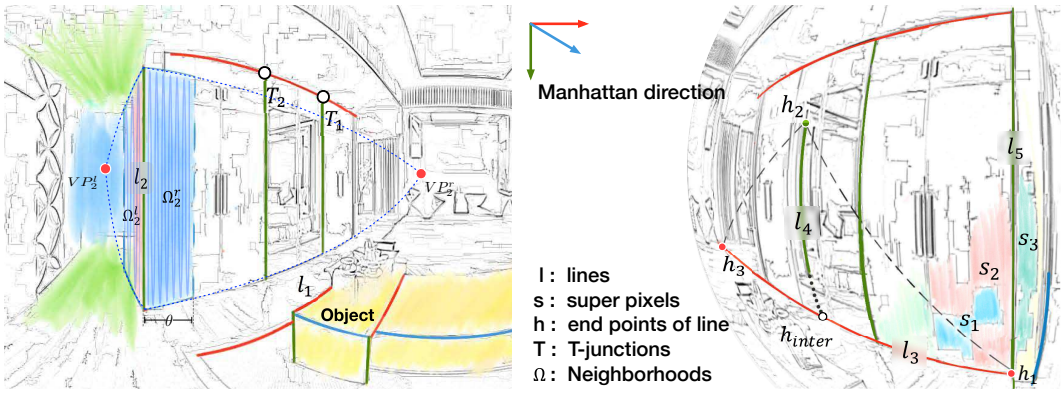

Figure 4. Geometric reasoning. Left: Line occlusions. The Manhattan directions are color-coded. The shaded areas Ω_2^r, Ω_2^l are the spherical quadrilateral neighborhood of l_2 with vanishing points VP_2^r, VP_2^l, respectively. Pixels within Ω_2^r have normals consistent with the blue Manhattan direction, while those within Ω_2^l are voting for different Manhattan directions. Right: Geometry constraints. We use different colors to highlight constraints between different geometric entities (planes, lines, and superpixels). Illustrated here are constraints between intersected lines (l_3 with l_4), line and superpixels (l_5 with s_3), and adjacent superpixels (s_1 with s_2).

6. Extraction of Spatial Layout

We apply a panoramic graph-cut to over-segment the panorama. Instead of imposing the graph-cut onto the original panorama image, we uniformly sample the unit sphere (details are in the supplementary file) and then impose graph-cut onto the sphere for segmentation. We project the spherical superpixels back to the panorama. The number of superpixels varies for different scenes (generally from 500 to 800). Three example superpixels s_1, s_2, s_3 are shown in the right image of Fig. 4.

We denote the set of all superpixels S, and our goal is to recover the normal $\mathbf{n}_i$ for each superpixel s_i as well as its distance d_i to the viewpoint. To do this, we consider pairwise constraints among adjacent superpixels and lines, as shown in the right image of Fig. 4. Using occlusion detection, we can robustly represent the pairwise relationship between line-line, line-superpixel, and superpixel-superpixel in a constraint graph [30].

Additionally, object and ground region information is used as guidance to refine the structure of the constraint graph. We now describe how the ground is detected. Note that we use the Manhattan world assumption in recovering the 3D layout.

6.1. Ground Region Detection

Detecting the ground is a crucial step in computing the layout and object depths depend on knowing where the ground is. The GC labels the ground with a high precision but at rather low recall, while the surface normal estimation from a deep learning model preserves smoothness. We initialize the ground region with superpixels that are labeled as ground by GC; we then use the surface normal maps as seeds to propagate the ground region. This step is necessary because overall, the GC labels are not very accurate. During propagation, we keep a moving average for the pixel-wise normal, which is weighted by $cos(\phi)$ (ϕ being the pitch angle), to compensate for the non-uniform sampling nature of equirectangular projection. The propagation step ends when the angle difference between a new normal and the moving average is larger than $10°$.

6.2. Line Occlusion Detection

For each line l_i, we use two markers (b_i^l, b_i^r), $b_i^{\{l,r\}} \in \{0, 1\}$ to represent occlusion on its left/right sides. $b_i = 1$ indicates the corresponding side is coplanar with the line, while $b_i = 0$ indicates the side is occluded by the line. We use three measures to locally evaluate the possibility of each assignment for each line, namely, *normal consistency*, *object region prior*, and *likelihood of T-junction*:

$$E_{occl} = \omega_n E_n(b_i^l, b_i^r) + \omega_T E_T(b_i^l, b_i^r) + E_{bias}(b_i^l, b_i^r). \quad (2)$$

E_n represents the object region prior and normal consistency, E_T is the likelihood that a T-junction exists, and E_{bias} encodes preference for each assignment.

To measure *normal consistency*, we parametrize the sphere using 8-level subdivision of an embedding icosahedron. We extract *spherical quadrilateral neighborhoods* $\Omega_i^{l,r}$ for each line l_i, as defined by [30], and calculate the normal consistency measure as

$$C_i^{\{l,r\}} = \frac{\sum_{v \in \mathcal{V}_i^{\{l,r\}}} |\mathbf{m}_i^\intercal \mathbf{n}(v)|}{\sum_{v \in \mathcal{V}_i^{\{r,l\}}} |\mathbf{m}_i^\intercal \mathbf{n}(v)| + \tau}, \quad (3)$$

where $\mathcal{V}_i^{\{l,r\}}$ is the set of vertices located in $\Omega_i^{l,r}$, $\mathbf{m}_i$ is the direction of line l_i, and $\mathbf{n}(v)$ is the vertex normal of v acquired from the normal map. $C_i^{\{l,r\}}$ will be close to zero

where vertex normals are consistent and perpendicular to l_i, as neighbor Ω_2^r of l_2 in Fig. 4 (left), otherwise, $C_i^{\{l,r\}}$ will be a large value such as Ω_2^l.

We use weight parameter $c_{obj}^{\{l,r\}}$ to encode the *object region prior*. If line l_i is adjacent to the object mask on one side, we treat l_i as an edge of the object. The object is foreground with a high probability, with the other side of l_i being occluded by the object (e.g., l_1 in Fig. 4(left)). $c_{obj}^{\{l,r\}}$ is decreased on the object side and increased on the other side; $E_n(b_i^l, b_i^r)$ in Eqn. 2 is

$$E_n(b_i^l, b_i^r) = c_{obj}^l C_i^l b_i^l + c_{obj}^r C_i^r b_i^r. \tag{4}$$

As with [30, 17], we also estimate the likelihood of *T-junction* $E_T(b_i^l, b_i^r)$ as evidence of occlusion. The left image in Fig. 4 shows two detected T-junctions T_1, T_2. Let $T_i^{l,r} \in \{0, 1\}$ be the likelihood of a T-junction existing on the {left, right} side of line l_i. The occlusion measure is

$$E_T(b_i^l, b_i^r) = \max\{0, T_i^l - T_i^r\} b_i^l + \max\{0, T_i^r - T_i^l\} b_i^r. \tag{5}$$

We consider $b_i^{\{l,r\}} = 0$ to be a singular configuration, and penalize it by assigning a larger value to corresponding E_{bias}. E_{bias} is a mapping from each assignment of $b_i^{\{l,r\}}$ to a scalar; details are in Section. 9. We minimize E_{occl} for each line independently.

6.3. Layout Constraint

We construct constraints for the 3D layout using background superpixels and lines. We parametrize each superpixel $s_i \in S$ with a normal vector $\mathbf{n}_i \in \mathbb{R}^3$ and a scalar $d_i \in \mathbb{R}$ representing the distance from the viewpoint to s_i. As before, $\mathbf{m}_i$ is the direction of line l_i and is one of the three Manhattan directions. $d_i^L, \mathbf{h}_i^L$ are the depth and normalized direction of an endpoint of l_i, respectively. A pair of adjacent superpixels (s_i, s_j) share a set of points $N_{i,j}$ on their boundaries. We focus on two types of constraints: connection and coplanarity.

Connection Constraints. Spatial smoothness is a prior often used for handling ill-posed problems; in our context, connection constraints are a form of spatial smoothness. We can impose stronger connection constraints because of the additional information extracted in the line occlusion detection step. For each pair of adjacent superpixels (s_i, s_j), we deem them to be connected if no occluding lines intersect $N_{i,j}$ and they have similar depths. In the right image of Fig. 4, the pair of adjacent superpixels s_1, s_2 is one such example of being connected, while s_2, s_3 are not connected (being occluded by l_5). Let $\mathbf{h} \in \mathbb{R}^3$ be the direction of the point in $N_{i,j}$, and $d_i(\mathbf{h})$ be the depth inferred from superpixel s_i, so that $d_i(\mathbf{h}) = d_i/(\mathbf{n_i}^\intercal \mathbf{h})$. We define E_{con}^{pp} to represent the cost:

$$E_{con}^{pp} = \sum_{N_{i,j}} \sum_{\mathbf{h} \in N_{i,j}} \|\mathbf{h}^\intercal (d_i \mathbf{n}_j - d_j \mathbf{n}_i)\|^2. \tag{6}$$

We deem line l_i and superpixel s_j to be connected if s_j is not on the occluded side of l_i. Let N_{l_i,s_j} be the set of common points of l_i, s_j, and let $M_i^\perp$ be the set of two Manhattan directions perpendicular to $\mathbf{m}_i$. We enforce the depth consistency on N_{l_i,s_j} using the following objective:

$$E_{con}^{lp} = \sum_{N_{l_i,s_j}} \sum_{\mathbf{h} \in N_{l_i,s_j}} \sum_{\mathbf{m} \in M_i^\perp} \|\mathbf{m}^\intercal (d_j \mathbf{h} - (\mathbf{h}^\intercal \mathbf{n}_j) d_i^L \mathbf{h}_i^L)\|^2. \tag{7}$$

Coplanarity Constraints. We apply these constraints on lines and superpixels to check for coplanarity. The general idea for coplanarity detection is similar to that of [12], but we extend the framework to equirectangular coordinates. For each pair of lines (l_i, l_j) of different directions, we calculate their intersection $\mathbf{h_{inter}}$. We also compute distance $Q(\mathbf{h}_{inter}, l_i)$ between l_i to $\mathbf{h}_{inter}$ defined as:

$$Q(\mathbf{h}_{inter}, l_i) = \min_{\mathbf{h} \in l_i} \arccos\left(\frac{\mathbf{h}^\intercal \mathbf{h}_{inter}}{\|\mathbf{h}\| \|\mathbf{h}_{inter}\|}\right). \tag{8}$$

We define the distance between $\mathbf{h}_{inter}$ and the pair of lines (l_i, l_j) to be $\max(Q(\mathbf{h}_{inter}, l_i), Q(\mathbf{h}_{inter}, l_j))$. If this value is less than 10°, we further check the normal consistency of (l_i, l_j).

We denote H to be the spherical convex hull (spherical triangle or quadrilateral) for the point set of (l_i, l_j) and $\mathbf{h}_{inter}$. We then project H back to the surface normal map to retrieve normal values. We deem (l_i, l_j) to be coplanar if: 1) the average normal is less than 15° from the expected Manhattan direction, and 2) the standard deviation of normal values in H is less than 5°. We also verify that H is located on both the non-occlusion sides of l_i, l_j. As the illustration, in Fig. 4(right), (l_3, l_4) is a pair of coplanar lines, and their convex hull is a spherical triangle.

If a pair of lines are determined to be coplanar, we find the superpixels that intersect H; we refer to this set of superpixels as $S_{cop}(l_i, l_j)$. For each superpixel, we enforce its normal perpendicular to the direction of (l_i, l_j) by optimizing

$$E_{cop}^{p} = \sum_{(l_i,l_j)} \sum_{s_k \in S_{cop}(l_i,l_j)} \|\mathbf{n}_k^\intercal \mathbf{m}_i\|^2 + \|\mathbf{n}_k^\intercal \mathbf{m}_j\|^2. \tag{9}$$

Coplanarity provides an additional type of constraint among pairs of detached lines and superpixels. For each $s_k \in S_{cop}(l_i, l_j)$, we apply constraints on d_k similar to the connection case:

$$E_{cop}^{lp} = \sum_{(l_i,l_j)} \sum_{s_k \in S_{cop}(l_i,l_j)} \|d_k - \mathbf{n}_k^\intercal \mathbf{h}_i^L d_i^L\|^2 + \|d_k - \mathbf{n}_k^\intercal \mathbf{h}_j^L d_j^L\|^2. \tag{10}$$

We directly minimize the difference between the normal of these superpixels and the expected Manhattan direction.

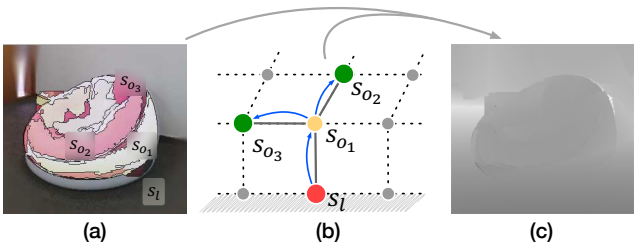

Figure 5. Object depth propagation. (a) Background (layout) and object superpixels (s_l and s_o, respectively). (b) Our propagation process. (c) Resulting depthmap of both background (layout) and object.

7. Extraction of Object Depth

Indoor objects such as sofas and beds almost always rest on the ground; as such, we can infer object depth by propagating depth information from the ground plane and avoid "floating" objects or objects interpenetrating with the ground. Fig. 5 shows the process of object depth propagation that starts from the layout (background) superpixel s_l to connected object superpixel s_{o_1}, before propagating to other nodes (s_{o_2}, s_{o_3}).

To compute object depth, we first find adjacent pairs of superpixels located near the boundary of ground and object regions. Let us denote such a pair (s_o, s_l), as shown in Fig. 5(a). We check the line occlusion detection results to make sure no occluding lines are covering the intersection of (s_o, s_l). Subsequently, depth consistency is imposed along the points of intersection of both superpixels, similar to the connection constraints in Eqn. 6.

For superpixels and lines located inside the object region, we apply *connection* constraints between adjacent superpixels. Similar to Eqn. 6, we impose consistency of depths on the boundary pixels among adjacent superpixels and check no *occluding line* intersecting their boundary. We impose consistency of depth on the boundary pixels among adjacent superpixels within the same side of any occluding line. We choose not to detect coplanarity for the lines and superpixels due to the possibility of the surface being curved.

Finally, we sum all the connection terms as E_{con} and sum all coplanarity terms as E_{cop}, yielding a weighted energy objective function:

$$E = \omega_{con}(E^{pp}_{con} + E^{lp}_{con}) + \omega_{cop}(E^{p}_{cop} + E^{lp}_{cop}), \quad (11)$$

where $\omega_{con}, \omega_{cop}$ are two hyperparameters assigned manually.

8. Optimizing Depth of Entire Panorama

The energy function is the sum of non-linear squares. We initialize the normals for spatial layout with orientation map (OM), enforcing the initial layout normals in three Manhattan directions. We then use the surface normal map to initialize object normals. We can alternatingly solve all depth scalars d while fixing superpixel normals $\mathbf{n}$ and solve superpixel normals while fixing depth. The complete algorithm is described in Algorithm 1; note that $1/2$ refers to the intermediate results generated from updating d. During each step, we use weighted averages to update normals conservatively, with β as the update rate. To avoid the degenerate case in which all depth values are zero, we find the superpixels labeled as ground in GC, and enforce the depth scalar d of such superpixels to be at least 1.

Algorithm 1: Our constrained optimization algorithm.

Initialize superpixel normal in layout with OM ;
Initialize superpixel normal in object region with surface normal map ;
repeat
  $d^{k+1} \leftarrow \arg\min_d(\omega_{con}E^{k}_{con} + \omega_{cop}E^{k}_{cop})$;
  $\tilde{\mathbf{n}} \leftarrow \arg\min_{\mathbf{n}}(\omega_{con}E^{k+1/2}_{con} + \omega_{cop}E^{k+1/2}_{cop})$;
  $\mathbf{n}^{k+1} \leftarrow (1-\beta)\mathbf{n}^{k} + \beta\tilde{\mathbf{n}}$;
  $\mathbf{n}^{k+1} \leftarrow \frac{\mathbf{n}^{k+1}}{\|\mathbf{n}^{k+1}\|}$;
  $k \leftarrow k+1$;
until $\Delta|d| < \epsilon_1$ *and* $\Delta|\mathbf{n}| < \epsilon_2$ *or maximum iteration*;
return $(d^k, \mathbf{n}^k)$;

9. Experimental Results

We validated our approach using three datasets. One dataset we use is SUN360 database [29], which consists of 40 full-view indoor panoramas (with no depth information). Another dataset of 20 panoramas is created using textured 3D point clouds that we captured using a FARO 3D scanner. Each panorama (with depth) is generated by projecting the point cloud to image space. Note that because the scanner does cover the full panoramic space, we manually labeled regions within the panorama as having valid data for evaluation purposes. The third dataset consists of 60 synthetic panoramas we created using 3ds Max; note that we also generate a depth map and normal map for each panorama. Quantitative evaluation is performed on these datasets for comparison with other state-of-the-art methods.

Implementation details. In our experiments, we set weights $w_s = 0.3, w_o = 0.7$ for generating object mask. For line occlusion detection, the weight $c^{\{l,r\}}_{obj} = 0.1$ in object region; otherwise $c^{\{l,r\}}_{obj} = 1$. We set weights for normal consistency and T-junction $\omega_n = 3$, $\omega_T = 1$, and $E_{bias} = 0$ if $\mathbf{b}_i = (1,1)$, $E_{bias} = 10$ if $\mathbf{b}_i = (0,0)$, otherwise $E_{bias} = 2$. For connection and coplanarity constraints, we set $\omega_{con} = 0.8, \omega_{cop} = 0.2$. For Algorithm 1, we set

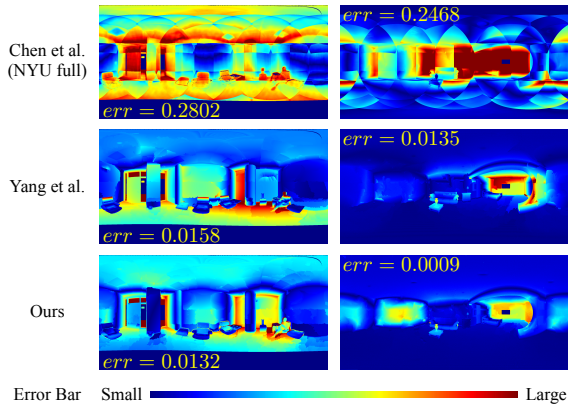

Figure 6. Comparison of depth error maps (normalized by mean value) and average L_2 errors for two panoramas.

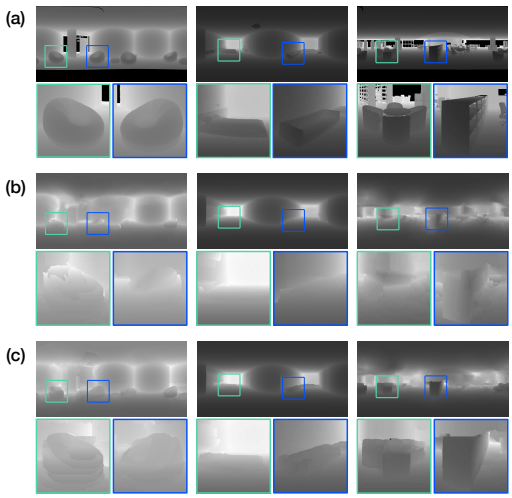

Figure 7. Comparison of depth maps for three panoramas: (a) ground truth, (b) results of [30], and (c) our results.

weight $\beta = 0.7$ and $\epsilon_1, \epsilon_2 = 10^{-4}$. Although we formulate the layout constraints and object constraints in a single constraint graph, we solve the layout depth and the object depth separately to avoid solving the large sparse equation. We use parameter $\omega_{ground} = 2$ in solving for object depth, and further enforce the depth consistency between adjacent ground superpixels and object superpixels.

We implement our algorithm using MATLAB with C++ mex functions on a PC with Intel Core i7-4790 (3.60GHz) processor and 8GB RAM. Cue extraction takes about 5 minutes, and the subsequent inference for a single 2048×1024 full panorama takes less than 2 minutes. It takes about 1 minute to construct the constraint graph. Solving the iterative optimization problem takes less than 10 seconds.

Quantitative analysis. We evaluate the quality of our recovered 3D room model on the labeled dataset and compare the depth maps with ground truth. For FARO scanner data, we compute depth map error in manually labeled valid regions only. The error metric we use is the *cosine distance*, i.e., $1 - \mathbf{d}_1^\intercal \mathbf{d}_2 / \|\mathbf{d}_1\| \|\mathbf{d}_2\|$, where $\mathbf{d}_i$ is the vector uniformly sampled from depth maps. Note that we normalize the predicted and ground truth depth maps using the mean value to eliminate scale ambiguity.

	Background		Object Region	
	FARO	Syn.	FARO	Syn.
Ours	**5.59**	**3.12**	**2.43**	**3.27**
Yang et al. [30]	6.10	7.56	2.94	3.53
Chen et al. [4]	8.51	10.01	7.62	16.33

Table 1. Comparison of background and object region depth cosine distance ($\times 10^{-2}$) for the FARO and synthetic datasets.

We compare our method with the state-of-the-art geometry-based method [30] and data-driven method [4]. The network of Chen et al. [4] takes a perspective image as input and outputs depth along the *z-axis*. We warp the perspective depths generated by network back to the panorama and average the depths where they overlap; the resulting merged-by-averaging panoramic depth map is used for comparison.

Table 1 compares the average depth cosine errors over the background (columns 2 and 3) and over object regions only (columns 4 and 5). Fig. 6 shows typical L_2 error maps for two panoramas. Table 2 shows the time complexity and performance corresponding to each cue extraction, while self-evaluation results are shown in Table 3. Our experiments show that our method significantly outperforms other competing techniques on the quality of recovered depth, especially the object regions. In particular, our results compared with Chen et al. [4] appear to point to the benefit of using panoramic images with stronger global constraints for depth inference.

Cue extraction	TC	AT (secs)
Line segment	$O(n_{ls})$	20
Vanishing point	$O(n_{vp}^2)$	40
Geometric context	$O(n_{gc}^2)$	90
Orientation map	$O(n_{om}^2)$	60
Surface normal estimation	$O(d^2)$	40
Saliency & object detection	$O(d^2)$	30

Table 2. Time complexity and performance. TC = time complexity, AT = average time, n_x = #features (line segment ($x = ls$), vanishing point ($x = vp$), or size of geometric context ($x = gc$) and orientation map ($x = om$)), d = #input nodes of fully connected layer in neural network.

Qualitative analysis. Fig. 7 compares depth maps for three panoramas. Despite a large number of superpixels in our segmentation map, our depth map has better detail com-

Figure 8. Our reconstruction results on eight different panoramas. In each case, immediately below the input panorama is the estimated depth map; to its right is the shaded mesh and two virtual views.

	cosine ($\times 10^{-2}$)	L_2
Proposed	**5.12**	**0.16**
Proposed (no SF)	5.30	0.18
Proposed (no SC)	5.48	0.19
Proposed (no GC)	6.55	0.23
Proposed (no OM)	6.13	0.21

Table 3. Self-evaluation using FARO and synthetic datasets. SF = surface normal, SC = semantic cue, GC = geometric context, and OM = orientation map.

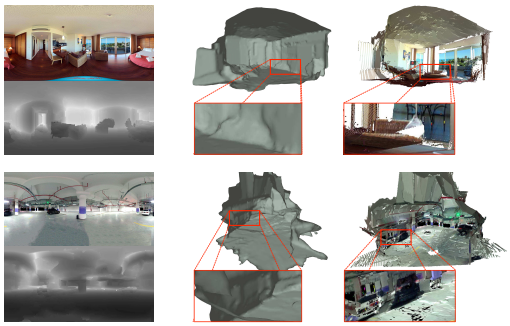

Figure 9. Failure cases. In the first row, the sofa is not recovered. In the second row, the cars are interpreted as wall texture.

pared to that for Yang et al. [30]. This can be seen in the closeup views. Fig. 8 shows some reconstruction results; our system can convert 2D panoramas of indoor scenes to 3D room models with objects. The different virtual views show how well the object depths have been extracted.

Our technique can fail if the object mask incorrectly localizes objects, as can be seen in the top row of Fig. 9. Our system is highly dependent on the accuracy of semantic cues; failures occur where both object and saliency detection fail. Here, since the object mask is misaligned, our system is not able to recover the shape of the sofa. The second failure case shown in the bottom row is a result of an incorrect normal map; this causes the cars to appear to be part of the wall.

10. Concluding Remarks

We have presented a new system to add depth to a 2D panorama of an indoor scene. Our system uses semantic cues and geometric cues to partition the panorama into background (layout) and foreground (object). The layout, which includes the ground plane, is recovered using geometric cues. Layout information is then used to help extract object depth. Experiments show that our approach can handle challenging cluttered scenes that are problematic for state-of-the-art techniques.

Our approach is currently limited to indoor scenes; it would be interesting to investigate how to extend it for outdoor scenes. Conceptually, a similar processing pipeline can be adopted if we can effectively exploit cues unique to outdoor structures such as buildings and roads. Given the popularity of panoramic video, another direction would be to add depth to such videos with additional use of structure from motion and temporal regularization.

Acknowledgement

This work is partially supported by National Science Foundation under the Grant IIS-1422477. Majority of the work was performed while Yang Yang was an intern at Plex-VR Inc.

References

[1] A. Bansal, B. Russell, and A. Gupta. Marr revisited: 2d-3d alignment via surface normal prediction. In *Conference on Computer Vision and Pattern Recognition*, pages 5965–5974, 2016. 3

[2] S. Y. Bao, A. Furlan, L. Fei-Fei, and S. Savarese. Understanding the 3d layout of a cluttered room from multiple images. In *Applications of Computer Vision (WACV), 2014 IEEE Winter Conference on*, pages 690–697. IEEE, 2014. 2

[3] R. Cabral and Y. Furukawa. Piecewise planar and compact floorplan reconstruction from images. In *CVPR, 2014 IEEE Conference on*, pages 628–635. IEEE, 2014. 2

[4] W. Chen, Z. Fu, D. Yang, and J. Deng. Single-image depth perception in the wild. In *Advances in Neural Information Processing Systems*, pages 730–738, 2016. 2, 7

[5] W. Choi, Y.-W. Chao, C. Pantofaru, and S. Savarese. Understanding indoor scenes using 3d geometric phrases. In *Computer Vision and Pattern Recognition*, pages 33–40, 2013. 2

[6] D. Crandall, A. Owens, N. Snavely, and D. Huttenlocher. Discrete-continuous optimization for large-scale structure from motion. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3001–3008. IEEE, 2011. 2

[7] E. Delage, H. Lee, and A. Y. Ng. A dynamic bayesian network model for autonomous 3d reconstruction from a single indoor image. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2418–2428. IEEE, 2006. 2

[8] V. Hedau, D. Hoiem, and D. Forsyth. Thinking inside the box: Using appearance models and context based on room geometry. In *European Conference on Computer Vision*, pages 224–237. Springer, 2010. 2

[9] V. Hedau, D. Hoiem, and D. A. Forsyth. Recovering the spatial layout of cluttered rooms. In *ICCV*, pages 1849–1856. IEEE Computer Society, 2009. 2, 3

[10] D. Hoiem, A. A. Efros, and M. Hebert. Automatic photo pop-up. *ACM transactions on graphics (TOG)*, 24(3):577–584, 2005. 2

[11] D. Hoiem, A. A. Efros, and M. Hebert. Recovering surface layout from an image. *International Journal of Computer Vision*, 75(1):151–172, 2007. 2

[12] S. Ikehata, I. Boyadzhiev, Q. Shan, and Y. Furukawa. Panoramic structure from motion via geometric relationship detection. *CoRR*, 2016. 2, 5

[13] S. Ikehata, H. Yang, and Y. Furukawa. Structured indoor modeling. In *International Conference on Computer Vision*, pages 1323–1331, 2015. 2

[14] K. Karsch, C. Liu, and S. B. Kang. Depth transfer: Depth extraction from video using non-parametric sampling. *Pattern Analysis and Machine Intelligence*, 36:2144–2158, 2014. 2

[15] D. C. Lee, M. Hebert, and T. Kanade. Geometric reasoning for single image structure recovery. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2136–2143. IEEE, 2009. 2, 3

[16] F. Liu, C. Shen, and G. Lin. Deep convolutional neural fields for depth estimation from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5162–5170, 2015. 2

[17] S. Ramalingam and M. Brand. Lifting 3d manhattan lines from a single image. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 497–504, 2013. 2, 5

[18] J. Redmon and A. Farhadi. YOLO9000: better, faster, stronger. *CoRR*, 2016. 3

[19] C. Richardt, Y. Pritch, H. Zimmer, and A. Sorkine-Hornung. Megastereo: Constructing high-resolution stereo panoramas. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1256–1263, 2013. 2

[20] A. Roy and S. Todorovic. Monocular depth estimation using neural regression forest. In *Conference on Computer Vision and Pattern Recognition*, pages 5506–5514, 2016. 2

[21] A. Saxena, S. H. Chung, and A. Y. Ng. Learning depth from single monocular images. In *NIPS*, volume 18, pages 1–8, 2005. 2

[22] A. Saxena, S. H. Chung, and A. Y. Ng. 3-d depth reconstruction from a single still image. *International journal of computer vision*, 76(1):53–69, 2008. 2

[23] 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

[24] A. G. Schwing, T. Hazan, M. Pollefeys, and R. Urtasun. Efficient structured prediction for 3d indoor scene understanding. In *Computer Vision and Pattern Recognition*, pages 2815–2822. IEEE, 2012. 2

[25] N. Snavely, S. Seitz, and R. Szeliski. Photo tourism: Exploring image collections in 3d. acm transactions on graphics. *ACM Transactions on Graphics*, 2006. 2

[26] R. G. von Gioi, J. Jakubowicz, J.-M. Morel, and G. Randall. Lsd: A fast line segment detector with a false detection control. *IEEE transactions on pattern analysis and machine intelligence*, 32(4):722–732, 2010. 3

[27] H. Wang, S. Gould, and D. Roller. Discriminative learning with latent variables for cluttered indoor scene understanding. *Communications of the ACM*, 56(4):92–99, 2013. 2

[28] P. Wang, X. Shen, Z. Lin, S. Cohen, B. Price, and A. L. Yuille. Towards unified depth and semantic prediction from a single image. In *Computer Vision and Pattern Recognition*, pages 2800–2809, 2015. 2

[29] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*, pages 3485–3492. IEEE, 2010. 6

[30] H. Yang and H. Zhang. Efficient 3d room shape recovery from a single panorama. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 5422–5430, 2016. 2, 4, 5, 7, 8

[31] J. Zhang, S. Sclaroff, Z. Lin, X. Shen, B. Price, and R. Mech. Minimum barrier salient object detection at 80 fps. In *IEEE International Conference on Computer Vision*, pages 1404–1412, 2015. 3

[32] Y. Zhang, S. Song, P. Tan, and J. Xiao. Panocontext: A whole-room 3d context model for panoramic scene understanding. In *European Conference on Computer Vision*, pages 668–686. Springer, 2014. 2, 3