3D Scene Reconstruction with Multi-layer Depth and Epipolar Transformers

Daeyun Shin[1] Zhile Ren[2] Erik B. Sudderth[1] Charless C. Fowlkes[1]
[1]University of California, Irvine [2]Georgia Institute of Technology

https://research.dshin.org/iccv19/multi-layer-depth

Abstract

We tackle the problem of automatically reconstructing a complete 3D model of a scene from a single RGB image. This challenging task requires inferring the shape of both visible and occluded surfaces. Our approach utilizes viewer-centered, multi-layer representation of scene geometry adapted from recent methods for single object shape completion. To improve the accuracy of view-centered representations for complex scenes, we introduce a novel "Epipolar Feature Transformer" that transfers convolutional network features from an input view to other virtual camera viewpoints, and thus better covers the 3D scene geometry. Unlike existing approaches that first detect and localize objects in 3D, and then infer object shape using category-specific models, our approach is fully convolutional, end-to-end differentiable, and avoids the resolution and memory limitations of voxel representations. We demonstrate the advantages of multi-layer depth representations and epipolar feature transformers on the reconstruction of a large database of indoor scenes.

1. Introduction

When we examine a photograph of a scene, we not only perceive the 3D shape of visible surfaces, but effortlessly infer the existence of many invisible surfaces. We can make strong predictions about the complete shapes of familiar objects despite viewing only a single, partially occluded aspect, and can infer information about the overall volumetric occupancy with sufficient accuracy to plan navigation and interactions with complex scenes. This remains a daunting visual task for machines despite much recent progress in detecting individual objects and making predictions about their shape. *Convolutional neural networks* (CNNs) have proven incredibly successful as tools for learning rich representations of object identity which are invariant to intra-category variations in appearance. Predicting 3D shape rather than object category has proven more challenging since the output space is higher dimensional and carries more structure than simple regression or classification tasks.

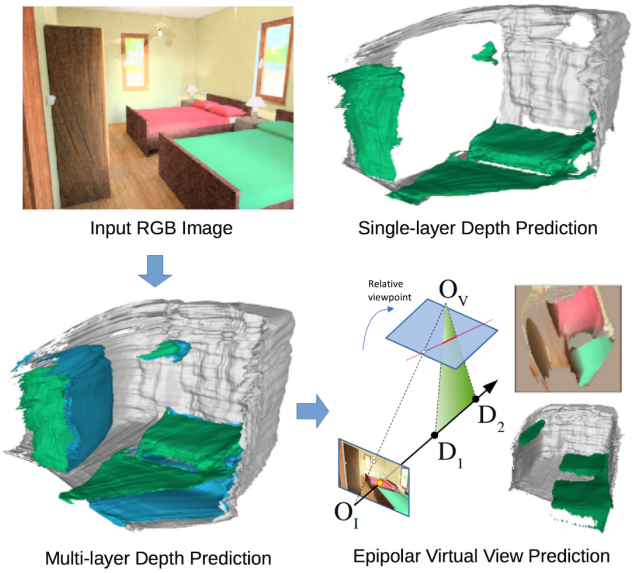


Figure 1: Given a single input view of a scene (top left), we would like to predict a complete geometric model. Depth maps (top right) provide an efficient representation of scene geometry but are incomplete, leaving large holes (e.g., the wardrobe). We propose multi-layer depth predictions (bottom left) that provide complete view-based representations of shape, and introduce an epipolar transformer network that allows view-based inference and prediction from virtual viewpoints (like overhead views, bottom right).

Early successes at using CNNs for shape prediction leveraged direct correspondences between the input and output domain, regressing depth and surface normals at every input pixel [8]. However, these so-called 2.5D representations are incomplete: they don't make predictions about the back side of objects or other occluded surfaces. Several recent methods instead manipulate voxel-based representations [40] and use convolutions to perform translation-covariant computations in 3D. This provides a more complete representation than 2.5D models, but suffers from substantial storage and computation expense that scales cubically with resolution of the volume being modeled (without specialized representations like octrees [31]). Other

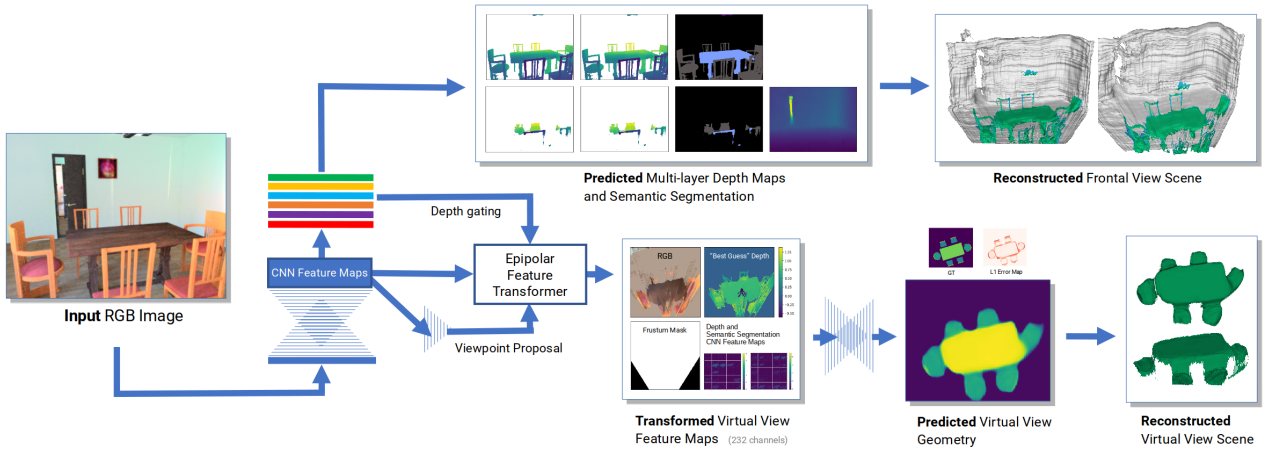


Figure 2: Overview of our system for reconstructing a complete 3D scene from a single RGB image. We first predict a multi-layer depth map that encodes the depths of front and back object surfaces as seen from the input camera. Given the extracted feature map and predicted multi-layer depths, the epipolar feature transformer network transfers features from the input view to a virtual overhead view, where the heights of observed objects are predicted. Semantic segmentation masks are inferred and inform our geometry estimates, but explicit detection of object instances is not required, increasing robustness.

approaches represent shape as an unstructured point cloud [29, 41], but require development of suitable convolutional operators [11, 48] and fail to capture surface topology.

In this paper, we tackle the problem of automatically reconstructing a *complete* 3D model of a scene from a single RGB image. As depicted in Figures 1 and 2, our approach uses an alternative shape representation that extends view-based 2.5D representations to a complete 3D representation. We combine *multi-layer* depth maps that store the depth to multiple surface intersections along each camera ray from a given viewpoint, with *multi-view* depth maps that record surface depths from different camera viewpoints.

While multi-view and multi-layer shape representations have been explored for single object shape completion, for example by [35], we argue that multi-layer depth maps are particularly well suited for representing full 3D scenes. *First*, they compactly capture high-resolution details about the shapes of surfaces in a large scene. Voxel-based representations allocate a huge amount of resources to simply modeling empty space, ultimately limiting shape fidelity to much lower resolution than is provided by cues like occluding contours in the input image [40]. A multi-layer depth map can be viewed as a run-length encoding of dense representations that stores only transitions between empty and occupied space. *Second*, view-based depths maintain explicit correspondence between input image data and scene geometry. Much of the work on voxel and point cloud representations for single object shape prediction has focused on predicting a 3D representation in an object-centered coordinate system. Utilizing such an approach for scenes requires additional steps of detecting individual objects and estimating their pose in order to place them back into some global scene coordinate system [44]. In contrast, view-based multi-depth predictions provide a single, globally coherent scene representation that can be computed in a "fully convolutional" manner from the input image.

One limitation of predicting a multi-layer depth representation from the input image viewpoint is that the representation cannot accurately encode the geometry of surfaces which are nearly tangent to the viewing direction. In addition, complicated scenes may contain many partially occluded objects that require a large number of layers to represent completely. We address this challenge by predicting additional (multi-layer) depth maps computed from virtual viewpoints elsewhere in the scene. To link these predictions from virtual viewpoints with the input viewpoint, we introduce a novel *Epipolar Feature Transformer* (EFT) network module. Given the relative poses of the input and virtual cameras, we transfer features from a given location in the input view feature map to the corresponding epipolar line in the virtual camera feature map. This transfer process is modulated by predictions of surface depths from the input view in order to effectively re-project features to the correct locations in the overhead view.

To summarize our contributions, we propose a view-based, multi-layer depth representation that enables fully convolutional inference of 3D scene geometry and shape completion. We also introduce EFT networks that provide geometrically consistent transfer of CNN features between cameras with different poses, allowing end-to-end training for multi-view inference. We experimentally characterize the completeness of these representations for describing the 3D geometry of indoor scenes, and show that models trained to predict these representations can provide better

recall and precision of scene geometry than existing approaches based on object detection.

2. Related Work

The task of recovering 3D geometry from 2D images has a rich history dating to the visionary work of Roberts [32].

Monocular Object Shape Prediction. Single-view 3D shape reconstruction is challenging because the output space is under-constrained. Large-scale datasets like ShapeNet [1, 52] facilitate progress in this area, and recent methods have learned geometric priors for object categories [22, 51], disentangled primitive shapes from objects [13, 60], or modeled surfaces [15, 35, 55]. Other work aims to complete the occluded geometric structure of objects from a 2.5D image or partial 3D scan [33, 6, 50, 54]. While the quality of such 3D object reconstructions continues to grow [23, 48], applications are limited by the assumption that input images depict a single, centered object.

3D Scene Reconstruction. We seek to predict the geometry of full scenes containing an unknown number of objects; this task is significantly more challenging than object reconstruction. Tulsiani *et al.* [44] factorize 3D scenes into detected objects and room layout by integrating separate methods for 2D object detection, pose estimation, and object-centered shape prediction. Given a depth image as input, Song *et al.* [40] propose a volumetric reconstruction algorithm that predicts semantically labeled 3D voxels. Another general approach is to retrieve exemplar CAD models from a large database and reconstruct parts of scenes [18, 59, 14], but the complexity of CAD models may not match real-world environments. While our goals are similar to Tulsiani *et al.*, our multi-layered depth estimates provide a denser representation of complex scenes.

Representations for 3D Shape Prediction. Most recent methods use voxel representations to reconstruct 3D geometry [3, 40, 37, 46, 36], in part because they easily integrate with 3D CNNs [52] for high-level recognition tasks [25]. Other methods [9, 24] use dense point clouds representations. Classic 2.5D depth maps [8, 2] recover the geometry of visible scene features, but do not capture occluded regions. Shin *et al.* [35] empirically compared these representations for object reconstruction. We extend these ideas to whole scenes via a multi-view, multi-layer depth representation that encodes the shape of multiple objects.

Learning Layered Representations. Layered representations [47] have proven useful for many computer vision tasks including segmentation [12] and optical flow prediction [43]. For 3D reconstruction, decomposing scenes into layers enables algorithms to reason about object occlusions and depth orderings [16, 38, 49]. Layered 2.5D representations such as the two-layer decompositions of [45, 7] infer the depth of occluded surfaces facing the camera. Our multi-layer depth representation extends this idea by including the depth of back surfaces (equiv. object thickness). We also infer depths from virtual viewpoints far from the input view for more complete coverage of 3D scene geometry. Our use of layers generalizes [30], who used multiple intersection depths to model non-convexities for constrained scenes containing a single, centered object. Concurrently to our work, [27] predicts object-level thicknesses for volumetric RGB-D fusion and [10] estimates 3D human shape.

Multi-view Shape Synthesis. Many classic 3D reconstruction methods utilize multi-view inputs to synthesize 3D shapes [17, 39, 4]. Given monocular inputs, several recent methods explore ways of synthesizing object appearance or image features from novel viewpoints [58, 53, 20, 3, 28, 42]. Other work uses unsupervised learning from stereo or video sequences to reason about depths [57, 21]. Instead of simply transferring the pixel colors associated with surface points to novel views, we transfer whole CNN feature maps over corresponding object volumes, and thereby produce more accurate and complete 3D reconstructions.

3. Reconstruction with Multi-Layer Depth

Traditional depth maps record the depth at which a ray through a given pixel first intersects a surface in the scene. Such 2.5D representations of scene geometry accurately describe visible surfaces, but cannot encode the shape of partially occluded objects, and may fail to capture the complete 3D shape of unoccluded objects (due to self-occlusion). We instead represent 3D scene geometry by recording multiple surface intersections for each camera ray. As illustrated in Figure 3(a), some rays may intersect many object surfaces and require several layers to capture all details. But as the number of layers grows, multi-layer depths completely represent 3D scenes with multiple non-convex objects.

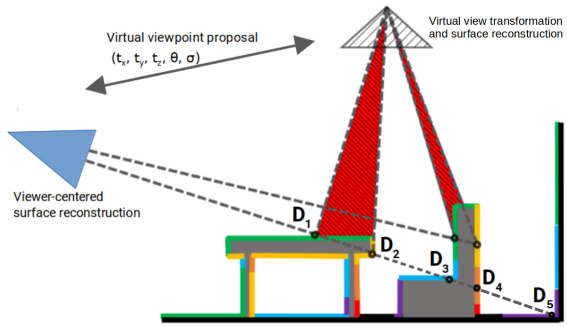
We use experiments to empirically determine a fixed number of layers that provides good coverage of typical natural scenes, while remaining compact enough for efficient learning and prediction. Another challenge is that surfaces that are nearly tangent to input camera rays are not well represented by a depth map of fixed resolution. To address this, we introduce an additional virtual view where tangent surfaces are sampled more densely (see Section 4).

3.1. Multi-Layer Depth Maps from 3D Geometry

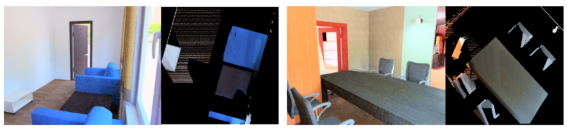
In our experiments, we focus on a five-layer model designed to represent key features of 3D scene geometry for

$\bar{D}_1$	$\bar{D}_{1,2}$	$\bar{D}_{1,2,3}$	$\bar{D}_{1..4}$	$\bar{D}_{1..5}$	$\bar{D}_{1..5}$ +Ovh.
0.237	0.427	0.450	0.480	0.924	0.932

Table 1: Scene surface coverage (recall) of ground truth depth layers with a 5cm threshold. Our predictions cover 93% of the scene geometry inside the viewing frustum.



(a) **3D volume inference** through multi-layer depth images



(b) **Input** image and **transformed** color features using $\bar{D}_1$ and $\bar{D}_2$.

Figure 3: Epipolar transfer of features from the input image to a virtual overhead view. Given multi-layer depth predictions of surface entrances and exits, each pixel in the input view is mapped to zero, one, or two segments of the corresponding epipolar line in the virtual view.

typical indoor scenes. To capture the overall room layout, we model the room envelope (floors, walls, ceiling, windows) that defines the extent of the space. We define the depth D_5 of these surfaces to be the *last* layer of the scene.

To model the shapes of observed objects, we trace rays from the input view and record the first intersection with a visible surface in depth map D_1. This resembles a standard depth map, but excludes the room envelope. If we continue along the same ray, it will eventually exit the object at a depth we denote by D_2. For non-convex objects the ray may intersect the same object multiple times, but we only record the *last* exit in D_2. As many indoor objects have large convex parts, the D_1 and D_2 layers are often sufficient to accurately reconstruct a large proportion of foreground objects

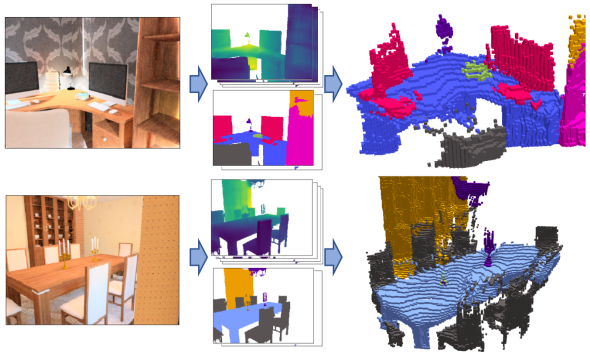


Figure 4: A volumetric visualization of our predicted multi-layer surfaces and semantic labels on SUNCG. We project the center of each voxel into the input camera, and the voxel is marked occupied if the depth value falls in the first object interval (D_1, D_2) or the occluded object interval (D_3, D_4).

in real scenes. While room envelopes typically have a very simple shape, the prediction of occluded structure behind foreground objects is more challenging. We define layer $D_3 > D_2$ as the depth of the next object intersection, and D_4 as the depth of the exit from that second object instance.

We let $(\bar{D}_1, \bar{D}_2, \bar{D}_3, \bar{D}_4, \bar{D}_5)$ denote the ground truth multi-layer depth maps derived from a complete 3D model. Since not all viewing rays intersect the same number of objects (e.g., when the room envelope is directly visible), we define a binary mask $\bar{M}_\ell$ which indicates the pixels where layer ℓ has support. Note that $\bar{M}_1 = \bar{M}_2$, and $\bar{M}_3 = \bar{M}_4$, since D_2 (first instance exit) has the same support as D_1. Experiments in Section 5 evaluate the relative importance of different layers in modeling realistic 3D scenes.

3.2. Predicting Multi-Layer Depth Maps

To learn to predict five-channel multi-layer depths $\mathcal{D} = (D_1, D_2, D_3, D_4, D_5)$ from images, we train a standard encoder-decoder network with skip connections, and use the Huber loss $\rho_h(.,.)$ to measure prediction errors:

$$L_d(\mathcal{D}) = \sum_{\ell=1}^{5} \left(\frac{\bar{M}_\ell}{||\bar{M}_\ell||_1} \right) \cdot \rho_h(D_\ell, \bar{D}_\ell). \quad (1)$$

We also predict semantic segmentation masks for the first and third layers. The structure of the semantic segmentation network is similar to the multi-layer depth prediction network, except that the output has 80 channels (40 object categories in each of two layers), and errors are measured via the cross-entropy loss. To reconstruct complete 3D geometry from multi-layer depth predictions, we use predicted masks and depths to generate meshes corresponding to the front and back surfaces of visible and partially occluded objects, as well as the room envelope. Without the back surfaces [34], ground truth depth layers $\bar{D}_{1,3,5}$ cover only 82% of the scene geometry inside the viewing frustum (vs. 92% including back surfaces of objects, see Table 1).

4. Epipolar Feature Transformer Networks

To allow for richer view-based scene understanding, we would like to relate features visible in the input view to feature representations in other views. To achieve this, we transfer features computed in input image coordinates to the coordinate system of a "virtual camera" placed elsewhere in the scene. This approach more efficiently covers some parts of 3D scenes than single-view, multi-layer depths.

Figure 2 shows a block diagram of our *Epipolar Feature Transformer* (EFT) network. Given features F extracted from the image, we choose a virtual camera location with transformation mapping T and transfer weights W, and use these to warp F to create a new "virtual view" feature map G. Like *spatial transformer networks* (STNs) [19] we perform a parametric, differentiable "warping" of a feature

map. However, EFTs incorporate a weighted pooling operation informed by multi-view geometry.

Epipolar Feature Mapping. Image features at spatial location (s, t) in an input view correspond to information about the scene which lies somewhere along the ray

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = z\mathbf{K_I}^{-1} \begin{bmatrix} s \\ t \\ 1 \end{bmatrix}, \qquad z \geq 0,$$

where $\mathbf{K}_I \in \mathbb{R}^{3\times3}$ encodes the input camera intrinsic parameters, as well as the spatial resolution and offset of the feature map. z is the depth along the viewing ray, whose image in a virtual orthographic camera is given by

$$\begin{bmatrix} u(s,t,z) \\ v(s,t,z) \end{bmatrix} = \mathbf{K_V} \left(z\mathbf{R}\mathbf{K_I}^{-1} \begin{bmatrix} s \\ t \\ 1 \end{bmatrix} + \mathbf{t} \right), \qquad z \geq 0.$$

Here $\mathbf{K}_V \in \mathbb{R}^{2\times3}$ encodes the virtual view resolution and offset, and $\mathbf{R}$ and $\mathbf{t}$ the relative pose.[1] Let $T(s,t,z) = (u(s,t,z), v(s,t,z))$ denote the forward mapping from points along the ray into the virtual camera, and $\Omega(u,v) = \{(s,t,z) : T(s,t,z) = (u,v)\}$ be the pre-image of (u,v).

Given a feature map computed from the input view $F(s,t,f)$, where f indexes the feature dimension, we synthesize a new feature map G corresponding to the virtual view. We consider general mappings of the form

$$G(u,v,f) = \frac{\sum_{(s,t,z)\in\Omega(u,v)} F(s,t,f)W(s,t,z)}{\sum_{(s,t,z)\in\Omega(u,v)} W(s,t,z)},$$

where $W(s,t,z) \geq 0$ is a gating function that may depend on features of the input image.[2] When $\Omega(u,v)$ is empty, we set $G(u,v,f) = 0$ for points (u,v) outside the viewing frustum of the input camera, and otherwise interpolate feature values from those of neighboring virtual-view pixels.

Choice of the Gating Function W. By design, the transformed features are differentiable with respect to F and W. Thus in general we could assign a loss to predictions from the virtual camera, and learn an arbitrary gating function W from training data. However, we instead propose to leverage additional geometric structure based on predictions about the scene geometry produced by the frontal view.

Suppose we have a scene depth estimate $D_1(s,t)$ at every location in the input view. To simplify occlusion reasoning we assume that relative to the input camera view, the virtual camera is rotated around the x-axis by $\theta < 90°$ and translated in y and z to sit above the scene so that points

[1] For a perspective model the righthand side is scaled by $z'(s,t,z)$, the depth from the virtual camera of the point at location z along the ray.

[2] For notational simplicity, we have written G as a sum over a discrete set of samples Ω. To make G differentiable with respect to the virtual camera parameters, we perform bilinear interpolation.

Figure 5: Single image scene reconstruction via multi-layer depth maps. Estimates of the front (green) and back (cyan) surfaces of objects, as seen from the input view, are complemented by heights estimated by a virtual overhead camera (dark green) via our epipolar feature transform. Room envelope estimates are rendered in gray.

which project to larger t in the input view have larger depth in the virtual view. Setting the gating function as

$$W^1_{\text{surf}}(s,t,z) = \delta[D_1(s,t) = z] \prod_{\hat{t}=0}^{t-1} \delta[D_1(s,\hat{t}) + (t-\hat{t})\cos\theta \neq z]$$

yields an epipolar feature transform that *re-projects* each feature at input location (s,t) into the overhead view via the depth estimate D_1, but only in cases where it is not occluded by a patch of surface higher up in the scene. In our experiments we compute W^ℓ_{surf} for each D_ℓ, $\ell \in \{1,2,3,4\}$, and use $W_{\text{surf}} = \max_\ell W^\ell_{\text{surf}}$ to transfer input view features to both visible and occluded surfaces in the overhead feature map. We implement this transformation using a z-buffering approach by traversing the input feature map from bottom to top, and overwriting cells in the overhead feature map.

Figure 3(b) illustrates this feature mapping applied to color features using the ground-truth depth map for a scene. In some sense, this surface-based reprojection is quite conservative because it leaves holes in the interior of objects (e.g., the interior of the orange wood cabinet). If the frontal view network features at a given spatial location encode the presence, shape, and pose of some object, then those features really describe a whole volume of the scene behind the object surface. It may thus be preferable to instead transfer the input view features to the entire expected volume in the overhead representation.

To achieve this, we use the multi-layer depth representation predicted by the frontal view to define a range of scene

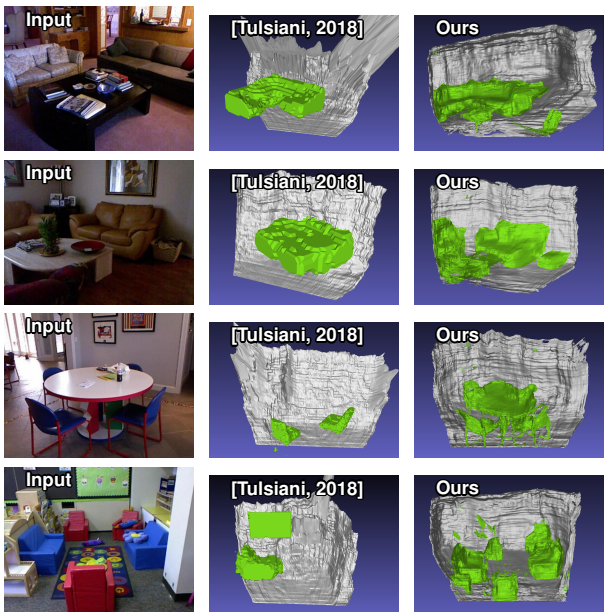


Figure 6: Evaluation of 3D reconstruction on the NYUv2 [26] dataset. Tulsiani *et al.* [44] are sensitive to the performance of 2D object detectors, and their voxelized output is a coarse approximation of the true 3D geometry.

depths to which the input view feature should be mapped. If $D_1(s,t)$ is the depth of the front surface and $D_2(s,t)$ is the depth at which the ray exits the back surface of an object instance, we define a volume-based gating function:

$$W_{\text{vol}}(s,t,z) = \delta[z \in (D_1(s,t), D_2(s,t))].$$

As illustrated in Figure 3(a), volume-based gating copies features from the input view to entire segments of the epipolar line in the virtual view. In our experiments we use this gating to generate features for (D_1, D_2) and concatenate them with a feature map generated using (D_3, D_4).

Overhead Viewpoint Generation. For cluttered indoor scenes, there may be many overlapping objects in the input view. Overhead views of such scenes typically have much less occlusion and should be simpler to reason about geometrically. We thus select a virtual camera that is roughly overhead and covers the scene content visible from the reference view. We assume the input view is always taken with the gravity direction in the y, z plane. We parameterize the overhead camera relative to the reference view by a translation (t_x, t_y, t_z) which centers it over the scene at fixed height above the floor, a rotation θ which aligns the overhead camera to the gravity direction, and a scale σ that captures the radius of the orthographic camera frustum.

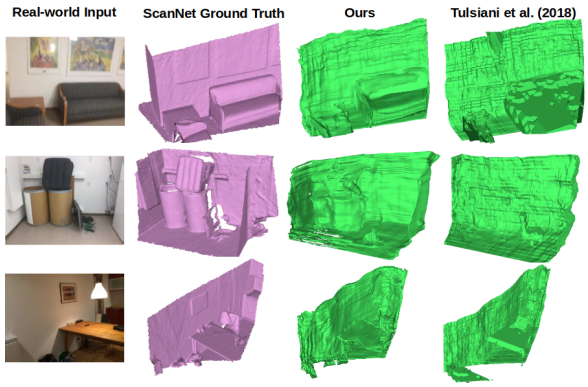
5. Experiments

Because we model complete descriptions of the ground-truth 3D geometry corresponding to RGB images, which is not readily available for natural images, we learn to predict multi-layer and multi-view depths from physical renderings of indoor scenes [56] provided by the SUNCG dataset [40].

5.1. Generation of Training Data

The SUNCG dataset [40] contains complete 3D meshes for 41,490 houses that we render to generate our training data. For each rendered training image, we extract the subset of the house model that is relevant to the image content, without making assumptions about the room size. We transform the house mesh to the camera's coordinate system and truncate polygons that are outside the left, top, right, bottom, and near planes of the perspective viewing frustum. Objects that are projected behind the depth image of the room envelope are also removed. The final ground truth mesh that we evaluate against contains all polygons from the remaining objects, as well as the true room envelope.

For each rendered training image, we generate target multi-depth maps and segmentation masks by performing multi-hit ray tracing on the ground-truth geometry. We similarly compute ground-truth height maps for a virtual orthographic camera centered over each scene. To select an overhead camera viewpoint for training that covers the relevant scene content, we consider three heuristics: (i) Convert the true depth map to a point cloud, center the overhead camera over the mean of these points, and set the camera radius to 1.5 times their standard deviation; (ii) Center the overhead camera so that its principal axis lies in the same plane as the input camera, and offset in front of the input view by the mean of the room envelope depths; (iii) Select a square bounding box in the overhead view that encloses



	Precision	Recall
$D_{1,2,3,4,5}$ & Overhead	**0.221**	**0.358**
Tulsiani *et al.* [44]	0.132	0.191

Table 2: We quantitatively evaluate the synthetic-to-real transfer of 3D geometry prediction on the ScanNet [4] dataset (threshold of 10cm). We measure recovery of true object surfaces and room layouts within the viewing frustum.

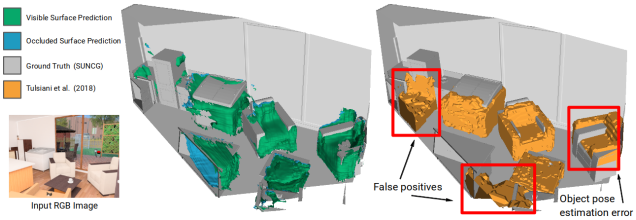


Figure 7: Qualitative comparison of our viewer-centered, end-to-end scene surface prediction (left) and the object-based detection and voxel shape prediction of [44] (right). Object-based reconstruction is sensitive to detection and pose estimation errors, while our method is more robust.

all points belonging to objects visible from the input view. None of these heuristics worked perfectly for all training examples, so we compute our final overhead camera view via a weighted average of these three candidates.

5.2. Model Architecture and Training

As illustrated in Figure 2, given an RGB image, we first predict a multi-layer depth map as well as a 2D semantic segmentation map. Features used to predict multi-layer depths are then mapped through our EFT network to synthesize features for a virtual camera view, and predict an orthographic height map. We then use the multi-layer depth map, semantic segmentation map, and overhead height map to predict a dense 3D reconstruction of the imaged scene.

We predict multi-layer depth maps and semantic segmentations via a standard convolutional encoder-decoder with skip connections. The network uses dilated convolution and has separate output branches for predicting each depth layer using the Huber loss specified in Section 3.2. For segmentation, we train a single branch network using a softmax loss to predict 40 semantic categories derived from the SUNCG mesh labels (see supplement for details).

Our overhead height map prediction network takes as input the transformed features of our input view multi-layer depth map. The overhead model integrates 232 channels (see Figure 2) including epipolar transformations of a 48-channel feature map from the depth prediction network, a 64-channel feature map from the semantic segmentation network, and the RGB input image. These feature maps are extracted from the frontal networks just prior to the predictive branches. Other inputs include a "best guess" overhead height map derived from frontal depth predictions, and a mask indicating the support of the input camera frustum. The frustum mask can be computed by applying the epipolar transform with $F = 1, W = 1$. The best-guess overhead depth map can be computed by applying an unnormalized gating function $W(s,t,z) = z \cdot \delta[D_1(s,t) = z]$ to the y-coordinate feature $F(s,t) = t$.

We also train a model to predict the virtual camera parameters which takes as input feature maps from our multi-

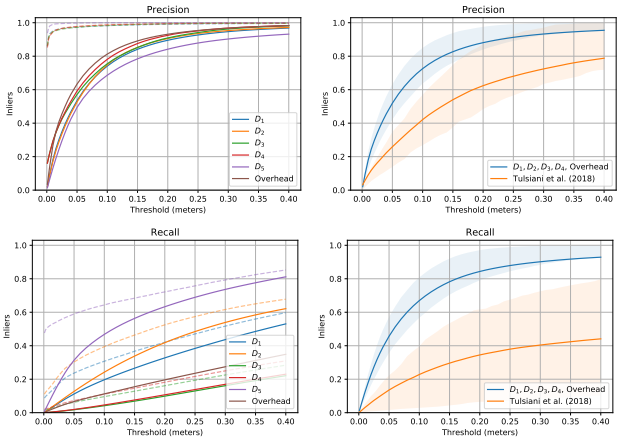


Figure 8: Precision and recall of scene geometry as a function of match distance threshold. *Left:* Reconstruction quality for different model layers. Dashed lines are the performance bounds provided by ground-truth depth layers $(\bar{D}_1, \bar{D}_2, \bar{D}_3, \bar{D}_4, \bar{D}_5)$. *Right:* Accuracy of our model relative to the state-of-the-art, evaluated against objects only. The upper and lower band indicate 75th and 25th quantiles. The higher variance of Tulsiani *et al.* [44] may be explained in part by the sensitivity of the model to having the correct initial set of object detections and pose estimates.

depth prediction network, and attempts to predict the target overhead viewpoint (orthographic translation (t_x, t_y) and frustum radius σ) chosen as in Section 5.1. While the EFT network is differentiable and our final model can in principle be trained end-to-end, in our experiments we simply train the frontal model to convergence, freeze it, and then train the overhead model on transformed features without backpropagating overhead loss back into the frontal-view model parameters. We use the Adam optimizer to train all of our models with batch size 24 and learning rate 0.0005 for 40 epochs. The Physically-based Rendering [56] dataset uses a fixed downward tilt camera angle of 11 degrees, so we do not need to predict the gravity angle. At test time, the height of the virtual camera is the same as the input frontal camera and assumed to be known. We show qualitative 3D reconstruction results on the SUNCG test set in Figure 5.

5.3. Evaluation

To reconstruct 3D surfaces from predicted multi-layer depth images as well as the overhead height map, we first convert the depth images and height maps into a point cloud and triangulate vertices that correspond to a 2×2 neighborhood in image space. If the depth values of two adjacent pixels is greater than a threshold $\delta \cdot a$, where δ is the footprint of the pixel in camera coordinates and $a = 7$, we do not create an edge between those vertices. We do not predict the room envelope from the virtual overhead view, so only pixels with height values higher than 5 cm above the floor are considered for reconstruction and evaluation.

	Precision	Recall
D_1	0.525	0.212
D_1 & Overhead	**0.553**	**0.275**
$D_{1,2,3,4}$	0.499	0.417
$D_{1,2,3,4}$ & Overhead	**0.519**	**0.457**

Table 3: Augmenting the frontal depth prediction with the predicted virtual view height map improves both precision and recall (match threshold of 5cm).

Metrics. We use precision and recall of surface area as the metric to evaluate how closely the predicted meshes align with the ground truth meshes, which is the native format for SUNCG and ScanNet. Coverage is determined as follows: We uniformly sample points on surface of the ground truth mesh then compute the distance to the closest point on the predicted mesh. We use sampling density $\rho = 10000/\text{meter}^2$ throughout our experiments. Then we measure the percentage of inlier distances for given a threshold. *Recall* is the coverage of the ground truth mesh by the predicted mesh. Conversely, *precision* is the coverage of the predicted mesh by the ground truth mesh.

3D Scene Surface Reconstruction. To provide an upper-bound on the performance of our multi-layer depth representation, we evaluate how well surfaces reconstructed from ground-truth depths cover the full 3D mesh. This allows us to characterize the benefits of adding additional layers to the representation. Table 1 reports the coverage (recall) of the ground-truth at a threshold of 5cm. The left panels of Figure 8 show a breakdown of the precision and recall for the individual layers of our model predictions along with the upper bounds achievable across a range of inlier thresholds.

Since the room envelope is a large component of many scenes, we also analyze performance for objects (excluding the envelope). Results summarized in Table 3 show that the addition of multiple depth layers significantly increases recall with only a small drop in precision, and the addition of overhead EFT predictions boosts both precision and recall.

Ablation Study on Transformed Features. To further demonstrate the value of the EFT module, we evaluate the accuracy of the overhead height map prediction while incrementally excluding features. We first exclude channels that correspond to the semantic segmentation network features and compare the relative pixel-level L1 error. We then exclude features from the depth prediction network, using only RGB, frustum mask and best guess depth image. This baseline corresponds to taking the prediction of the input view model as an RGB-D image and re-rendering it from the virtual camera viewpoint. The L1 error increases respectively from 0.132 to 0.141 and 0.144, which show that applying the EFT to the whole CNN feature map outperforms simple geometric transfer.

Comparison to the State-of-the-art. Finally, we compare the scene reconstruction performance of our end-to-end approach with the object-based Factored3D [44] method using their pre-trained weights, and converting voxel outputs to surface meshes using marching cubes. We evaluated on 3960 examples from the SUNCG test set and compute precision and recall on objects surfaces (excluding envelope). As Figure 8 shows, our method yields roughly 3x improvement in recall and 2x increase in precision, providing estimates which are both more complete and more accurate. Figure 7 highlights some qualitative differences. To evaluate with an alternative metric, we voxelized scenes at 2.5cm^3 resolution (shown in Figure 4). Using the voxel intersection-over-union metric, we see significant performance improvements over Tulsiani *et al.* [46] (0.102 to 0.310) on objects (see supplement for details).

Reconstruction on Real-world Images. Our network model is trained entirely on synthetically generated images [56]. We test the ability of the model to generalize to the NYUv2 dataset [26] via the promising comparison to Tulsiani *et al.* [44] in Figure 6.

We additionally test our model on images from the ScanNetv2 dataset [4]. The dataset contains RGB-D image sequences taken in indoor scenes, as well as 3D reconstructions produced by BundleFusion [5]. For each video sequence from the 100 test scenes, we randomly sample 5% of frames, and manually select 1000 RGB images to compare our algorithm to Tulsiani *et al.* [44]. We select images where the pose of the camera is almost perpendicular to the gravity orientation, the amount of motion blur is small, and the image does not depict a close-up view of a single object. We treat the provided 3D reconstructions within each viewing frustum as ground truth annotations. As summarized in Table 2, our approach has significantly improved precision and recall to Tulsiani *et al.* [44].

6. Conclusion

Our novel integration of deep learning and perspective geometry enables complete 3D scene reconstruction from a single RGB image. We estimate multi-layer depth maps which model the front and back surfaces of multiple objects as seen from the input camera, as well as the room envelope. Our epipolar feature transformer network geometrically transfers input CNN features to estimate scene geometry from virtual viewpoints, providing more complete coverage of real-world environments. Experimental results on the SUNCG dataset [40] demonstrate the effectiveness of our model. We also compare with prior work that predicts voxel representations of scenes, and demonstrate the significant promise of our multi-view and multi-layer depth representations for complete 3D scene reconstruction.

Acknowledgements. This research was supported by NSF grants IIS-1618806, IIS-1253538, CNS-1730158, and a hardware donation from NVIDIA.

References

[1] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 3

[2] Weifeng Chen, Zhao Fu, Dawei Yang, and Jia Deng. Single-image depth perception in the wild. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 730–738, 2016. 3

[3] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2016. 3

[4] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3, 6, 8

[5] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics*, 36(4):76a, 2017. 8

[6] Angela Dai, Charles Ruizhongtai Qi, and Matthias Nießner. Shape completion using 3d-encoder-predictor cnns and shape synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6545–6554, 2017. 3

[7] Helisa Dhamo, Keisuke Tateno, Iro Laina, Nassir Navab, and Federico Tombari. Peeking behind objects: Layered depth prediction from a single image. *Pattern Recognition Letters*, 2019. 3

[8] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2366–2374, 2014. 1, 3

[9] Haoqiang Fan, Hao Su, and Leonidas Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2463–2471, 2017. 3

[10] Valentin Gabeur, Jean-Sébastien Franco, Xavier Martin, Cordelia Schmid, and Gregory Rogez. Moulding humans: Non-parametric 3d human shape estimation from single images. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 3

[11] Matheus Gadelha, Rui Wang, and Subhransu Maji. Multiresolution tree networks for 3d point cloud processing. In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2018. 2

[12] Soumya Ghosh and Erik B Sudderth. Nonparametric learning for layered segmentation of natural images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2272–2279, 2012. 3

[13] Rohit Girdhar, David F Fouhey, Mikel Rodriguez, and Abhinav Gupta. Learning a predictable and generative vector representation for objects. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 484–499. Springer, 2016. 3

[14] Saurabh Gupta, Pablo Arbeláez, Ross Girshick, and Jitendra Malik. Aligning 3d models to rgb-d images of cluttered scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015. 3

[15] Christian Häne, Shubham Tulsiani, and Jitendra Malik. Hierarchical surface prediction for 3d object reconstruction. In *IEEE International Conference on 3D Vision (3DV)*, pages 412–420, 2017. 3

[16] Phillip Isola and Ce Liu. Scene collaging: Analysis and synthesis of natural images with semantic layers. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 3048–3055, 2013. 3

[17] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568. ACM, 2011. 3

[18] Hamid Izadinia, Qi Shan, and Steven M Seitz. Im2cad. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[19] Max Jaderberg, Karen Simonyan, Andrew Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015. 4

[20] Dinghuang Ji, Junghyun Kwon, Max McFarland, and Silvio Savarese. Deep view morphing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[21] Huaizu Jiang, Erik Learned-Miller, Gustav Larsson, Michael Maire, and Greg Shakhnarovich. Self-supervised depth learning for urban scene understanding. *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 3

[22] Abhishek Kar, Shubham Tulsiani, Joao Carreira, and Jitendra Malik. Category-specific object reconstruction from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1966–1974, 2015. 3

[23] Hiroharu Kato, Yoshitaka Ushiku, Tatsuya Harada, Andrew Shin, Leopold Crestel, Hiroharu Kato, Kuniaki Saito, Katsunori Ohnishi, Masataka Yamaguchi, Masahiro Nakawaki, et al. Neural 3d mesh renderer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[24] Chen-Hsuan Lin, Chen Kong, and Simon Lucey. Learning efficient point cloud generation for dense 3d object reconstruction. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018. 3

[25] Daniel Maturana and Sebastian Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 922–928, 2015. 3

[26] Pushmeet Kohli Nathan Silberman, Derek Hoiem and Rob Fergus. Indoor segmentation and support inference from rgbd images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012. 6, 8

[27] Andrea Nicastro, Ronald Clark, and Stefan Leutenegger. X-section: Cross-section prediction for enhanced rgbd fusion. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2019. 3

[28] Eunbyung Park, Jimei Yang, Ersin Yumer, Duygu Ceylan, and Alexander Berg. Transformation-grounded image generation network for novel 3d view synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017. 3

[29] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 2

[30] Stephan R Richter and Stefan Roth. Matryoshka networks: Predicting 3d geometry via nested shape layers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1936–1944, 2018. 3

[31] Gernot Riegler, Ali O Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6620–6629, 2017. 1

[32] Lawrence G Roberts. *Machine perception of three-dimensional solids*. PhD thesis, Massachusetts Institute of Technology, 1963. 3

[33] Jason Rock, Tanmay Gupta, Justin Thorsen, JunYoung Gwak, Daeyun Shin, and Derek Hoiem. Completing 3d object shape from one depth image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2484–2493, 2015. 3

[34] Jonathan Shade, Steven Gortler, Li-wei He, and Richard Szeliski. Layered depth images. 1998. 4

[35] Daeyun Shin, Charless C. Fowlkes, and Derek Hoiem. Pixels, voxels, and views: A study of shape representations for single view 3d object shape prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 3

[36] Edward Smith, Scott Fujimoto, and David Meger. Multi-view silhouette and depth decomposition for high resolution 3d object representation. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 6479–6489, 2018. 3

[37] Edward J. Smith and David Meger. Improved adversarial systems for 3d object generation and reconstruction. In *1st Annual Conference on Robot Learning, CoRL 2017, Mountain View, California, USA, November 13-15, 2017, Proceedings*, pages 87–96, 2017. 3

[38] Paul Smith, Tom Drummond, and Roberto Cipolla. Layered motion segmentation and depth ordering by tracking edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(4):479–494, 2004. 3

[39] Noah Snavely, Steven M Seitz, and Richard Szeliski. Modeling the world from internet photo collections. *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 80(2):189–210, 2008. 3

[40] Shuran Song, Fisher Yu, Andy Zeng, Angel X Chang, Manolis Savva, and Thomas Funkhouser. Semantic scene completion from a single depth image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 1, 2, 3, 6, 8

[41] Hang Su, Varun Jampani, Deqing Sun, Subhransu Maji, Evangelos Kalogerakis, Ming-Hsuan Yang, and Jan Kautz. SPLATNet: Sparse lattice networks for point cloud processing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2530–2539, 2018. 2

[42] Hao Su, Fan Wang, Li Yi, and Leonidas Guibas. 3d-assisted image feature synthesis for novel views of an object. *arXiv preprint arXiv:1412.0003*, 2014. 3

[43] Deqing Sun, Erik B Sudderth, and Michael J Black. Layered segmentation and optical flow estimation over time. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1768–1775, 2012. 3

[44] Shubham Tulsiani, Saurabh Gupta, David Fouhey, Alexei A Efros, and Jitendra Malik. Factoring shape, pose, and layout from the 2d image of a 3d scene. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 2, 3, 6, 7, 8

[45] Shubham Tulsiani, Richard Tucker, and Noah Snavely. Layer-structured 3d scene inference via view synthesis. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 3

[46] Shubham Tulsiani, Tinghui Zhou, Alexei A Efros, and Jitendra Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3, 8

[47] J. Y. A. Wang and E. H. Adelson. Representing moving images with layers. *IEEE Transactions on Image Processing (TIP)*, 3(5):625–638, Sept. 1994. 3

[48] Nanyang Wang, Yinda Zhang, Zhuwen Li, Yanwei Fu, Wei Liu, and Yu-Gang Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 2, 3

[49] Zhe Wang, Liyan Chen, Shaurya Rathore, Daeyun Shin, and Charless Fowlkes. Geometric pose affordance: 3d human pose with scene constraints. *arXiv preprint arXiv:1905.07718*, 2019. 3

[50] Jiajun Wu, Yifan Wang, Tianfan Xue, Xingyuan Sun, Bill Freeman, and Josh Tenenbaum. Marrnet: 3d shape reconstruction via 2.5 d sketches. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 540–550, 2017. 3

[51] Jiajun Wu, Chengkai Zhang, Xiuming Zhang, Zhoutong Zhang, William T Freeman, and Joshua B Tenenbaum. Learning 3D Shape Priors for Shape Completion and Reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 3

[52] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3D ShapeNets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1912–1920, 2015. 3

[53] Xinchen Yan, Jimei Yang, Ersin Yumer, Yijie Guo, and Honglak Lee. Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 1696–1704, 2016. 3

[54] Bo Yang, Stefano Rosa, Andrew Markham, Niki Trigoni, and Hongkai Wen. 3d object dense reconstruction from a single depth view. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2018. 3

[55] Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Josh Tenenbaum, Bill Freeman, and Jiajun Wu. Learning to reconstruct shapes from unseen classes. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2263–2274, 2018. 3

[56] Yinda Zhang, Shuran Song, Ersin Yumer, Manolis Savva, Joon-Young Lee, Hailin Jin, and Thomas Funkhouser. Physically-based rendering for indoor scene understanding using convolutional neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 6, 7, 8

[57] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[58] Tinghui Zhou, Richard Tucker, John Flynn, Graham Fyffe, and Noah Snavely. Stereo magnification: Learning view synthesis using multiplane images. In *ACM Transactions on Graphics (SIGGRAPH)*, 2018. 3

[59] Chuhang Zou, Ruiqi Guo, Zhizhong Li, and Derek Hoiem. Complete 3d scene parsing from single rgbd image. *International Journal of Computer Vision (IJCV)*, 2018. 3

[60] Chuhang Zou, Ersin Yumer, Jimei Yang, Duygu Ceylan, and Derek Hoiem. 3d-prnn: Generating shape primitives with recurrent neural networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, pages 900–909, 2017. 3