Multi-layer Depth and Epipolar Feature Transformers for 3D Scene Reconstruction

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://www.ics.uci.edu/~daeyuns/layered-epipolar-cnn

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.

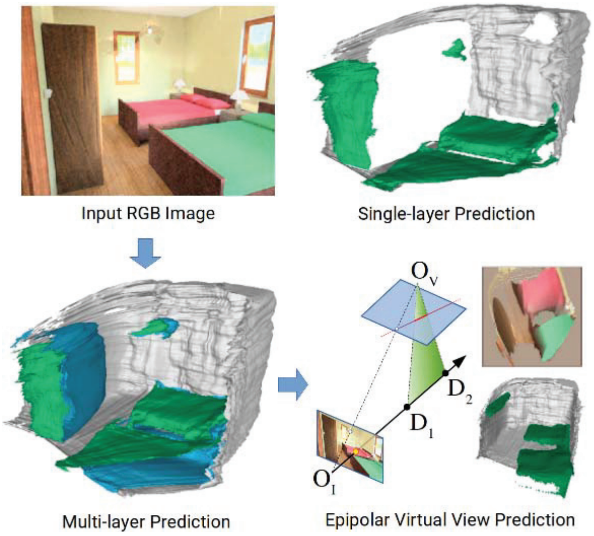


Figure 1: Given a single input view of a scene (top left), we would like to predict a complete geometric model. Depth maps (top right) provide an efficient representation of scene geometry but are incomplete, leaving large holes (e.g., the wardrobe). We propose multi-layer depth predictions (bottom left) 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 [7]. 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 [37] and use convolutions to perform translation-covariant computations in 3D. This provides a more com-

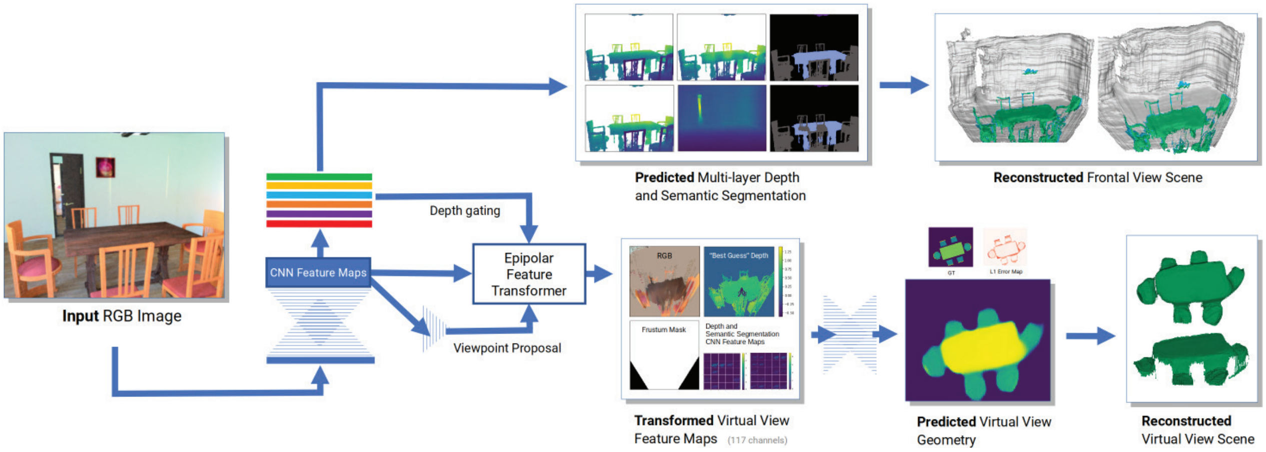


Figure 2: Block diagram of our system for reconstructing a complete scene from a single RGB input image. Our model first predicts a multi-layer depth map that encodes the depth of both front and back surfaces of objects in the scene from the input view. Given the extracted feature map and predicted multi-depth, the network generates overhead camera parameters and then transforms features from the input view into the overhead viewpoint where further inference and predictions occur in the overhead image coordinate system.

plete 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 octtrees [28]). Other approaches represent shape as an unstructured point cloud [26, 38], but require development of suitable convolutional operators [9, 45] 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 Figure 1, our approach uses an alternative shape representation that seeks to extend 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 [32], 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 [37]. 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 [41]. 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. Additionally, complicated scenes may involve many overlapping objects which require a large number of layers to constitute a complete representation. We address this 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 *Epipolar Feature Transformer* (EFT) networks that provide geometrically consistent transfer of CNN feature maps 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 geometric properties from 2D images has a rich history in computer vision, dating back to the visionary work of Roberts [29].

Monocular object shape prediction. The problem of single-view 3D shape reconstruction is challenging because the output space is under-constrained. Large-scale datasets like ShapeNet [1, 48] facilitate progress in this field, and recent methods learn geometric priors for object categories [20, 47], disentangle primitive shapes from objects [11, 56], or model surfaces [13, 32, 51]. Another line of work aims to complete the occluded geometric structure of objects from a 2.5D image or partial 3D scan [30, 5, 46, 50]. While the quality of such 3D object reconstructions continues to grow [21, 45], applications are limited by the assumption that input images depict a single, centered object.

3D scene reconstruction. We are interested in predicting the geometry of full scenes containing an unknown number of objects; this task is significantly more challenging than object reconstruction. Tulsiani *et al.* [41] 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.* [37] 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 [16, 55, 12], 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. To represent reconstructed 3D geometry, most recent methods use voxels [3, 37, 34, 43, 33]. This representation is easy to integrate with 3D CNNs [48] that seek to learn features for high-level recognition tasks [23]. Other methods [8, 22] use dense point clouds to represent 3D reconstructions. Classic 2.5D depth maps [7, 2] recover the geometry of visible scene features, but do not capture occlusion. Shin *et al.* [32] empirically compared these representations for object reconstruction. We focus on extending these ideas to whole scenes using a view-based multi-layer depth representation that encodes complete shape of multiple objects.

Learning layered representations. Layered representations [44] have proven useful for many computer vision tasks including segmentation [10] and optical flow prediction [40]. For 3D reconstruction, decomposing scenes into layers enables algorithms to reason about object occlusions and depth orderings [14, 35]. Layered 2.5D representations such as the two-layer decompositions of [42, 6] 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) as well as inferring depths from virtual viewpoints far from the input view to provide more complete full 3D scene geometry. Our use of layers is similar to [27], who used multiple intersection depths to model non-convexities for single object shape completion.

Multi-view synthesis. Many classic 3D reconstruction methods utilize multi-view inputs to synthesize 3D shapes [15, 36, 4]. Given monocular inputs, several recent methods explore ways of synthesizing object appearance or image features from novel viewpoints [54, 49, 18, 3, 25, 39]. Other work uses unsupervised learning from stereo or video inputs to reason about depths [53, 19]. We generalize the notion of transferring pixel colors associated with surface points between viewpoints to transferring whole CNN feature maps over corresponding object volumes, yielding more accurate and complete 3D reconstruction.

3. Modeling Scenes with Multi-Layer Depth Maps

Traditional depth maps record the depth at which a ray through a given pixel first intersects a surface in the scene. This so-called 2.5D representation of scene geometry can provide accurate descriptions of visible surfaces and is a natural fit for CNNs. However, it can't encode the shape of partially occluded objects or even the complete 3D shape of fully visible objects (due to self-occlusion). We propose to represent 3D geometry of a scene by recording multiple surface intersections each camera ray. As illustrated in Figure 4(a), some rays may intersect many distinct object surfaces, and thus require a large number of depth layers to capture in full detail. However, provided we have enough layers to handle multiple overlapping objects and non-convexities, multi-layer depth can provide a complete description of scene geometry.

Two interesting challenges that arise are: (1) choosing a fixed number of layers that achieves good coverage of typical scenes while still remaining compact to compute and learn, and (2) surfaces that are nearly tangent to input view rays are not well represented by a depth map of fixed resolution. We approach first challenge in an empirical fashion, specifying the set of layers and validating the choice experimentally (Section 5). The second challenge we address by

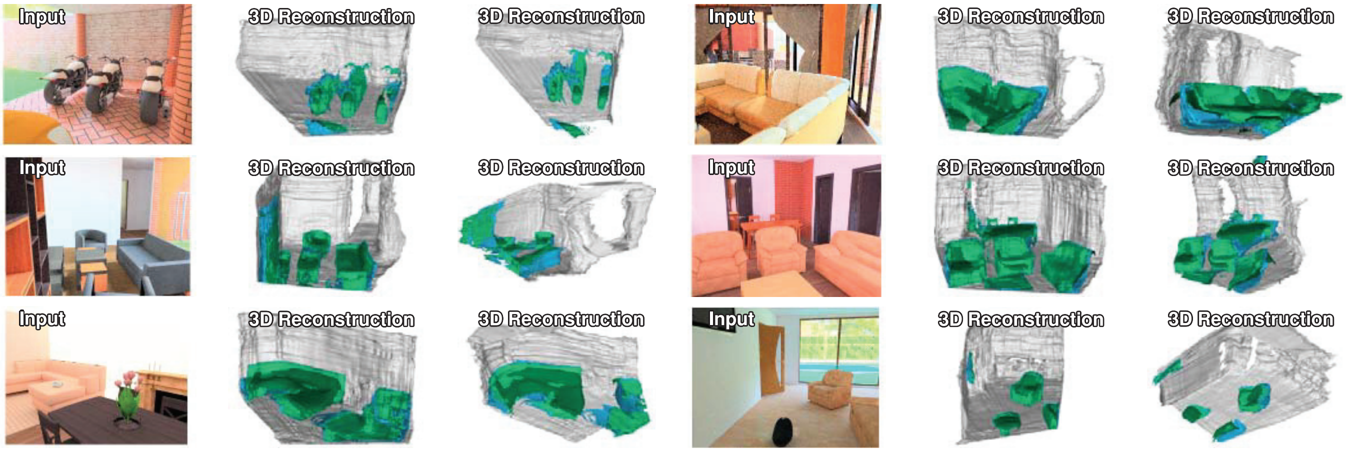


Figure 3: Single image scene completion results using multi-layer depth maps. The green and cyan layers represent the estimated front and completed back surfaces of objects respectively from the input view (see Section 3). The dark green layer corresponds to depth estimated from a virtual overhead camera viewpoint using epipolar transformed features (see Section 4). Gray is the predicted room envelope.

introducing an additional virtual view where tangent surfaces are sampled more densely (Section 4).

3.1. Multi-Layer Depth Maps from 3D Geometry

In our experiments, we focus on a four-layer model designed to represent key features of 3D scene geometry for typical indoor scenes. To capture the overall layout of indoor scenes, we start with the room envelope (floors,walls,ceiling,windows) that defines the extent of the space. We define the depth of these surfaces to be the *last* layer of the scene; we denote the corresponding depth channel by D_4.

To specify the shape of objects within the room, we trace rays from the input view and record the first intersection with a visible surface which we store 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 which we record in depth map 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 accurately captured by front and back layers, at least when seen from typical viewing orientations, the D_1 and D_2 layers are often sufficient to accurately reconstruct all foreground objects in real indoor scenes.

Unlike the room envelope which typically has a very simple shape, the prediction of occluded structure behind foreground objects is a more challenging task. As a first step towards capturing this geometry, we define the third depth map D_3 to be the depth of the *last* intersection of each ray before hitting the room envelope. Importantly, this layer explicitly encodes the open space adjacent to walls and floors.

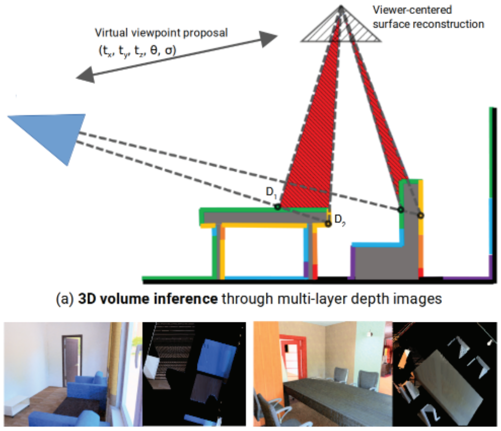
We let $(\bar{D}_1, \bar{D}_2, \bar{D}_3, \bar{D}_4)$ denote the ground truth multi-layer depth maps derived from a complete 3D model. Rays may not encounter all four of the intersections defined above: they may only hit one object, or the room envelope may be directly visible. We therefore let binary mask $\bar{M}_l$ indicate the pixels where layer l has support, which is defined as the segmentation of all foreground pixels. We only segment layers 1 and 3, because D_1 (first-hit) and D_2 (instance-exit) have the same segmentation due to symmetry. In Section 5, we provide experiments exploring the relative importance of different layers, and demonstrating the benefits of multi-layer representations of 3D scenes.

3.2. Predicting Multi-Layer Depth Maps

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

$$L_d(\mathcal{D}) = \sum_{\ell=1}^{4} \left(\frac{\bar{M}_\ell}{||\bar{M}_\ell||_1} \right) \cdot \rho_h(D_\ell, \bar{D}_\ell). \tag{1}$$

Because our pixel-wise multi-layer depth prediction is agnostic to high-level semantic information, we also predict semantic segmentation masks for the first and third layers, for use in subsequent 3D reconstruction algorithms. 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). The loss function is the cross-entropy. And we apply log-space transformation $\log_2{(\bar{D}_l + 0.5)}$ to all of our depth channels.



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

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

Figure 4: Diagram showing epipolar transfer to overhead view. A location in the input view is associated with a segment of the epipolar line in the virtual view determined by the multi-layer depth prediction of surface entrance and exit.

To reconstruct 3D geometry from multi-layer depth predictions, we first utilize D_1 and D_2 to construct frontal objects in regions that object instances are predicted to occupy. Occluded objects are similarly reconstructed via D_3 and its corresponding segmentation mask. Finally, we reconstruct the room layout using depth predictions from the final layer.

4. Epipolar Feature Transformer Networks

To allow for richer view-based inference about a scene, we would like to relate features visible in the input view to feature representations in other views. Specifically, we would like to transfer features computed in an input camera image coordinate system to the camera coordinate system of a "virtual camera" placed elsewhere in the scene. This makes it possible to overcome some of the limitations of a single-view multi-layer depth representation of geometry.

Figure 2 shows a block diagram of our approach which we term an Epipolar Feature Transformer network (EFT). Given features F extracted from the image, we will choose a virtual camera location, calculate transformation mapping T and transfer weights W and use these to "warp" F to create a new featuremap G corresponding to the (virtual) viewpoint. The overall structure is thus similar to spatial transformer networks (STNs) [17] in performing a parametric, differentiable "warping" of a feature map. However, our mapping incorporates a weighted pooling operation which is specific to 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 and z is the depth along the ray.

The image of this ray in a virtual orthographic camera is given by

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

where $\mathbf{K}_V \in \mathbb{R}^{2\times3}$ 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 would like to 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 $W \geq 0$ is a gating function that may depend on features of the input image. [2] When $\Omega(u,v)$ is empty, we can interpolate from neighboring feature values or set the value to 0 as appropriate (e.g., when (u,v) images points outside the viewing frustum of the input camera).

Choice of gating function W. By design, the transformed features are differentiable w.r.t. F and W so in general we can assign a loss to predictions from the virtual camera and learn an arbitrary gating function W from training data. However, we propose to leverage additional geometric structure based on predictions about the scene geometry produced by the frontal view.

Suppose we have an estimate of the scene depth map $D_1(s,t)$ at every location in the input view. For simplicity in reasoning about occlusion, let us assume that relative to the input camera view, the virtual camera is rotated around the x-axis by $\theta < 90$ degrees and translated in y and z to sit above the scene so that points which project to larger s in

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

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

the input view have larger depth in the virtual view. Setting the weighting function to

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

yields an epipolar feature transform that *re-projects* each feature at input location (s,t) into the overhead viewpoint based on the depth estimate D_1 whenever it is not occluded by a patch of surface higher up in the scene.

Figure 4 (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 since 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. Thus we propose that the input view features should instead be transferred over the whole expected volume in the overhead representation.

To achieve this, we can make use of our multi-layer depth representation predicted by the frontal view to specify a range of scene depths over which the input view feature applies. Suppose $D_1(s,t)$ 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 can define a volume-based gating function by

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

As illustrated in Figure 4 (a), this has the effect of taking a feature from the input view and copying it along a whole segment of the epipolar line in the virtual view.

Overhead viewpoint generation. For cluttered indoor scenes, there may be many overlapping objects in the input view. We posit that overhead orthographic views of such scenes should involve much less occlusion and be simpler to reason about geometrically. Thus, we would like to 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 yz-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 a 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

5.1. Dataset

To train our model requires complete descriptions of ground-truth geometry associated with a given input RGB image. Since such data is not readily available for natural images, we use the physically based renderings of indoor scene [52] based on the SUNCG dataset [37] as input and learn to predict our multi-layer depth representation, as well as epipolar feature transformations. The dataset contains 41490 houses and 2551 object models.

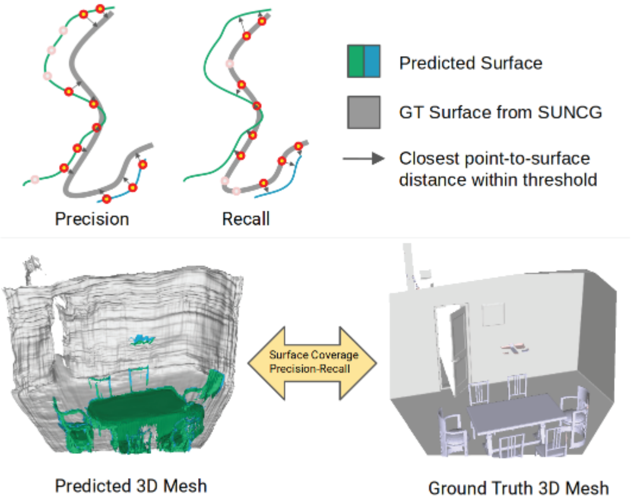


Figure 5: Illustration of our 3D precision-recall metrics. *Top*: We perform a bidirectional surface coverage evaluation on the reconstructed triangle meshes. *Bottom*: The ground truth mesh consists of all 3D surfaces within the field-of-view and in front of the room envelope. See Figure 2 for the corresponding input and output images.

Ground truth geometry. The SUNCG dataset [37] contains the complete 3D meshes of houses that we rendered to generate our training dataset. For each rendered RGB image, we need to extract a subset of the house model that are relevant to the scene. Our model does not make any assumptions about the size of the room, so we want to include all objects that share the same room envelope and inside the viewing frustum. We first 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. We then keep all the remaining meshes within the field-of-view. The final ground truth mesh (Figure 5) that we evaluate against consists of polygons from the remaining objects and the mesh of the ground truth room depth image.

Training data generation. We generate training data corresponding to the camera parameters for each rendered view in the SUNCG dataset. For each input RGB view, 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 overhead height

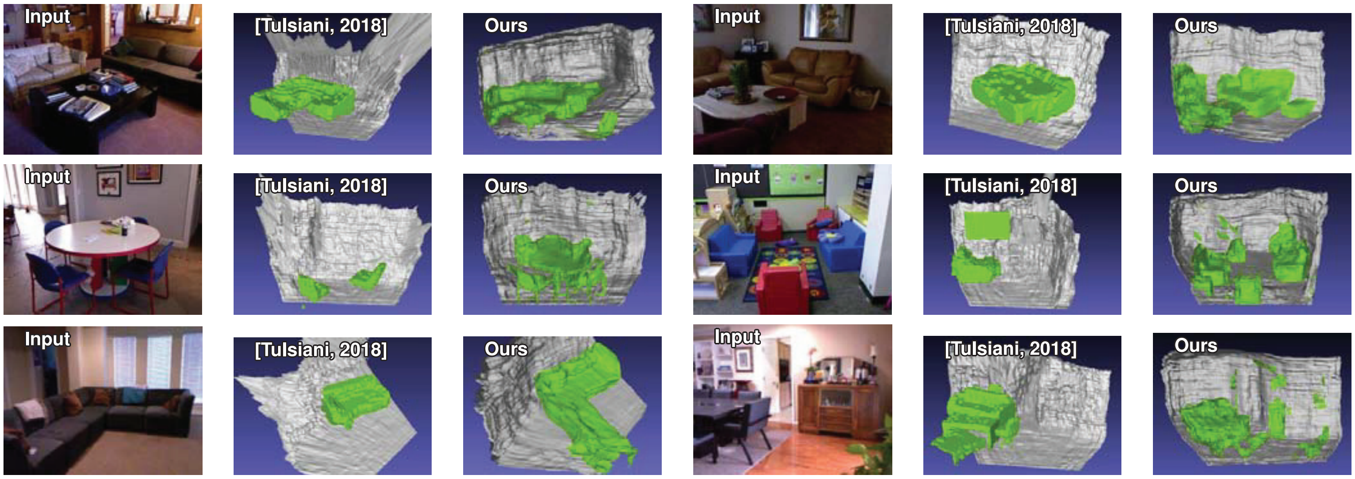


Figure 6: Evaluating 3D reconstruction on the NYUv2 dataset [24]. Tulsiani *et al.* [41] rely heavily on the performance of object detection, and the voxelized output is a coarse representation of the reconstructed geometry. Green region is the detected objects.

maps corresponding to a virtual orthographic camera centered over each scene.

To select an overhead camera viewpoint that covers the relevant area visible from the input, we considered three possible heuristics. (1) Convert the ground-truth frontal depth to a point cloud and center the overhead camera over the mean of the pointcloud with a radius set to 1.5x the pointcloud standard deviation. (2) Center the overhead camera so that its principal axis lies in the same plane as the input camera view and offset in front of the input view by the mean of the room envelope depth values. (3) Select a square bounding box in the overhead view that encloses all points belonging to objects visible from the input view. We found that none of these heuristics worked perfectly for all training examples so took a weighted average of the three candidates as our final overhead camera target for each scene.

5.2. Model architecture and training

As a recap, we provide an overview of our system. Given a RGB image, we first predict a multi-layer depth map as well as a 2D semantic segmentation map. After that, we take the intermediate feature before predicting the multi-layer depth map as input, and predict a camera transformer. Then we apply EFT and synthesize a virtual view feature using the camera transformer, and predict an orthographic height map. We then use the pixel-wise multi-layer depth, semantic segmentation and predicted height map to reconstruct a dense 3D reconstruction of the scene.

For predicting multi-layer depth maps and segmentations from the input RGB image, we use 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 described in Section 3.2. For segmentation, we train a single branch network using standard softmax loss to predict 40 semantic categories derived from the SUNCG mesh labels (see Appendix for details).

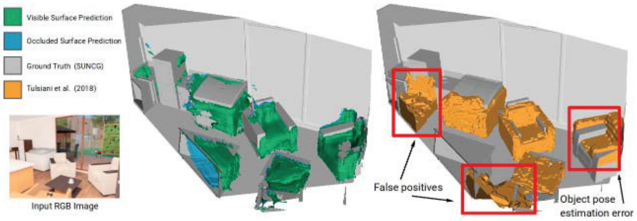

🟩 Visible Surface Prediction
🟦 Occluded Surface Prediction
⬜ Ground Truth (SUNCG)
🟧 Tulsiani et al. (2018)
Input RGB Image
False positives
Object pose estimation error

Figure 7: Comparison with the state-of-the-art approach. *Left*: Our viewer-centered, end-to-end scene surface prediction. *Right*: Object-based detection and voxel shape prediction (Tulsiani *et al.* [41], 2018). Their method is prone to detection and pose estimation error, while our method can under- or overestimate distances along the viewing direction.

Our overhead height map prediction network takes as input the transformed features of our input view multi-depth model. The overhead model input consists of 117 channels (as in Figure 2) which include transformed versions: (1) a 48 channel feature map from the depth prediction network, (2) a 64 channel feature map from the semantic segmentation network and (3) the RGB input image. These feature maps are extracted from the frontal networks just prior to the predictive branches. In addition, we consider a "best guess" overhead depth map based on the frontal depth prediction and a single channel mask that depicts the in-

$\bar{D}_1$	$\bar{D}_{1,2}$	$\bar{D}_{1,2,3}$	$\bar{D}_{1..4}$	$\bar{D}_{1..4}$ +Ovh.
0.243	0.448	0.503	0.906	0.916

Table 1: Scene surface coverage (recall) of ground-truth depth layers at threshold 0.05 (5cm). Our final representation covers 91 % of the scene geometry inside the viewing frustum.

put camera frustum as seen from the overhead perspective. 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 similarly by using an unnormalized gating function $W(s,t,z) = z \cdot \delta[D_1(s,t) = z]$ applied to the y-coordinate feature $F(s,t) = s$.

Finally, we train a model to predict the virtual camera parameters which takes as input the RGB image and attempts to predict the target overhead viewpoint heuristically chosen based on ground-truth geometry (described above). The overhead viewpoint predictor takes the feature maps from the depth prediction network as input and outputs the orthographic translation and frustum radius parameters which is trained trained with L1 loss. While our final model can in principle be trained end-to-end (since the EFT is differentiable), 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 Adam optimizer to train all of our models with batch size 24 and learning rate 0.0005 for 40 epochs. The Physically-based Rendering [52] dataset uses a fixed downward tilt camera angle of 11 degrees, so we do not need to predict the gravity angle. We use an orthographic virtual camera for virtual view prediction, therefore the viewpoint proposal network outputs three values (t_x, t_y, σ) relative to the input camera viewpoint. At test time, the height of the virtual camera is the same as the input frontal camera and assumed to be known.

3D mesh generation. In order 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, we do not create an edge between those vertices. We use $a = 7$ throughout our experiments. 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.

5.3. Evaluation

Metrics. We use precision and recall of surface area as the metric to evaluate how closely the predicted meshes align with the ground truth. 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. This is illustrated in Figure 5. *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 the surfaces reconstructed

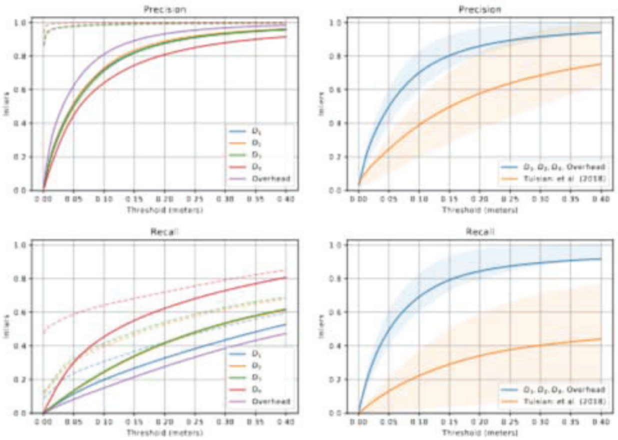


Figure 8: Precision and recall of scene geometry as a function of match distance threshold. *Left* column: Reconstruction quality of different model layers. Dashed lines indicate the upper bound performance given by ground-truth depth layers $(\bar{D}_1, \bar{D}_2, \bar{D}_3, \bar{D}_4)$. *Right* column: Reconstruction of our model in comparison to the state-of-the-art approach, evaluated against objects only. The upper and lower band indicate 75th and 25th quantiles. The higher variance of Tulsiani *et al.* [41] may be explained in part by the sensitivity of the model to having the correct initial set of object detections and pose estimates.

	L1 error
All features	**0.132**
Without semantics	0.141
Without semantics and depth features	0.144

Table 2: Transformed feature map ablation study. L1 error of overhead virtual view height map prediction, evaluated on both objects and rooms.

from ground-truth depths covers the full 3D mesh. This allows us to characterize the benefit of adding additional layers to the representation. Table 1 reports the coverage (recall) of the ground-truth at a threshold of 0.05 (5cm). The left panels of Figure 8 shows 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 dominant component of most scenes, we also analyzed performance on objects (excluding the envelope). These results are summarized in Table 3 which shows that: (a) the addition of multiple depth layers significantly increases recall with only a small drop in precision, and (b) the addition of the overhead surface predictions further improves 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-wise 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. As shown in Table 2, applying the EFT to the whole CNN feature map outperforms simple geometric transfer.

Comparison to state-of-the-art. Finally, we compare the scene reconstruction performance of our end-to-end approach with the object-based Factored3D [41] (Tulsiani *et al.*, 2018), using their pre-trained weights and converting voxel outputs to surface meshes using marching cubes. We evaluated on 4000 scenes from the SUNCG test set and compute precision and recall on objects surfaces (excluding envelope). As Figure 8 shows, our method yields roughly 5x improvement in recall and 2x increase in precision, providing estimates which are both more complete and more accurate. Figure 7 highlights some qualitative differences between the two methods.

Reconstruction on real-world images. Our network model is trained entirely on synthetically generated images [52]. We test the ability of the model to generalize to the NYUv2 dataset [24]. Figure 6 compares the output of our models with those Tulsiani *et al.* [41] on NYUv2 test images.

6. Conclusion

We've introduced a novel approach to complete 3D scene reconstruction from a single RGB image. We propose to estimate a per-pixel multi-layer depth map which represents front and back surfaces of objects as well as the room envelope. We also introduce *Epipolar Feature Transformer* networks that are capable of transforming input view features in order to hallucinate an over-head view of 3D scenes. The predicted 3D structure in this view provides complimentary cue for scene reconstruction. Experimental results on the SUNCG dataset [37] demonstrate the effectiveness of our design. We also compare with existing approaches that aim to predict voxel representations of scenes, and demonstrate significant potential of our multi-view multi-layer depth inference for reconstructing complete scenes.

	Precision	Recall
D_1	0.505	0.215
D_1 & Overhead	**0.540**	**0.298**
$D_{1,2,3}$	0.475	0.445
$D_{1,2,3}$ & Overhead	**0.499**	**0.494**

Table 3: Augmenting the frontal depth prediction with the predicted virtual view height map improves both precision and recall (match threshold of 5cm). We evaluate against ground truth object surfaces within the viewing frustum.

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

References

[1] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 3

[2] W. Chen, Z. Fu, D. Yang, and J. Deng. Single-image depth perception in the wild. In *Advances in Neural Information Processing Systems (NIPS)*, pages 730–738, 2016. 3

[3] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. 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] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[5] A. Dai, C. R. Qi, and M. 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. IEEE, 2017. 3

[6] H. Dhamo, K. Tateno, I. Laina, N. Navab, and F. Tombari. Peeking behind objects: Layered depth prediction from a single image. *arXiv preprint arXiv:1807.08776*, 2018. 3

[7] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2366–2374, 2014. 1, 3

[8] H. Fan, H. Su, and L. 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. IEEE, 2017. 3

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

[10] S. Ghosh and E. 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. IEEE, 2012. 3

[11] R. Girdhar, D. F. Fouhey, M. Rodriguez, and A. 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

[12] S. Gupta, P. Arbeláez, R. Girshick, and J. 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

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

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

[15] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. 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

[16] H. Izadinia, Q. Shan, and S. M. Seitz. Im2cad. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[17] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *Advances in Neural Information Processing Systems (NIPS)*, 2015. 5

[18] D. Ji, J. Kwon, M. McFarland, and S. Savarese. Deep view morphing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 3

[19] H. Jiang, E. Learned-Miller, G. Larsson, M. Maire, and G. Shakhnarovich. Self-supervised depth learning for urban scene understanding. *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 3

[20] A. Kar, S. Tulsiani, J. Carreira, and J. 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

[21] H. Kato, Y. Ushiku, T. Harada, A. Shin, L. Crestel, H. Kato, K. Saito, K. Ohnishi, M. Yamaguchi, M. Nakawaki, et al. Neural 3d mesh renderer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[22] C.-H. Lin, C. Kong, and S. Lucey. Learning efficient point cloud generation for dense 3d object reconstruction. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018. 3

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

[24] P. K. Nathan Silberman, Derek Hoiem and R. Fergus. Indoor segmentation and support inference from rgbd images. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2012. 7, 9

[25] E. Park, J. Yang, E. Yumer, D. Ceylan, and A. 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

[26] C. R. Qi, H. Su, K. Mo, and L. 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

[27] S. R. Richter and S. Roth. Matryoshka networks: Predicting 3d geometry via nested shape layers. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018. 3

[28] G. Riegler, A. O. Ulusoy, and A. 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. IEEE, 2017. 2

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

[30] J. Rock, T. Gupta, J. Thorsen, J. Gwak, D. Shin, and D. 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

[31] S. Rota Bulò, L. Porzi, and P. Kontschieder. In-place activated batchnorm for memory-optimized training of dnns. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, 2018. 12

[32] D. Shin, C. C. Fowlkes, and D. 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

[33] E. Smith, S. Fujimoto, and D. Meger. Multi-view silhouette and depth decomposition for high resolution 3d object representation. In *Advances in Neural Information Processing Systems (NIPS)*, pages 6479–6489, 2018. 3

[34] E. J. Smith and D. 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

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

[36] N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *Proc. of the IEEE Interna-*

tional Conference on Computer Vision (ICCV), 80(2):189–210, 2008. 3

[37] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. 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, 9

[38] H. Su, V. Jampani, D. Sun, S. Maji, E. Kalogerakis, M.-H. Yang, and J. 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

[39] H. Su, F. Wang, L. Yi, and L. Guibas. 3d-assisted image feature synthesis for novel views of an object. *arXiv preprint arXiv:1412.0003*, 2014. 3

[40] D. Sun, E. B. Sudderth, and M. 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. IEEE, 2012. 3

[41] S. Tulsiani, S. Gupta, D. Fouhey, A. A. Efros, and J. 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, 7, 8, 9

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

[43] S. Tulsiani, T. Zhou, A. A. Efros, and J. 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

[44] 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

[45] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 2, 3

[46] J. Wu, Y. Wang, T. Xue, X. Sun, B. Freeman, and J. Tenenbaum. Marrnet: 3d shape reconstruction via 2.5 d sketches. In *Advances in Neural Information Processing Systems (NIPS)*, pages 540–550, 2017. 3

[47] J. Wu, C. Zhang, X. Zhang, Z. Zhang, W. T. Freeman, and J. B. Tenenbaum. Learning 3D Shape Priors for Shape Completion and Reconstruction. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018. 3

[48] Z. Wu, S. Song, A. Khosla, F. Yu, L. Zhang, X. Tang, and J. 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

[49] X. Yan, J. Yang, E. Yumer, Y. Guo, and H. Lee. Perspective transformer nets: Learning single-view 3d object reconstruction without 3d supervision. In *Advances in Neural Information Processing Systems (NIPS)*, pages 1696–1704, 2016. 3

[50] B. Yang, S. Rosa, A. Markham, N. Trigoni, and H. Wen. 3d object dense reconstruction from a single depth view. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2018. 3

[51] X. Zhang, Z. Zhang, C. Zhang, J. Tenenbaum, B. Freeman, and J. Wu. Learning to reconstruct shapes from unseen classes. In *Advances in Neural Information Processing Systems (NIPS)*, pages 2263–2274, 2018. 3

[52] Y. Zhang, S. Song, E. Yumer, M. Savva, J.-Y. Lee, H. Jin, and T. 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, 8, 9

[53] T. Zhou, M. Brown, N. Snavely, and D. 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

[54] T. Zhou, R. Tucker, J. Flynn, G. Fyffe, and N. Snavely. Stereo magnification: Learning view synthesis using multiplane images. 2018. 3

[55] C. Zou, R. Guo, Z. Li, and D. Hoiem. Complete 3d scene parsing from single rgbd image. *International Journal of Computer Vision (IJCV)*, 2018. 3

[56] C. Zou, E. Yumer, J. Yang, D. Ceylan, and D. Hoiem. 3d-prnn: Generating shape primitives with recurrent neural networks. In *Proc. of the IEEE International Conference on Computer Vision (ICCV)*, pages 900–909, 2017. 3

A1. Multi-layer Depth Prediction

See Figure 9 for network parameters of our multi-layer depth prediction model. All batch normalization layers have momentum 0.005, and all activation layers are Leaky ReLUs layers with $\alpha = 0.01$. We use In-place Activated BatchNorm [31] for all of our batch normalization layers. We trained the network for 40 epochs. The meta parameters (learning rates, momentum, batch size, epochs, etc) are the same for all the networks in our system.

A2. Multi-layer Semantic Segmentation

See Figure 10 for network parameters for multi-layer semantic segmentation. We construct a binary mask for all foreground objects, and define segmentation mask M_l as all non-background pixels at layer l. As mentioned in section 3.1, D_1 and D_2 the same segmentation due to symmetry, so we only segment layers 1 and 3. The purpose of the foreground object labels is to be used as a supervisory signal for feature extraction F_{seg}, which is used as input to our Epipolar Feature Transformer Networks. Table 1 in our paper reports an ablation study that demonstrates the efficacy of the semantic features.

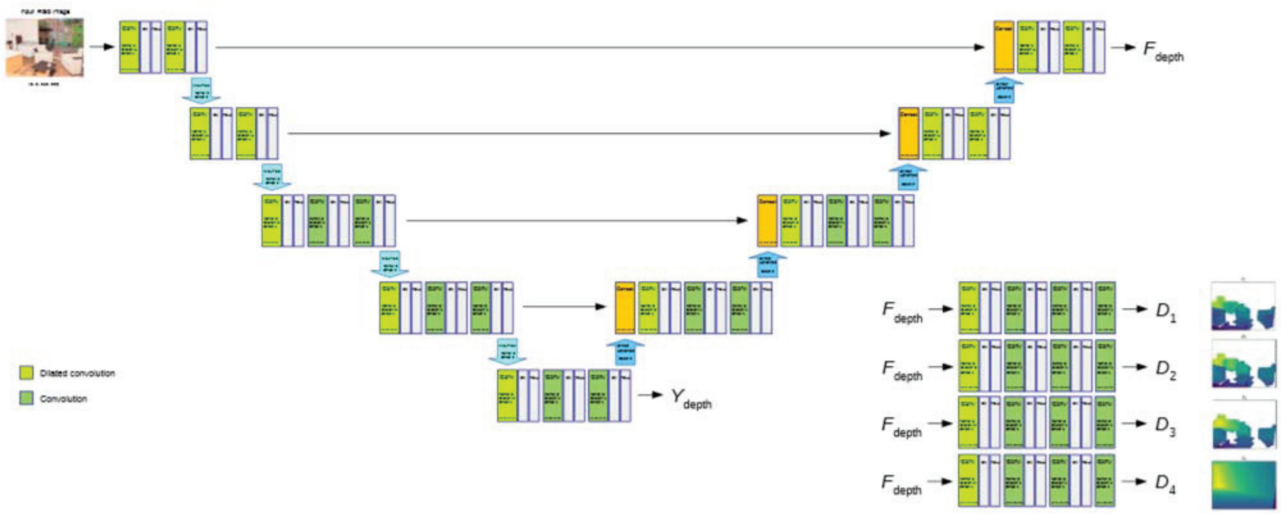

F_{depth}
Y_{depth}
Dilated convolution
Convolution
F_{depth} → D_1
F_{depth} → D_2
F_{depth} → D_3
F_{depth} → D_4

Figure 9: Network architecture for multi-layer depth prediction. The horizontal arrows in the network represent skip connections. This figure, along with following figures, is best viewed in color and on screen.

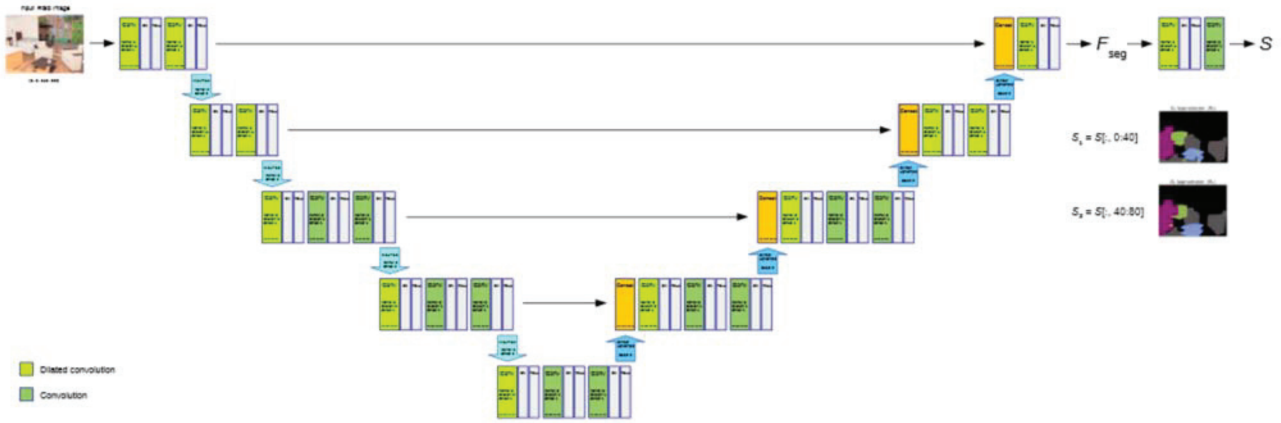


Figure 10: Network architecture for multi-layer semantic segmentation network. (Best viewed in color and on screen)

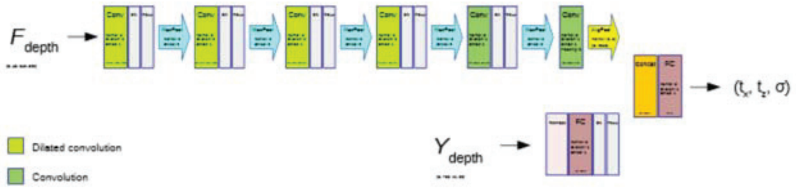


Figure 11: Network architecture for virtual camera pose proposal network. (Best viewed in color and on screen)

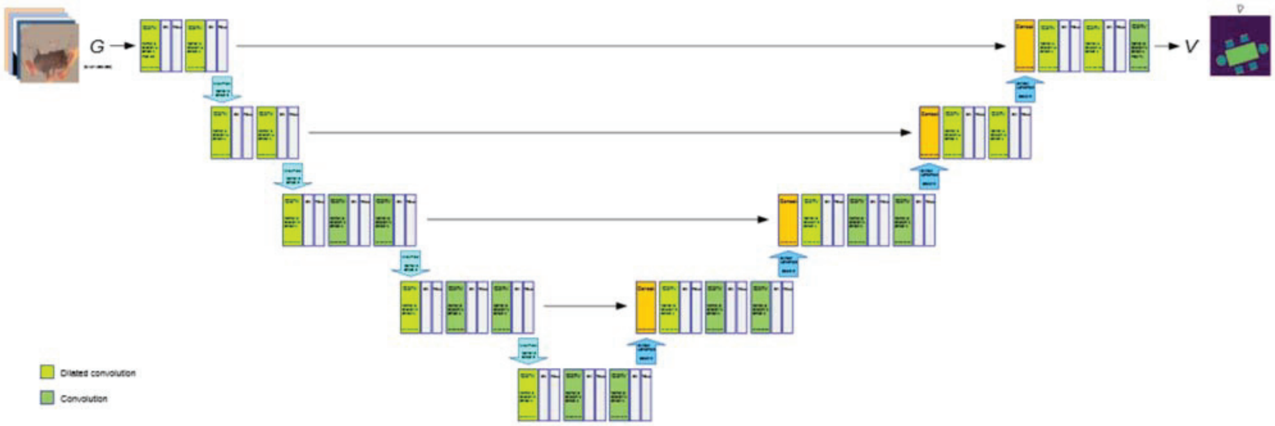


Figure 12: Network architecture for virtual view prediction network. (Best viewed in color and on screen)