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. As Figure 3 shows, our method yields roughly 5x improvement in recall and 2x increase in precision, providing estimates which are both more complete and more accurate.

1. Introduction

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 [2]. 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 [9]. 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 [7]). Other approaches represent shape as an unstructured point cloud [6, 10], but require development of suitable convolutional operators [3, 12] and fail to capture surface topology. 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 [8], 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 ultimately limit shape fidelity to much lower resolution than is provided by cues like occluding contours in the input image [9]. *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 [11]. 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

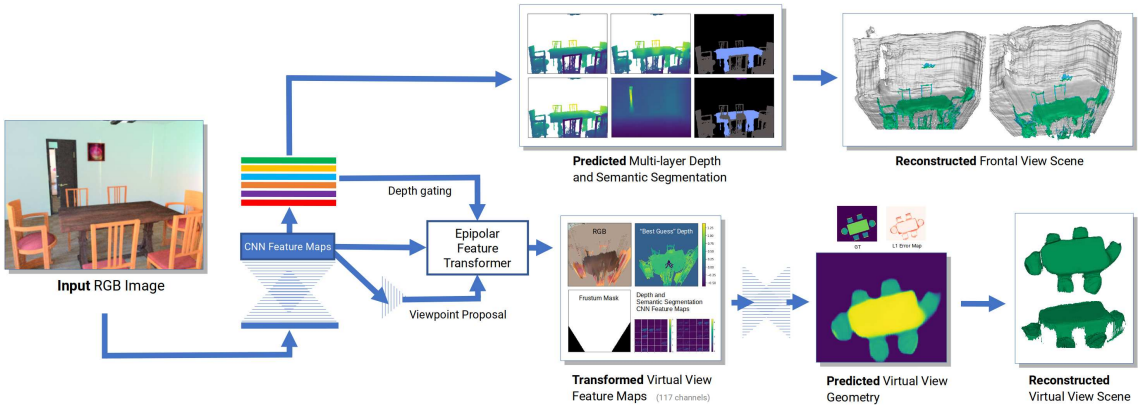

Figure 1: Overview of our system for reconstructing a complete 3D scene from a single RGB image. We first predict a multi-layer depth map that encodes the depths of front and back object surfaces as seen from the input camera. Given the extracted feature map and predicted multi-layer depths, the epipolar feature transformer network transfers features from the input view to a virtual overhead view, where the heights of observed objects are predicted. Explicit detection of object instances is not required, increasing robustness.

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. Reconstruction with Multi-Layer Depth

We perform multi-hit ray tracing on the ground-truth models from the SUNCG dataset [9] and represent the 3D scene geometry by recording multiple surface intersections. As illustrated in Figure 2(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 (Table 1) to empirically determine a fixed number of layers that provides good coverage of typical indoor 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 3).

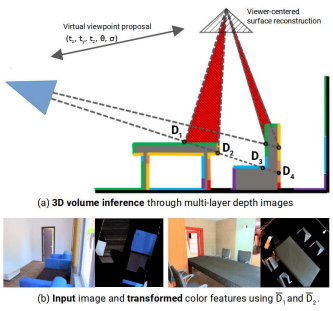

Figure 2: Epipolar transfer of features from the input image to a virtual overhead view.

Multi-Layer Depth Maps from 3D Geometry. To capture the overall extent of the space within the viewing frustum, we define the depth D_5 of the room envelope to be the *last* layer of the scene. We then model the shapes of observed objects by tracing rays from the input view. The first intersection D_1 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. To predict occluded structure behind foreground objects, we continue the same procedure and define layers D_3, D_4 as the depths of the next object intersection and the exit from that second object instance, respectively. 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. We also define a binary mask $\bar{M}_\ell$ which indicates the pixels where

$\bar{D}_1$	$\bar{D}_{1,2}$	$\bar{D}_{1,2,3}$	$\bar{D}_{1..4}$	$\bar{D}_{1..5}$	$\bar{D}_{1..5}$ +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.

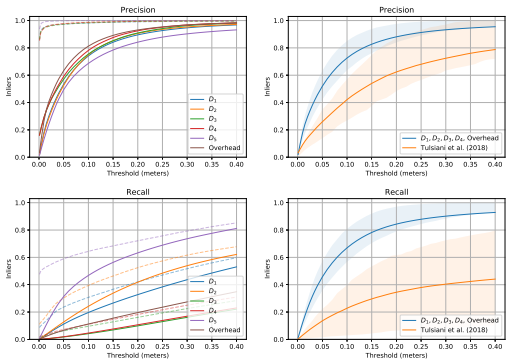

Figure 3: Precision and recall of scene geometry as a function of match distance threshold. *Left:* Reconstruction quality for different model layers. Dashed lines are the performance bounds provided by ground-truth depth layers. *Right:* Accuracy of our model relative to the state-of-the-art The upper and lower band indicate 75th and 25th quantiles. The higher variance of Tulsiani *et al*. [11] may be explained in part by the sensitivity of the model to having the correct initial set of object detections and pose estimates.

layer $\ell \in \{1, 3\}$ has support. Note that $\bar{M}_1 = \bar{M}_2$, and $\bar{M}_3 = \bar{M}_4$, due to symmetry. Experiments in Figure 3 evaluate the relative importance of different layers in modeling realistic 3D scenes.

Predicting Multi-Layer Depth Maps. To learn to predict the five-channel depths given a single image as input, we train a standard encoder-decoder network with skip connections and minimize the Huber loss. Our pixel-wise multi-layer depth prediction is agnostic to high-level semantic information, so we also predict a layer-wise semantic segmentation. M_ℓ is defined as the non-background pixels at predicted segmentation layer ℓ. The purpose of the foreground labels, though not required, is to be used as a supervisory signal for feature extraction in our EFT network.

3. 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 1 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) [4] 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 $\begin{pmatrix} x \\ y \\ z \end{pmatrix} = z\mathbf{K_I}^{-1} \begin{pmatrix} s \\ t \\ 1 \end{pmatrix}$ for $z \geq 0$, where $\mathbf{K}_I \in \mathbb{R}^{3\times3}$ encodes the input camera intrinsic parameters. z is the depth along the viewing ray, whose image in a virtual orthographic camera is given by

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

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.

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

4. 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

[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.

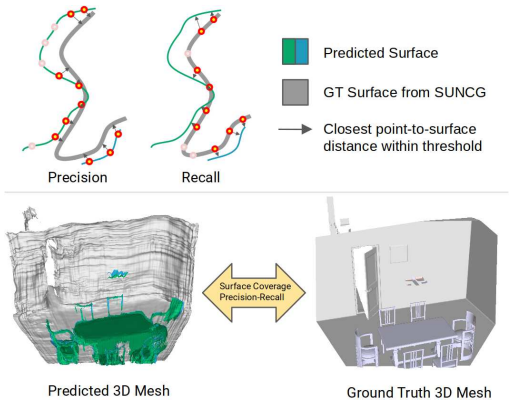

Figure 4: Illustration of our 3D precision-recall metrics. *Top*: We perform a bidirectional surface coverage evaluation on the reconstructed triangle meshes. *Bottom*: The ground truth mesh consists of all 3D surfaces within the field-of-view and in front of the room envelope. See Figure 1 for the corresponding input and output images.

	Precision	Recall
$D_{1,2,3,4,5}$ & Overhead	**0.221**	**0.358**
Tulsiani *et al*. [11]	0.132	0.191

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

multi-layer and multi-view depths from physical renderings of indoor scenes [13] provided by the SUNCG dataset [9].

We test our models on 4000 SUNCG scenes as well as ScanNet [1] and NYU [5]. We evaluate precision and recall of points uniformly sampled from the ground-truth and predicted surfaces (Figure 4). ScanNet contains more complete geometry of real-world scenes, so we can provide a real-world quantitative evaluation as well.

To reconstruct 3D surfaces from the output of our network model, we first convert the predicted depth images into a point cloud and triangulate vertices that correspond to a 2×2 neighborhood in image space within a threshold relative to the pixel footprint in camera coordinates.

Our model is trained entirely synthetically, and we provide quantitative results for both synthetic (Figure 3 and Table 3) and real-world (Table 2) scenes that significantly outperform the object-based approach [11]. 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. Figure 5 visualizes the output reconstruction of our models on synthetic images. Figure 6 shows a qualitative comparison on real-world images against the obeject-based approach [11].

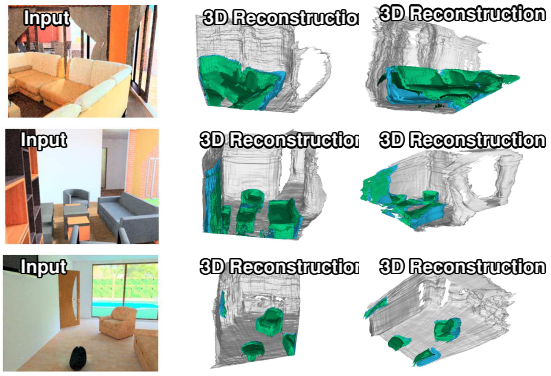

Figure 5: Estimates of the front (green) and back (cyan) surfaces of objects are complemented by heights estimated by a virtual overhead camera (dark green). Room envelope estimates are rendered in gray.

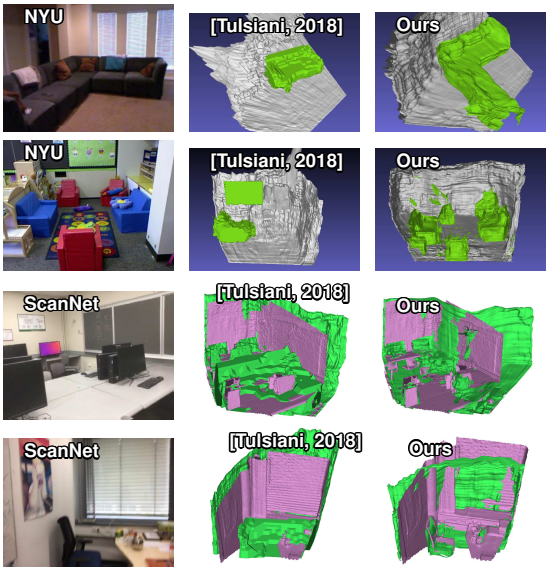

Figure 6: Evaluation of 3D reconstruction on the NYUv2 [5] and ScanNet [1] dataset, where green regions are predicted geometry and pink regions are ground truth. Tulsiani *et al*. [11] are sensitive to the performance of 2D object detectors, and their voxelized output is a coarse approximation of the true 3D geometry.

	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).

References

[1] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *CVPR*, 2017. 4

[2] D. Eigen, C. Puhrsch, and R. Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NIPS*, pages 2366–2374, 2014. 1

[3] M. Gadelha, R. Wang, and S. Maji. Multiresolution tree networks for 3d point cloud processing. In *ECCV*, September 2018. 1

[4] M. Jaderberg, K. Simonyan, A. Zisserman, et al. Spatial transformer networks. In *NIPS*, 2015. 3

[5] P. K. Nathan Silberman, Derek Hoiem and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. 4

[6] C. R. Qi, H. Su, K. Mo, and L. J. Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *CVPR*, 2017. 1

[7] G. Riegler, A. O. Ulusoy, and A. Geiger. Octnet: Learning deep 3d representations at high resolutions. In *CVPR*, pages 6620–6629. IEEE, 2017. 1

[8] 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 *CVPR*, 2018. 1

[9] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic scene completion from a single depth image. In *CVPR*, 2017. 1, 2, 4

[10] 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 *CVPR*, pages 2530–2539, 2018. 1

[11] 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 *CVPR*, 2018. 1, 3, 4

[12] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang. Pixel2mesh: Generating 3d mesh models from single rgb images. *ECCV*, 2018. 1

[13] 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 *CVPR*, 2017. 4