Voxel-based 3D Detection and Reconstruction of Multiple Objects from a Single Image — Supplementary Material

Feng Liu **Xiaoming Liu**
Department of Computer Science and Engineering
Michigan State University, East Lansing MI 48824
{liufeng6, liuxm}@msu.edu

In this supplementary material, we provide:

◇ Additional implementation details including network structures, 3D heatmap calculation and data preparation.

◇ Additional experimental results including visualization of the learned voxel features, additional detection and reconstruction results, and shape representation comparison.

1 Implementation Details

1.1 Network Structures

We use a ResNet-34 without bottleneck layers as the backbone to extract features. The detailed architecture is depicted in Fig. 1. The network takes an image as input and generates a D-channel multi-scale 2D feature maps $\mathbf{F} \in \mathbb{R}^{W_F \times H_F \times D}$. In our experiments, the image size of Pix3D, ShapeNet-pairs and -triplets data is 256×256 pixels, and 480×640 for ScanNet-MDR data. The 3D keypoint, regression and coarse-level voxelization branches are separately implemented using a two-convolution-layers network with sizes of $3 \times 3 \times 128$ and $1 \times 1 \times *$, where $*$ is the feature channel of the respective output branch, *i.e.,* 8 for the regression branch.

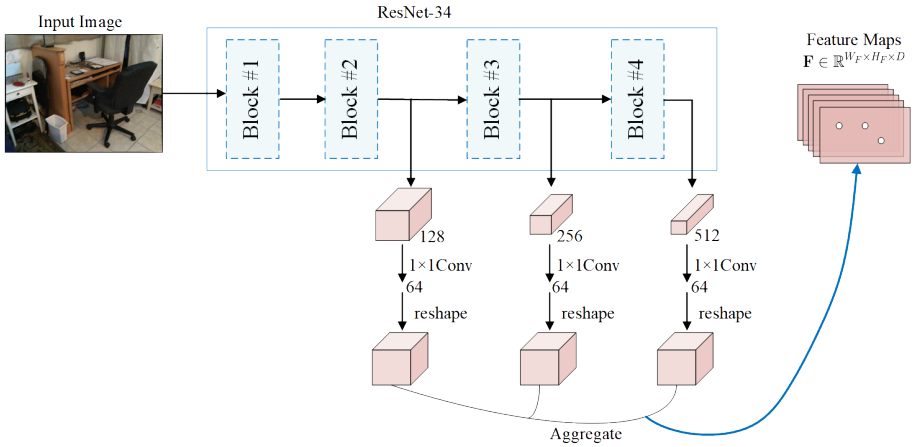

Figure 1: The architecture of the 2D feature extraction network. We reshape the feature maps to the original image size with bilinear interpolation.

35th Conference on Neural Information Processing Systems (NeurIPS 2021), Sydney, Australia.

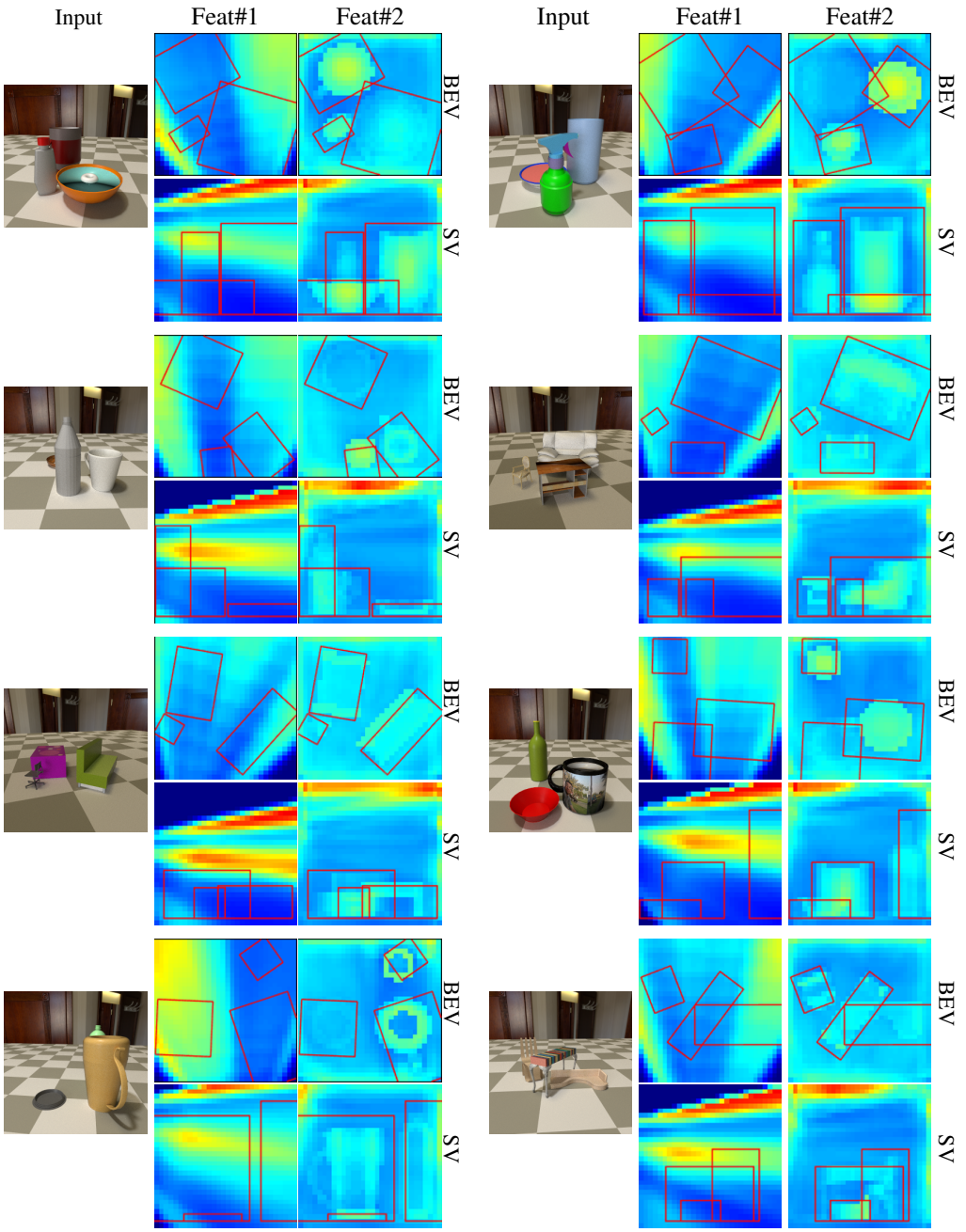

Figure 2: Visualizations of the learned voxel features $\mathbf{G}$. Feat $\#1$: the preliminary $\mathbf{G}$ before feeding to positional encoding and 3D U-Net. Feat $\#2$: the output of the 3D U-Net. We visualize the feature maps in both bird's-eye view (BEV) and side view (SV) by applying average pooling to $\mathbf{G}$. As can be seen, the learned voxel features indeed reflect the object geometry and scene context information. The red boxes show the ground-truth position of the objects.

1.2 3D Heatmap Calculation

For each ground-truth 3D keypoint of class c, we locate it at the pre-defined 3D grid and obtain its corresponding voxel center $\mathbf{c}_{3d}^{*} = [x_c, y_c, z_c]^T$. We then splat the center point onto the 3D grid and

generate a heatmap using a Gaussian kernel

$$\mathcal{Y}^*_{xyzc} = \exp\left(-\frac{(x-x_c)^2+(y-y_c)^2+(z-z_c)^2}{2\sigma_c^2}\right), \tag{1}$$

where σ_c is an object size-adaptive standard deviation, and is set to the diagonal length of the 3D box in our experiments. If two Gaussians of the same class overlap, we take the element-wise maximum [1].

1.3 Data Preparation

The training of our local PCA-SDF model needs a regular of point-SDF pairs for each boundary voxel. We first voxelize and binarize the surface in our pre-defined 3D grid $\mathbf{V}$. For each occupancy voxel of the binary grid, we sample a regular lattice $\mathbf{q} \in \mathbb{R}^{k\times k\times k\times 3}$ (see Fig.3 (c) in the main paper) and compute their SDFs $\mathbf{s} \in \mathbb{R}^{k\times k\times k\times 1}$ toward the surface by the work [2]. Based on the computed SDFs, each voxel can be further divided into full occupancy voxel or boundary voxel. For instance, if all elements of s are negative, it is a full occupancy voxel; otherwise, it is a boundary voxel. Additionally, we can obtain the continuous coarse-level voxelization by measuring the ratio of the number of negative elements for every voxel.

2 Experimental Results

2.1 Visualization of the Voxel Features

A central claim of our approach is that we learn geometry and context preserving voxel feature representation which boosts the 3D detection and reconstruction performances. To investigate this, we visualize the learned voxel feature $\mathbf{G}$ via two different views: bird's-eye view (BEV) and side view (SV). For either view, we apply average pooling across the feature channels. As shown in Fig. 2, we show the features at two different stages: before feeding to positional encoding and 3D U-Net (Feat #1) and output of the 3D U-Net (Feat #2). As can be observed, the voxel features (Feat #2) visualize the objects' contours, which indicates that the voxel features indeed learn the object geometry information. Additional, the voxel features reflect the context information (*i.e.,* position and distance) of objects. Moreover, as mentioned in the main paper, the 2D-to-3D feature lifting operator suffers from the feature smearing issue, wherein all voxels along a camera ray receive the same feature information (see Feat #1). Thanks to the positional encoding and 3D U-Net, as shown in Fig. 2, our voxel features can be semantically queried for any 3D location, even if it is not visible from the input view.

2.2 Additional Reconstruction and Detection Results (video)

We provide additional reconstruction and detection results on Pix3D, ShapeNet-triplets and ScanNet-MDR datasets (please also refer to the supplementary video). Since CoReNet [3] and Points2Objects [4] do not release the models trained on Pix3D, we only provide qualitative comparisons with SOTA single object reconstruction method Liu *et al.* (CVPR21') [5] on Pix3D. As can be observed in Fig. 3 and the supplementary video, our reconstructions closely match the ground truth. Figure 4 and 5 show qualitative results of detection and reconstruction on ShapeNet-triplets and ScanNet-MDR datasets respectively.

2.3 Shape Representation Comparison (video)

In this experiment, we additionally provide qualitative shape representation comparisons with DeepSDF [6] and DeepLS [7] in Fig. 6 and the supplementary video. For the local PCA-SDF model, we also show the reconstructions on unseen categories with the models trained on one category in Fig. 7 (please also refer to the dynamic version in supplementary video). As can be seen, the model trained on one category (*i.e.*, chair) can reconstruct other categories' shapes at high quality.

References

[1] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019.

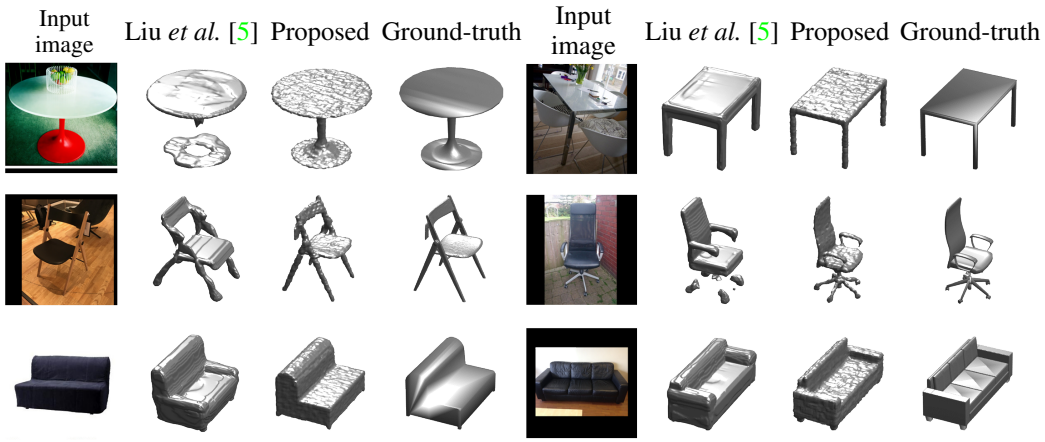

Figure 3: Qualitative single object reconstruction comparison with Liu *et al.* [5] on Pix3D dataset.

[2] Fun Shing Sin, Daniel Schroeder, and Jernej Barbič. Vega: non-linear fem deformable object simulator. In *Computer Graphics Forum*, 2013.

[3] Stefan Popov, Pablo Bauszat, and Vittorio Ferrari. CoReNet: Coherent 3D scene reconstruction from a single RGB image. In *ECCV*, 2020.

[4] Francis Engelmann, Konstantinos Rematas, Bastian Leibe, and Vittorio Ferrari. From points to multi-object 3D reconstruction. In *CVPR*, 2021.

[5] Feng Liu, Luan Tran, and Xiaoming Liu. Fully understanding generic objects: Modeling, segmentation, and reconstruction. In *CVPR*, 2021.

[6] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. DeepSDF: Learning continuous signed distance functions for shape representation. In *CVPR*, 2019.

[7] Rohan Chabra, Jan E Lenssen, Eddy Ilg, Tanner Schmidt, Julian Straub, Steven Lovegrove, and Richard Newcombe. Deep local shapes: Learning local SDF priors for detailed 3D reconstruction. In *ECCV*, 2020.

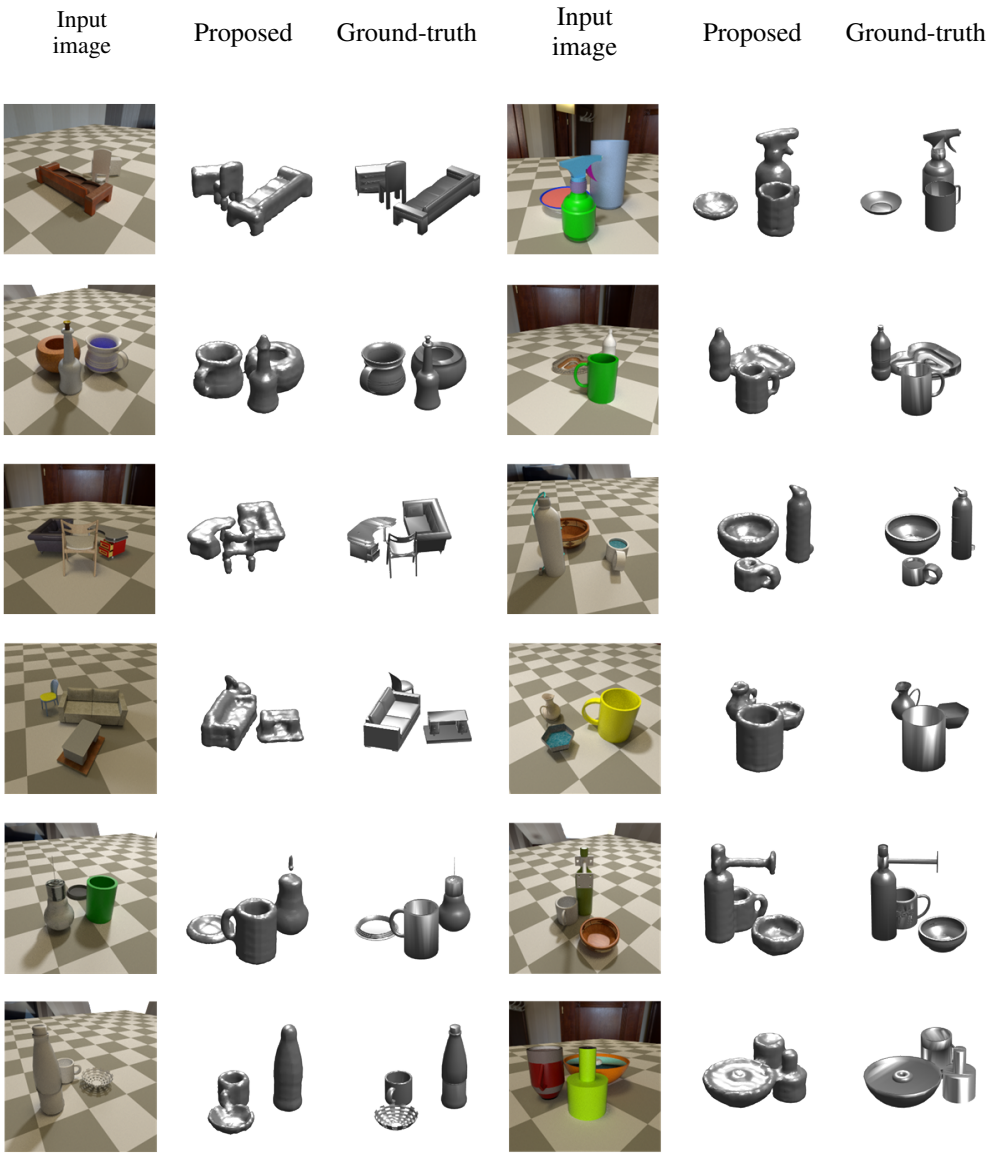

Figure 4: Qualitative results on ShapeNet-triplets dataset.

Figure 5: Qualitative results on ScanNet-MDR dataset.

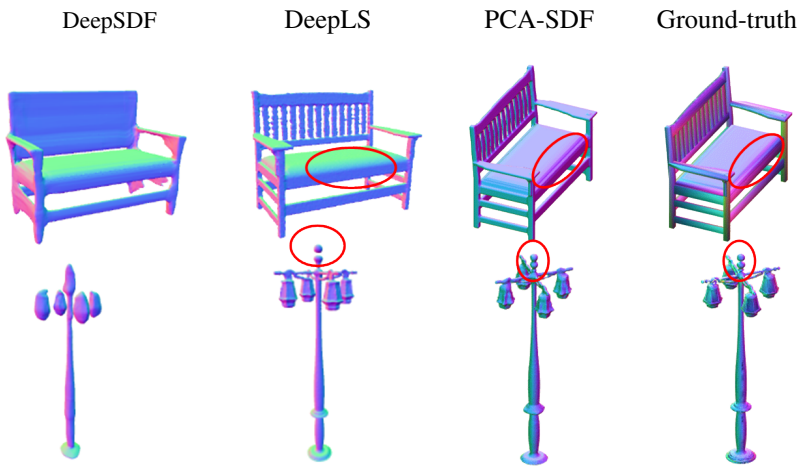

Figure 6: Qualitative comparison of our local PCA-SDF with DeepSDF [6] and DeepLS [7] on some shapes from the ShapeNet dataset.

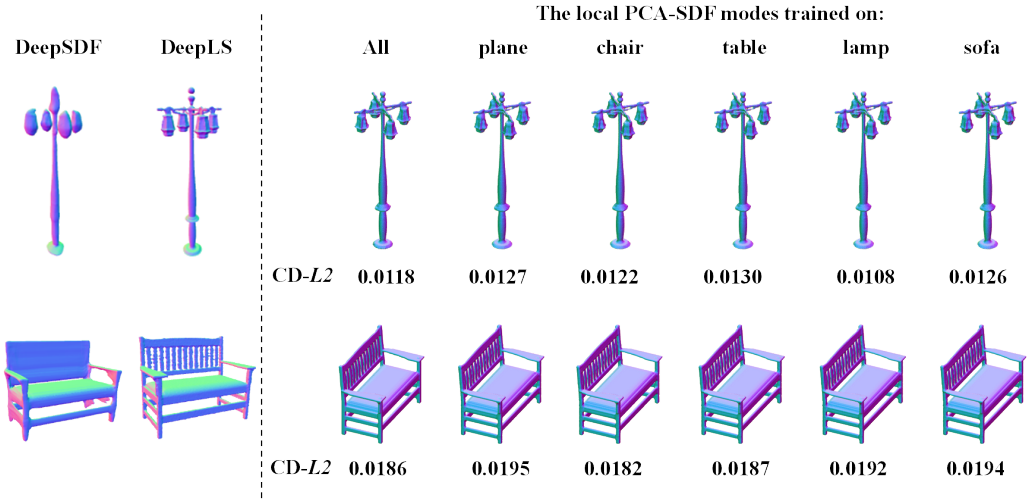

Figure 7: Two reconstructed examples on unseen categories with the models trained on one category.