MV-DeepSDF: Implicit Modeling with Multi-Sweep Point Clouds for 3D Vehicle Reconstruction in Autonomous Driving

Yibo Liu[1,2], Kelly Zhu[1,3], Guile Wu[1], Yuan Ren[1], Bingbing Liu[1], Yang Liu[1], Jinjun Shan[2]
[1]Huawei Noah's Ark Lab, [2]York University, [3]University of Toronto
{yorklyb,jjshan}@yorku.ca, kellyk.zhu@mail.utoronto.ca
{guile.wu, yuan.ren3, liu.bingbing, yang.liu9}@huawei.com

Supplementary Material

A. Overview

This material provides quantitative and qualitative experimental results, dataset and implementation details, and discussions that are supplementary to the main paper.

B. Dataset Details

The multi-sweep LiDAR point clouds for Waymo are collected from 136 unique vehicle instances of Waymo Open Dataset's tracking data [8], while those for KITTI are extracted from 233 unique vehicle instances of KITTI's tracking dataset [5]. In total, we extracted 3943 partial point clouds from the 136 vehicle instances of Waymo and 4235 partial point clouds from the 233 vehicle instances of KITTI. These raw multi-sweep point clouds are directly used as model input to obtain experimental results on our model and the state-of-the-art methods [7, 2, 3, 10]. When performing inference, all partial point cloud frames from the given instance are simultaneously passed into our model as input.

To construct the ground truth stacked point cloud, we first aggregate all partial point clouds within a multi-sweep to generate a dense stacked point cloud. Since this stacked point cloud contains unwanted noise, such as ground plane points and points lying on the exterior or interior of the vehicle surface, we perform statistical outlier removal on the stacked point cloud, which computes the average distance of a point from its neighbours and removes all points lying farther away from their neighbours than average. Denoising is essential for dataset processing since the presence of noise in the ground truth shape can result in false positives and false negatives in model performance, whereby a messy shape generated by a model that fits to the noise of the ground truth is deemed high fidelity and a smooth shape generated by a noise-robust model is deemed low fidelity.

C. Results on ShapeNetV2

We randomly preserve 300 vehicles from the car taxonomy of ShapeNetV2 [1] as the test dataset. The remainder is used to train our network in stage one. Each vehicle instance is comprised of 6 partial point clouds generated by PCGen [6] under Waymo's LiDAR parameters. Note that the following results are generated solely using PCGen [6] to sample partial point clouds and the use of other sampling techniques (*e.g.*, the approaches proposed in [7, 9]) would yield different results on the same ShapeNetV2 dataset [1].

Metrics. Since ground truth shapes are readily available in synthetic datasets such as ShapeNetV2 [1], we use Chamfer Distance (CD) [4] to evaluate the 3D reconstruction results. Following DeepSDF [7], we sample 30,000 points on the surface of both the ground truth and reconstructed mesh. Given two point sets, the CD is the sum of the squared distance of each point to the nearest point in the other point set:

$$CD\left(\boldsymbol{X}, \boldsymbol{Y}\right) = \sum_{x \in \boldsymbol{X}} \min_{y \in \boldsymbol{Y}} \|x - y\|_2^2 + \sum_{y \in \boldsymbol{Y}} \min_{x \in \boldsymbol{X}} \|x - y\|_2^2. \tag{1}$$

As outlined in the main paper, we only compare the results of our model with the best single-shot reconstruction result, which is the mesh with the minimum CD among the multiple single-shot reconstructed meshes.

Qualitative and Quantitative Comparison. The qualitative and quantitative comparison of our approach against the state-of-the-art methods (DeepSDF [7], C-DeepSDF [2], MendNet [3], and AdaPoinTr [10]) are presented in Figure 1 and Table 1, respectively. Note that when testing on ShapeNetV2 [1], since PCGen [6] is used to generate both the training and test dataset, we only present Ours, in contrast with the comparison of Ours with Ours-VDC in the

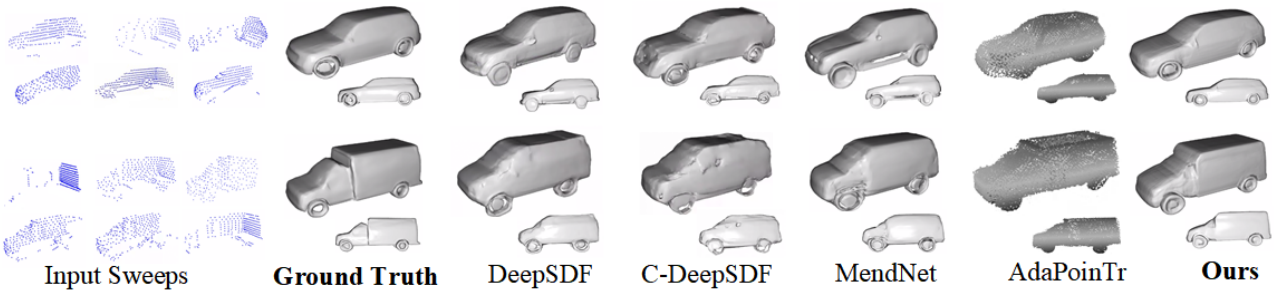

Figure 1. Visual comparison with the state-of-the-art methods (DeepSDF [7], C-DeepSDF [2], MendNet [3], and AdaPoinTr [10]) on the ShapeNetV2 [1] dataset.

main paper. The meshes generated by DeepSDF [7], C-DeepSDF [2], and MendNet [3] show high fidelity compared to their performance on real-world datasets, but still show inferior performance to Ours. AdaPoinTr [10] also produces shapes with decent fidelity, but the reconstructed result is not watertight and expresses the shape with a limited resolution which fails to describe the continuous surface of the vehicle.

Method \ Metric	$CD_{mean} \downarrow$	$CD_{median} \downarrow$
DeepSDF [7]	5.47	5.15
C-DeepSDF [2]	5.31	5.03
MendNet [3]	4.22	3.65
AdaPoinTr [10]	4.10	3.36
Ours	**3.17**	**2.54**

Table 1. Comparison of the proposed network with the state-of-the-art approaches on ShapeNetV2 [1]. CD is multiplied by 10^3.

D. Comparison to a Non-Learning Approach

We now present an alternative non-learning approach, computing the mean latent code, for the task of multi-sweep 3D vehicle reconstruction. As introduced in [7], linear interpolation between two latent codes in the latent space can also generate meaningful shape representations. Moreover, averaging is a common method of linear interpolation used for reducing error among multi-observation data. To this end, we investigate the effect of computing the mean latent code from the single-shot-based latent codes of a given multi-sweep and using this mean latent code for mesh reconstruction. We present the case shown in Figure 2, where two single-shot partial point clouds, PC 1 and PC 2, are used to generate two latent codes, $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$, and meshes, Mesh 1 and Mesh 2, respectively. We define the mean latent code as $\boldsymbol{z}_{mean} = 0.5(\boldsymbol{z}_1 + \boldsymbol{z}_2)$ and generate the corresponding mesh, denoted by Mean. As shown, Mean is simply a uniform fusion of Mesh 1 and Mesh 2. Moreover, Mean is inferior to Mesh 2, the best single-shot in this example, which is also inferior to Ours, the result of our proposed model.

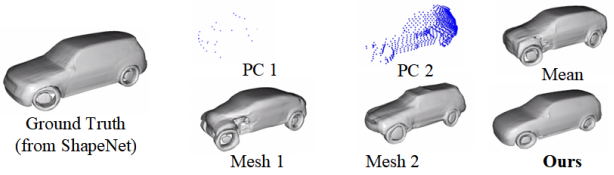

Figure 2. Comparison of the result of our approach to that of the mean latent code on ShapeNetV2 [1]. The proposed network is not fine-tuned.

Num of PCs	$ACD_{mean} \downarrow$	$ACD_{median} \downarrow$
3	3.47	2.44
6	3.36	2.26
9	3.32	2.21

Table 2. Ablation study on Waymo [8] using different numbers of point clouds per instance. ACD is multiplied by 10^3.

E. Effect of Number of Point Clouds

The number of frames corresponding to an individual vehicle instance in Waymo [8] and KITTI [5] ranges up to 240 partial point clouds per instance. However, the vast majority of instances only contain between 3 to 9 partial point clouds. In this section, we investigate the relationship between the number of partial point clouds provided for each instance during stage two of training and overall model performance. Table 2 presents the experimental results of providing different numbers of partial point clouds to our model on Waymo [8]. As shown, our model performance improves as the number of point clouds increases. However, since generating the latent code for each partial point cloud with DeepSDF [7] is a timely process (around 10 seconds), we choose 6 observations per instance as a trade-off between performance and efficiency.

F. Effect of Number of Points Per Point Cloud

The number of points captured in a single frame of Waymo [8] and KITTI [5] mostly falls into a range of 300 to 1000 points. Thus, we set the number of points per point cloud as 256 in our framework for performing FPS. In this

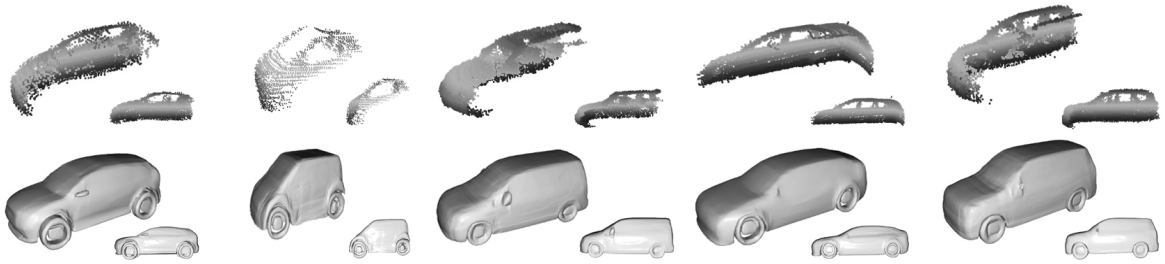

Figure 3. Additional visualization results of MV-DeepSDF on the Waymo [8] and KITTI [5] datasets.

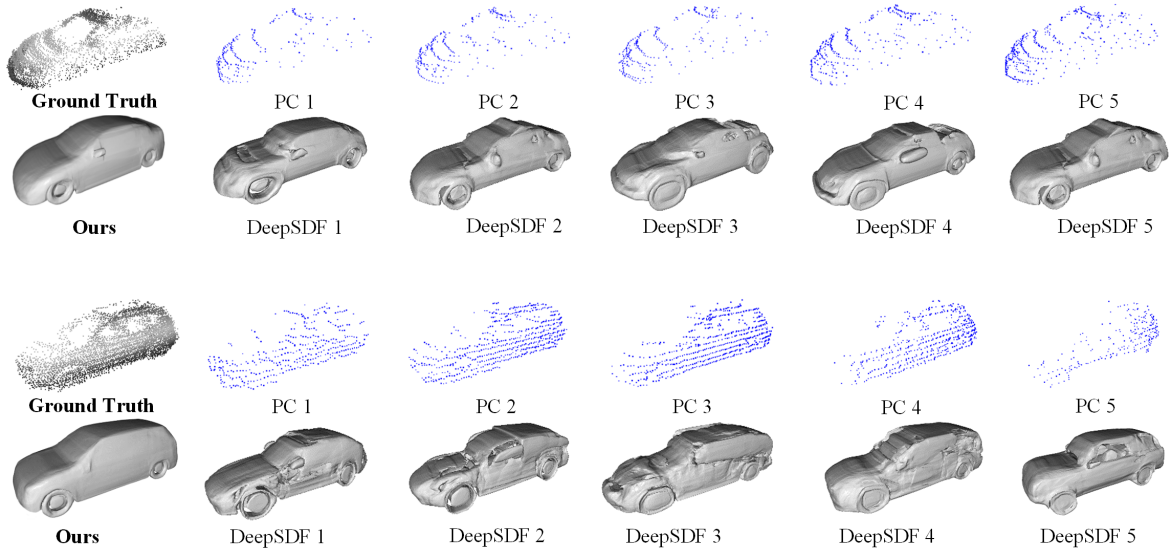

Figure 4. Visual comparison of MV-DeepSDF and DeepSDF [7] on Waymo [8]. Individual point clouds are given in the first row and their corresponding reconstruction results from vanilla DeepSDF in the second row.

section, we investigate the relationship between the number of points per point cloud during inference and model performance. Table 3 presents the experimental results of varying the number of points per point cloud on both DeepSDF and our model with Waymo [8]. As shown, when the number of points decreases, the performance of DeepSDF drops dramatically whereas our method holds steady.

Num of Points	256		128	
Metric	$\text{ACD}_{mean}\downarrow$	$\text{ACD}_{median}\downarrow$	$\text{ACD}_{mean}\downarrow$	$\text{ACD}_{median}\downarrow$
DeepSDF	6.26	5.81	12.52	8.51
Ours	**3.36**	**2.26**	**3.47**	**2.64**

Table 3. Ablation study on Waymo [8] using different numbers of points per point cloud. ACD is multiplied by 10^3.

G. Visualization Results

Due to the page limitation of the main paper, we present more visualization results of our model in Figure 3. Additionally, to evidently present the significant improvement of our method over the baseline vanilla DeepSDF [7], a visual comparison on Waymo [8] is given in Figure 4.

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.

[2] Yueqi Duan, Haidong Zhu, He Wang, Li Yi, Ram Nevatia, and Leonidas J Guibas. Curriculum deepsdf. In *Proc. of European Conference on Computer Vision*, pages 51–67. Springer, 2020.

[3] Shivam Duggal, Zihao Wang, Wei-Chiu Ma, Sivabalan Manivasagam, Justin Liang, Shenlong Wang, and Raquel Urtasun. Mending neural implicit modeling for 3d vehicle reconstruction in the wild. In *Proc. of IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1900–1909, 2022.

[4] Haoqiang Fan, Hao Su, and Leonidas J Guibas. A point set generation network for 3d object reconstruction from a single image. In *Proc. of the IEEE conference on computer vision and pattern recognition*, pages 605–613, 2017.

[5] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *The Inter-*

national Journal of Robotics Research, 32(11):1231–1237, 2013.

[6] Chenqi Li, Yuan Ren, and Bingbing Liu. Pcgen: Point cloud generator for lidar simulation. In *Proc. of International Conference on Robotics and Automation*, 2023.

[7] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proc. of the IEEE/CVF conference on computer vision and pattern recognition*, pages 165–174, 2019.

[8] Pei Sun, Henrik Kretzschmar, Xerxes Dotiwalla, Aurelien Chouard, Vijaysai Patnaik, Paul Tsui, James Guo, Yin Zhou, Yuning Chai, Benjamin Caine, et al. Scalability in perception for autonomous driving: Waymo open dataset. In *Proc. of the IEEE/CVF conference on computer vision and pattern recognition*, pages 2446–2454, 2020.

[9] Xumin Yu, Yongming Rao, Ziyi Wang, Zuyan Liu, Jiwen Lu, and Jie Zhou. Pointr: Diverse point cloud completion with geometry-aware transformers. In *Proc. of the IEEE/CVF international conference on computer vision*, pages 12498–12507, 2021.

[10] Xumin Yu, Yongming Rao, Ziyi Wang, Jiwen Lu, and Jie Zhou. Adapointr: Diverse point cloud completion with adaptive geometry-aware transformers. *arXiv preprint arXiv:2301.04545*, 2023.