Online Adaptation for Implicit Object Tracking and Shape Reconstruction in the Wild

Jianglong Ye
TuSimple
jianglong.yeh@gmail.com

Yuntao Chen
TuSimple
chenyuntao08@gmail.com

Naiyan Wang
TuSimple
winsty@gmail.com

Xiaolong Wang
UCSD
xiw012@ucsd.edu

Abstract

Tracking and reconstructing 3D objects from cluttered scenes are the key components for computer vision, robotics and autonomous driving systems. While recent progress in implicit function (e.g., DeepSDF) has shown encouraging results on high-quality 3D shape reconstruction, it is still very challenging to generalize to cluttered and partially observable LiDAR data. In this paper, we propose to leverage the continuity in video data. We introduce a novel and unified framework which utilizes a DeepSDF model to simultaneously perform object tracking and 3D reconstruction in the wild. We perform online adaptation with the DeepSDF model in the video, iteratively improving the shape reconstruction which leads to improvement on tracking, and vice versa. We experiment with the Waymo dataset, and show significant improvements over state-of-the-art methods for both tracking and shape reconstruction.

1. Introduction

Given a sequence of a LiDAR scans of the object in Figure 1, we humans can recognize it is a moving "car" even if the object is presented in a cluttered environment with only partial point clouds visible in each time step. Beyond recognizing the object, we are also able to imagine the full 3D structure of the object and its pose. In a video, by tracking the object and aggregating the information through time, the object 3D structure becomes more apparent to us. While 3D perception is easy for humans, 3D object tracking, pose estimation, and shape reconstruction are still very challenging problems in computer vision, and they are the key components for robotics and autonomous driving systems.

Recent development on implicit function has shown a tremendous success on high-quality 3D shape reconstruction [3, 10, 12, 1, 8]. Specifically, DeepSDF [12] is proposed to use a deep auto-decoder which takes a shape code and a coordinate as inputs to predict the signed distance to the shape surface. By training a category level DeepSDF, it learns the shape prior of the object category. With this prior, the DeepSDF can be used to reconstruct a new instance by optimizing the shape code via back-propagation. While DeepSDF has been shown to be very effective with synthetic data (e.g., ShapeNet [2]) and with dense observations (e.g., ScanNet [4]), it suffers from reconstruction with partial point clouds in cluttered scenes. Even with the learned prior, it either reconstructs close to mean shape or overfits to the noise given limited observations and the noisy artifacts around. As shown in the first time step in Figure 1, the reconstructed shape is very different from the ground-truth point clouds. With the help of the video, we should be able to align and aggregate multiple observations over time for a better reconstruction. For example, the car shape in the $5th$ time step is much closer to the ground-truth point clouds in Figure 1. However, it raises another challenge to

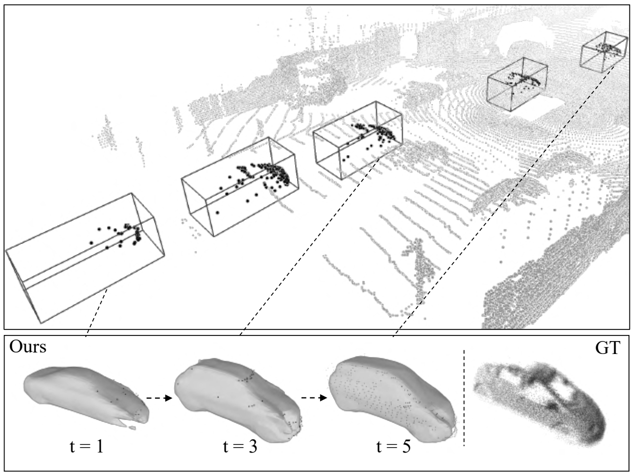


Figure 1. **3D object tracking and shape reconstruction in the wild.** We propose a novel framework to utilize DeepSDF to perform tracking and reconstruction simultaneously. We visualize 5 time steps of tracking and 3 reconstructed shapes out of them. While our model reconstructs a shape close to an average car with few point clouds in the beginning. By tracking the object over time, we obtain better and better shape close to the ground-truth point clouds, which in return helps the tracking.

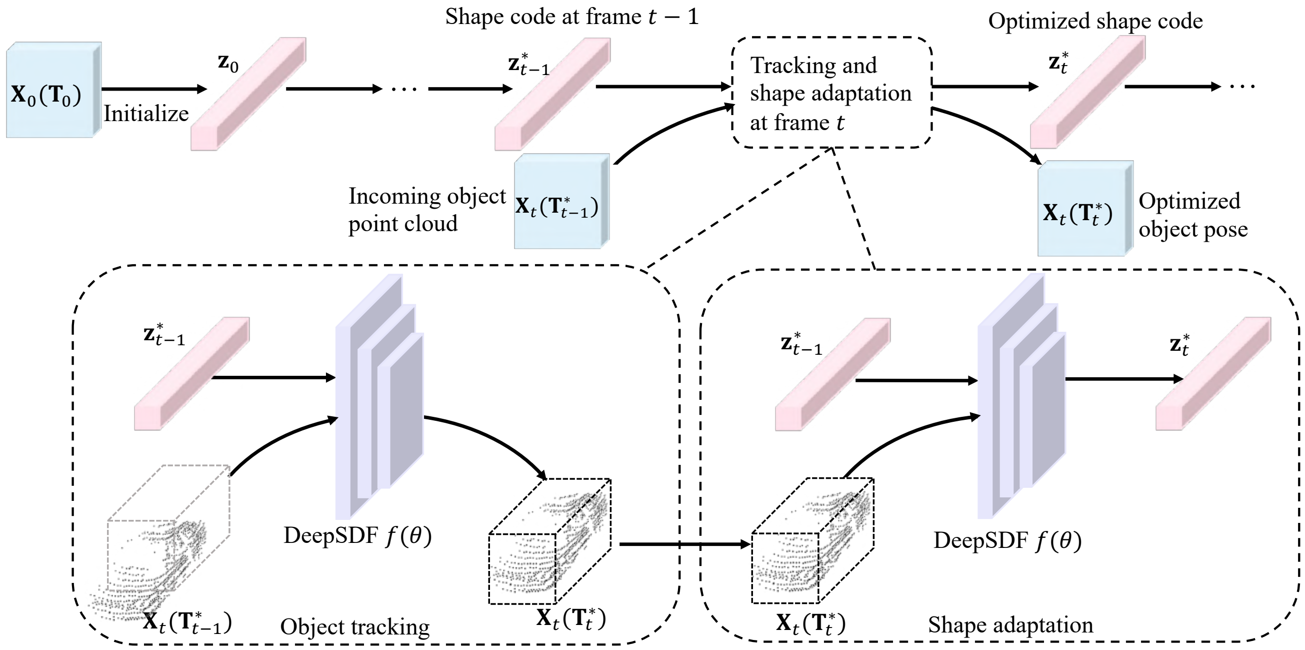

Shape code at frame $t-1$
Optimized shape code
$\mathbf{X}_0(\mathbf{T}_0)$
Initialize
$\mathbf{z}_0$
$\mathbf{z}^*_{t-1}$
Tracking and shape adaptation at frame t
$\mathbf{z}^*_t$
Incoming object point cloud
$\mathbf{X}_t(\mathbf{T}^*_{t-1})$
$\mathbf{X}_t(\mathbf{T}^*_t)$
Optimized object pose
$\mathbf{z}^*_{t-1}$
DeepSDF $f(\theta)$
$\mathbf{X}_t(\mathbf{T}^*_{t-1})$
Object tracking
$\mathbf{X}_t(\mathbf{T}^*_t)$
$\mathbf{z}^*_{t-1}$
$\mathbf{z}^*_t$
DeepSDF $f(\theta)$
$\mathbf{X}_t(\mathbf{T}^*_t)$
Shape adaptation

Figure 2. **Overview of our method.** After initialization of the shape code, tracking and shape reconstruction are performed iteratively. At a specific frame, the incoming object point cloud is first aligned to the previous shape, and then the shape is adapted to the aligned point cloud. Both procedures are based on DeepSDF.

perform tracking and pose estimation of the object across time.

In this paper, instead of taking object tracking and shape reconstruction as two separate tasks, we propose to solve them simultaneously using one single model. We introduce a novel framework which utilizes DeepSDF for tracking. When tracking the partial point clouds, we use them to update the shape code for the DeepSDF online, which leads to better shape reconstruction. With a better shape model, it improves tracking at the same time. Our framework iterates between object tracking and online adaptation along the video to improve performance for both tasks.

To perform tracking with DeepSDF, we first optimize the shape code via back-propagation using the given localized partial point clouds in the initial time step. While the shape code itself might not give perfect reconstruction, it offers a shape prior to construct a complete signed distance field of the object as a template. We then perform object tracking via optimizing a differentiable template matching process. Specifically, given the current model and point clouds in a new time step, we apply a 3D transformation on the partial point clouds (3D translation and rotation) and feed them to the DeepSDF model to compute their signed distances. If the 3D transformation is correct, the distance should be close to zero. We perform optimization via back-propagation to minimize the absolute distance. Since the 3D transformation operation is differentiable, we can back-propagate the gradients through the point clouds to adjust the transformation. Once the object point clouds are localized with the correct pose, we can then use the point clouds to optimize the shape code for updating the object shape.

We perform our experiments with LiDAR video data with cluttered scenes in the wild. Our DeepSDF model is first trained in the ShapeNet [2] dataset to obtain the shape prior. We then perform online adaptation with the model on the Waymo [13] dataset. We demonstrate that our method not only achieves state-of-the-art performance on 3D object tracking, but also improves shape reconstruction at the same time.

2. Method

2.1. Overview

We present the overview of our method in figure 2. As shown at the top of the figure, during the tracking process, we maintain a dynamic, adaptable object shape which is represented by the shape code. Since the object pose is provided at the first frame, we perform a normal shape reconstruction based on DeepSDF as an initialization. At a specific frame t, given the previous shape and the incoming unaligned object point cloud, we perform joint tracking and shape adaptation to optimize both object pose and shape code.

In particular, as shown at the bottom of the figure, an iterative optimization is performed at frame t. We first align the point cloud with the previous shape by minimizing the absolute distance of the point cloud. Then we adapt the shape to match it with the aligned point cloud by similarly minimizing the absolute distance of the point cloud.

In the paper, we define a 3D bounding box of the interested object with pose $\mathbf{T} \in SE(3)$ and size $\mathbf{b} = \{h, w, l\} \in \mathbb{R}^3$ (height, width and length). We use $\mathbf{X}(\mathbf{T}, \mathbf{b}) = \{\mathbf{x}|\mathbf{T}\mathbf{x} \leq \mathbf{b}, \mathbf{x} \in \mathbf{P}\}$ to denote the subset of LiDAR points $\mathbf{x}$ of a frame $\mathbf{P}$ that are inside the object bounding box. Since the size $\mathbf{b}$ of object is known and fixed, we omit it for simplicity and use $\mathbf{X}(\mathbf{T})$ to denote the posed object points hereafter.

2.2. Implicit Function for Shape Representation

Vehicles in the wild come with a wide range of shapes and partial observations as shown in Figure 1, which introduces challenges to the pre-parameterized shape models of car [6, 9]. Instead, we adopt the non-parametric DeepSDF for simultaneously tracking and shape reconstruction in this work.

DeepSDF f is a coordinate-based MLP parameterized by θ which approximates the SDF. It takes both 3D coordinates $\mathbf{x}$ and a learnable per-object shape code $\mathbf{z}$ as inputs. The DeepSDF function can be represented as,

$$f(\mathbf{x}, \mathbf{z}; \theta) = s, \quad \mathbf{x} \in \mathbb{R}^3, \mathbf{z} \in \mathbb{R}^d, s \in \mathbb{R}, \tag{1}$$

which encodes a one-to-one mapping between shape code $\mathbf{z}$ and 3D shape, and shape reconstruction for different objects can be achieved by optimizing the shape code for each object.

Given a set of surface points $\mathbf{X}$ of a 3D shape (a LiDAR scan) and corresponding SDF values s, we can reconstruct the shape from $\mathbf{X}$ by obtaining the optimal shape code $\mathbf{z}^*$ in the learned shape space via minimizing the SDF loss,

$$\mathbf{z}^* = \arg\min_{\mathbf{z}} \sum_{\mathbf{x} \in \mathbf{X}} \texttt{smooth\_l1}(f(\mathbf{x}, \mathbf{z}; \theta), s). \tag{2}$$

Note that we only optimize the shape code $\mathbf{z}$ here and we fix the network parameters θ.

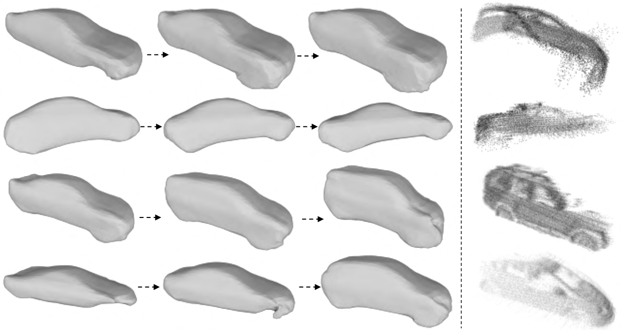
2.3. Joint Tracking and Shape Reconstruction in the Wild

Different from standalone shape reconstruction where dense surface points of objects are already well-posed in the canonical view, joint tracking and shape reconstruction in the wild requires us to estimate the object pose, making it far more challenging.

To track an object in 3D is to estimate its pose $\mathbf{T} \in SE(3)$ with respect to the LiDAR reference frame, as the size of the object $\mathbf{b} = \{h, w, l\}$ is already provided by the tracking template. Our joint tracking and shape reconstruction problem could be cast as

$$\min_{\mathbf{T}, \mathbf{z}} \sum_{\mathbf{x} \in \mathbf{X}(\mathbf{T})} \texttt{smooth\_l1}(f(\mathbf{x}, \mathbf{z}; \theta), 0). \tag{3}$$

Here all ground-truth SDF values are 0 as LiDAR points always come from the surface of an object.


(a) Shape evolution

(b) GT

Figure 3. **Shape evolution on Waymo.** The shapes in (a) are become more and more aligned with ground-truth point clouds in (b) via online adaptation.

2.4. Iterative Optimization

Directly solving Eq. 3 leads to sub-optimal results in practice as the pose $\mathbf{T}$ and the shape code $\mathbf{z}$ are two sets of variable with different scales and loss surfaces. Instead, we take an iterative optimization approach as described in Figure 2 to ease the difficulty of optimization.

During the first step of tracking, the initial object pose $\mathbf{T}_0$ is provided, so the corresponding optimal code $\mathbf{z}_0^*$ could be obtained by,

$$\mathbf{z}_0^* = \arg\min_{\mathbf{z}} \sum_{\mathbf{x} \in \mathbf{X}_0(\mathbf{T}_0)} \texttt{smooth\_l1}(f(\mathbf{x}, \mathbf{z}; \theta), 0) + \lambda \|\mathbf{z}\|_2^2. \tag{4}$$

A ℓ^2 regularizer is applied for the shape code $\mathbf{z}$ to prevent over-fitting. To perform tracking, we use the optimal shape code obtained from the last frame as,

$$\begin{aligned} \mathbf{T}_t^* = \arg\min_{\mathbf{T}} & \sum_{\mathbf{x} \in \mathbf{X}_t(\mathbf{T})} \texttt{smooth\_l1}(f(\mathbf{x}, \mathbf{z}_{t-1}^*; \theta), 0) \\ & + \phi(\mathbf{X}_t(\mathbf{T}), \bigcup_{i=0}^{t-1} \mathbf{X}_i(\mathbf{T}_i)). \end{aligned} \tag{5}$$

Here $\phi(\mathbf{X}, \mathbf{Y}) = \sum_{\mathbf{x} \in \mathbf{X}} \min_{\mathbf{y} \in \mathbf{Y}} \|\mathbf{x} - \mathbf{y}\|_2^2$ is an optional single-side Chamfer distance loss. The Chamfer distance loss between the object points in current frame and aggregated object points from previous time steps helps the pose estimation by capturing finer details of the object shape, as reconstructed shapes are tend to be over-smoothed.

After aligning the point clouds, we perform shape adaptation to utilize the observations to improve the shape quality. Specifically, we align the shape and the historical observations by minimizing the difference between them.

The shape adaptation is achieved via optimizing the shape code as,

$$\begin{aligned} \mathbf{z}_t^* = \arg\min_{\mathbf{z}} & \sum_{\mathbf{x} \in \bigcup_{i=0}^{t-1} \mathbf{X}_i(\mathbf{T}_i)} \texttt{smooth\_l1}(f(\mathbf{x}, \mathbf{z}; \theta), 0) \\ & + \lambda \|\mathbf{z}\|_2^2, \end{aligned} \tag{6}$$

		All		Easy		Medium		Hard	
Method	Adapted frames	Succ ↑	Prec ↑	Succ ↑	Prec ↑	Succ ↑	Prec ↑	Succ ↑	Prec ↑
SOTracker [11]	—	57.4	64.9	69.4	**75.7**	53.6	60.7	47.1	56.5
Ours	1	58.7	61.7	67.6	69.7	55.2	57.9	51.4	56.3
Ours	20	59.4	62.7	67.8	70.0	55.8	58.7	53.3	58.4
Ours	all	60.0	63.4	68.3	70.4	56.5	59.3	54.0	59.3
Ours w. CD loss	all	**62.3**	**65.7**	**71.5**	74.1	**58.8**	**61.8**	**54.9**	**59.9**

Table 1. **Tracking performance on Waymo.** ↑(↓) means the performance is better with larger (smaller) values. Our method achieves results that are comparable with the baseline on the easy subset and outperforms it by a large margin on the hard subset.

		All		Easy		Medium		Hard	
Method	Adapted frames	ACD ↓	Rec ↑	ACD ↓	Rec ↑	ACD ↓	Rec ↑	ACD ↓	Rec ↑
SOTracker [11]	—	2.81	82.25	2.49	84.51	2.87	82.31	3.09	79.92
Ours	1	2.43	85.39	2.35	86.51	2.71	84.60	2.22	85.06
Ours	20	**2.36**	**85.91**	**2.33**	**86.60**	**2.60**	**85.42**	**2.14**	**85.72**
Ours	all	2.42	84.64	2.30	84.81	2.64	84.39	2.31	84.72
Ours w. CD loss	all	2.50	84.71	2.30	86.40	2.76	83.78	2.44	83.98

Table 2. **Reconstruction performance on Waymo.** Our method performs better with online adaptation.

3. Experiments

We perform experiments on the Waymo open dataset [13] to evaluate both object tracking and shape reconstruction.

3.1. Experimental Settings

Data preparation. For training of the DeepSDF model, we use 2364 synthetic objects from the "car" category of the ShapeNet Core [2] dataset. For the supervision, we directly employ the processed SDF samples provided by DISN [14].

For tracking on Waymo, we follow the protocol in LiDAR-SOT [11]. The dataset is spilt into easy, medium, and hard subsets based on the average number of points of each tracklet.

Evaluation metrics. Since the ground-truth aggregated points are not complete, we employ Asymmetric Chamfer Distance (ACD) and Recall defined in SRW [5] to measure the shape fidelity.

In the paper, we set threshold of Recall $t = 0.2$. For the single object tracking task, we adopt Success and Precision defined in SC3D [7] to evaluate the tracking performance.

3.2. Experimental results

We compare our method with SOTracker, a recent single 3D object tracking method on Waymo. As shown in Table 1, our method achieves comparable results on the easy subset and outperforms baseline by a large margin on the hard subset. Since the difficulty is determined by the sparsity of point clouds, it's reasonable that shape reconstruction is more beneficial in the case of more sparse point clouds. Note that the objective function of SOTracker is the weighted sum of several terms: ICP term, shape term, motion consistency term, and motion prior Term. In our experiments on Waymo, we only utilize ICP term and shape term.

We also evaluate the performance of the shape reconstruction. Since shapes in SOTracker are represented by aggregated point clouds, metrics including ACD and Recall are not suitable. We first aggregate point clouds based on the predicted pose and then exploit our shape model to convert the point clouds to mesh for comparison. As shown in Table 2, our method outperforms the baseline on every subset. This experiment demonstrates that joint tracking and shape reconstruction leads to better performance than tracking followed by shape reconstruction.

To demonstrate the effectiveness of the online adaptation mechanism, we adapt shape during frames of different lengths and compare the performance of tracking and reconstruction. As shown in Table 1, the tracking performance continues to improve as the number of adaptation frames increases. Similarly, as shown in in Table 2, online adaptation also improves the quality of the shape. Figure 3 illustrates our shape evolution during tracking process on Waymo. We also find that as the number of adaptation increases, the gains for tracking and reconstruction performance become less. This suggests that as tracking proceeds, the pose is more likely to be estimated incorrectly, and adaptations at these frames are more likely to be noisy.

In addition, we ablate the Chamfer distance loss (CD loss, second term in Eq. 5) in Table 1 and Tabel 2 and observe it plays an important role in improving tracking.

4. Conclusions

In this paper, we present a novel and unified framework for object tracking and shape reconstruction in the wild. We propose to leverage the continuity in video data with a shape model. Specifically, we utilize a DeepSDF model to simultaneously perform object tracking and 3D reconstruction. During tracking process, we adapt shape model based on new observation to improve the shape quality, which leads to improvement on tracking, and vice versa. We perform experiments on Waymo, and outperform state-of-the-art methods by a large margin.

References

[1] 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 *European Conference on Computer Vision*, pages 608–625. Springer, 2020. 1

[2] 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. 1, 2, 4

[3] Zhiqin Chen and Hao Zhang. Learning implicit fields for generative shape modeling. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5939–5948, 2019. 1

[4] Angela Dai, Angel X Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5828–5839, 2017. 1

[5] Shivam Duggal, Zihao Wang, Wei-Chiu Ma, Sivabalan Manivasagam, Justin Liang, Shenlong Wang, and Raquel Urtasun. Secrets of 3d implicit object shape reconstruction in the wild. *arXiv preprint arXiv:2101.06860*, 2021. 4

[6] Francis Engelmann, Jörg Stückler, and Bastian Leibe. Joint object pose estimation and shape reconstruction in urban street scenes using 3d shape priors. In *German Conference on Pattern Recognition*, pages 219–230. Springer, 2016. 3

[7] Silvio Giancola, Jesus Zarzar, and Bernard Ghanem. Leveraging shape completion for 3d siamese tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 1359–1368, 2019. 4

[8] Chiyu Jiang, Avneesh Sud, Ameesh Makadia, Jingwei Huang, Matthias Nießner, Thomas Funkhouser, et al. Local implicit grid representations for 3d scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6001–6010, 2020. 1

[9] Lei Ke, Shichao Li, Yanan Sun, Yu-Wing Tai, and Chi-Keung Tang. Gsnet: Joint vehicle pose and shape reconstruction with geometrical and scene-aware supervision. In *European Conference on Computer Vision*, pages 515–532. Springer, 2020. 3

[10] Lars Mescheder, Michael Oechsle, Michael Niemeyer, Sebastian Nowozin, and Andreas Geiger. Occupancy networks: Learning 3d reconstruction in function space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4460–4470, 2019. 1

[11] Ziqi Pang, Zhichao Li, and Naiyan Wang. Model-free vehicle tracking and state estimation in point cloud sequences. *arXiv preprint arXiv:2103.06028*, 2021. 4

[12] Jeong Joon Park, Peter Florence, Julian Straub, Richard Newcombe, and Steven Lovegrove. Deepsdf: Learning continuous signed distance functions for shape representation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 165–174, 2019. 1

[13] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2446–2454, 2020. 2, 4

[14] Qiangeng Xu, Weiyue Wang, Duygu Ceylan, Radomir Mech, and Ulrich Neumann. Disn: Deep implicit surface network for high-quality single-view 3d reconstruction. *arXiv preprint arXiv:1905.10711*, 2019. 4