Robust 3D Tracking with Quality-Aware Shape Completion

Jingwen Zhang[1]*, Zikun Zhou[2]*†, Guangming Lu[1], Jiandong Tian[3], Wenjie Pei[1]†

[1]Harbin Institute of Technology, Shenzhen
[2]Peng Cheng Laboratory
[3]Shenyang Institute of Automation, Chinese Academy of Sciences
{jingwenz1022, zhouzikunhit}@gmail.com, luguangm@hit.edu.cn, tianjd@sia.cn, wenjiecoder@outlook.com

Abstract

3D single object tracking remains a challenging problem due to the sparsity and incompleteness of the point clouds. Existing algorithms attempt to address the challenges in two strategies. The first strategy is to learn dense geometric features based on the captured sparse point cloud. Nevertheless, it is quite a formidable task since the learned dense geometric features are with high uncertainty for depicting the shape of the target object. The other strategy is to aggregate the sparse geometric features of multiple templates to enrich the shape information, which is a routine solution in 2D tracking. However, aggregating the coarse shape representations can hardly yield a precise shape representation. Different from 2D pixels, 3D points of different frames can be directly fused by coordinate transform, i.e., shape completion. Considering that, we propose to construct a synthetic target representation composed of dense and complete point clouds depicting the target shape precisely by shape completion for robust 3D tracking. Specifically, we design a voxelized 3D tracking framework with shape completion, in which we propose a quality-aware shape completion mechanism to alleviate the adverse effect of noisy historical predictions. It enables us to effectively construct and leverage the synthetic target representation. Besides, we also develop a voxelized relation modeling module and box refinement module to improve tracking performance. Favorable performance against state-of-the-art algorithms on three benchmarks demonstrates the effectiveness and generalization ability of our method.

Introduction

3D object tracking in LiDAR point clouds aims to predict the target position and orientation in subsequent frames, given the initial state of the target object. Existing 3D trackers (Giancola, Zarzar, and Ghanem 2019; Qi et al. 2020; Hui et al. 2022; Shan et al. 2021; Zhou et al. 2022) predominantly follow the Siamese tracking paradigm (Bertinetto et al. 2016; Zhou et al. 2021), which has achieved astonishing success in 2D tracking. The pioneering study SC3D (Giancola, Zarzar, and Ghanem 2019) calculates the feature similarities between the template and randomly sampled candidates to track the target. After that, many advanced techniques are introduced to improve 3D tracking performance, including end-to-end tracking framework (Qi et al. 2020), transformer-based relation modeling (Shan et al. 2021; Zhou et al. 2022), box-aware feature fusion (Zheng et al. 2021), and contextual information modeling (Xu et al. 2023a; Guo et al. 2022).

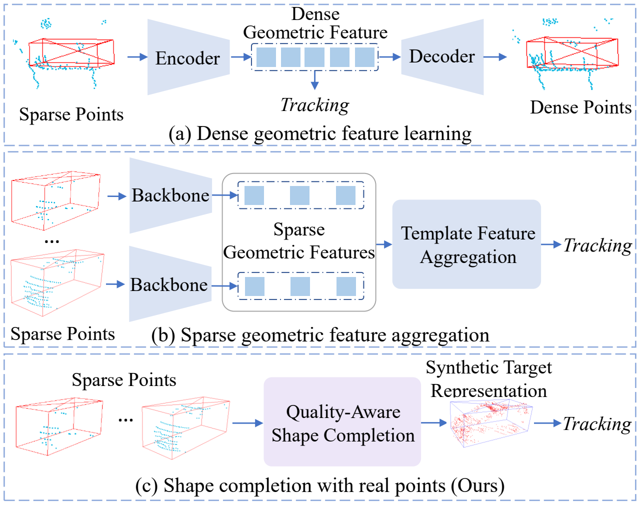


Figure 1: Different methods for addressing the challenges of sparsity and incompleteness. (a) Learning dense geometric features based on sparse points, which is a formidable task as the learned dense geometric features are with high uncertainty. (b) Aggregating the sparse geometric features of multiple templates, which is a sub-optimal solution as combining coarse shape representations can hardly obtain a precise shape. (c) Our method, which performs shape completion by adaptively fusing the real points of the target object from multiple frames to depict its shape precisely.

Despite the great progress, many existing trackers (Qi et al. 2020; Hui et al. 2022; Xu et al. 2023a; Shan et al. 2021; Zhou et al. 2022) pay less attention to the sparsity and incompleteness of the point clouds, which are usually caused by limited sensor capabilities and self-occlusion. For example, 51% of cars in the KITTI (Geiger, Lenz, and Urtasun 2012) dataset have less than 100 points. A typical challenging case is that only a few points of the template and the current target are overlapped due to the sparsity and incom-

[*] These authors contributed equally.

[†] Zikun Zhou and Wenjie Pei are corresponding authors.

pleteness, in which accurately matching the template with the real target is quite difficult. As a result, these methods struggle to discriminate the target in extremely sparse and incomplete point clouds.

Several methods have been proposed to address the challenges of sparsity and incompleteness. SC3D (Giancola, Zarzar, and Ghanem 2019) and V2B (Hui et al. 2021) adopt a strategy of learning dense geometric features based on sparse point clouds, as shown in Figure 1 (a). However, such a learning task is quite formidable since the learned dense geometric features are with high uncertainty. The trackers take the risk of misleading by the inaccurate dense features. TAT (Lan, Jiang, and Xie 2022) chooses to aggregate the sparse geometric features of multiple templates to obtain richer target shape information, which is a routine solution in 2D tracking (Wang et al. 2021; Zhang et al. 2019), as shown in Figure 1 (b). Although this strategy allows the tracker to take more target points into account, aggregating the coarse shape representations extracted from sparse points can hardly generate a precise shape representation. Hence, this aggregation strategy is a sub-optimal solution for addressing the challenges of sparsity and incompleteness.

Unlike the 2D image pixels, sparse 3D point clouds from different frames can be efficiently fused through coordinate transform to create a dense point cloud. Therefore, we propose to perform shape completion by fusing the target points from historical frames to construct a synthetic target representation for 3D tracking, as illustrated in Figure 1 (c). Herein the synthetic target representation consists of dense and complete point clouds depicting the shape of the target object precisely, enabling us to address the challenges of sparsity and incompleteness in 3D tracking.

In light of this idea, we design a robust 3D tracking framework that maintains a synthetic target representation by **s**hape **c**ompletion and performs 3D tracking in a **v**oxelized manner, termed SCVTrack. The tricky part of SCVTrack is that the synthetic target representation is sensitive to inaccurate historical predictions, and a noisy synthetic target representation can easily lead to tracking collapse. To alleviate the adverse effect of historical prediction errors, we propose a quality-aware shape completion module, which selectively fuses the well-aligned source points into the synthetic target representation. The shape completion naturally causes the imbalance between the point clouds of the template and the search area in terms of point density. It increases the difficulty of learning to model the relation between the two sets of point clouds. Therefore, we perform tracking based on the voxelized features instead of the point features to eliminate the imbalance. Besides, the voxelized tracking framework enables us to explicitly exploit the neighbor relation between voxels and is more computationally efficient. We also introduce a box refinement approach to further exploit the synthetic target representation to refine the target box, effectively improving tracking performance.

To conclude, we make the following contributions: (1) we propose a voxelized 3D tracking framework with shape completion to effectively leverage the real target points from historical frames to address the challenges of sparsity and incompleteness; (2) we design a quality-aware shape completion mechanism, taking the quality of the points into account for shape completion to alleviate the adverse effect of historical prediction errors; (3) we achieve favorable 3D tracking performance against state-of-the-art algorithms on three datasets, demonstrating the effectiveness of our method.

Related Work

3D object tracking. Early 3D trackers (Asvadi et al. 2016; Bibi, Zhang, and Ghanem 2016; Liu et al. 2018) based on RGB-Depth image pairs are vulnerable to lighting conditions that affect the RGB imaging quality. Recently, 3D tracking based on point clouds has drawn much more attention, as point clouds are robust to illumination changes. Most existing 3D trackers (Giancola, Zarzar, and Ghanem 2019; Qi et al. 2020; Fang et al. 2020; Zhou et al. 2022; Hui et al. 2022; Guo et al. 2022; Xu et al. 2023b) based on point clouds follow the Siamese tracking pipeline, which formulates 3D tracking as a template-candidate matching problem. Besides the Siamese tracking pipeline, M^{2}T (Zheng et al. 2022) recently proposes a motion-centric tracking paradigm, which directly predicts the target motion between two consecutive frames and achieves promising tracking performance. Despite the astonishing progress, the sparsity and incompleteness of 3D point clouds still plague these trackers.

An existing typical strategy to address the sparsity and incompleteness challenges is learning dense geometric features based on the given sparse point clouds. SC3D (Giancola, Zarzar, and Ghanem 2019) and V2B (Hui et al. 2021) are two methods following this strategy. However, such a learning task is quite challenging since the learned dense geometric features are with high uncertainty. As a result, these two methods only achieve limited tracking performance. A recently proposed approach, TAT (Lan, Jiang, and Xie 2022), adopts a multi-frame point feature aggregation strategy to enrich the shape information. Although it can alleviate the effect of sparsity and incompleteness, it is still a sub-optimal solution since aggregating the coarse shape representations extracted from sparse points can hardly generate a precise shape representation. Unlike the above methods, our method directly fuses the real target points from historical frames to construct a synthetic target representation for addressing the sparsity and incompleteness challenges.

Voxel-based 3D vision. Most existing 3D trackers (Giancola, Zarzar, and Ghanem 2019; Fang et al. 2020; Shan et al. 2021) follow the point-based deep learning paradigm (Li et al. 2018; Yang et al. 2020), performing tracking using the unordered point-based features. The voxel-based (Liu et al. 2019; Qi et al. 2016; Zhou and Tuzel 2018; Yin, Zhou, and Krahenbuhl 2021) learning paradigm is another popular way to process point data, which has been widely applied in 3D detection (Zhou and Tuzel 2018; Yan, Mao, and Li 2018; Lang et al. 2019; Yin, Zhou, and Krahenbuhl 2021). However, it has rarely been explored in 3D tracking, except for V2B (Hui et al. 2021) and MTM (Li et al. 2023). This paradigm assigns the points into different voxel bins and extracts structured voxelized features from unordered point clouds. In this work, we resort to voxelized relation modeling to deal with the imbalance issue in terms of point density due to shape completion.

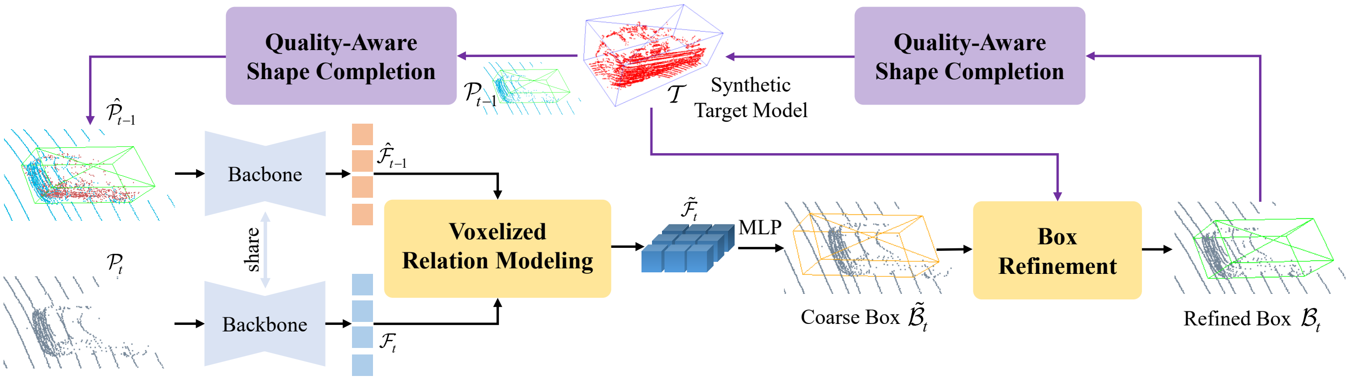


Figure 2: Overall framework of our SCVTrack, which mainly consists of a quality-aware shape completion module, a voxelized relation modeling module, and a box refinement module. It maintains a synthetic target representation $\mathcal{T}$ via quality-aware shape completion and performs 3D tracking in a voxelized manner. The red points in $\hat{\mathcal{P}}_{t-1}$ denote those coming from $\mathcal{T}$.

Method

Problem Definition

Given the initial 3D box of the target object, 3D tracking aims to estimate the target box in each subsequent frame. A 3D box is parameterized by its center position (xyz coordinate), orientation (heading angle θ around the up-axis), and size (width w, length l, and height h). The size of the target object, even for non-rigid objects like pedestrians and cyclists, remains approximately unchanged in 3D tracking. Thus, we only predict the translation $(\Delta x, \Delta y, \Delta z)$ and the rotation angle $(\Delta\theta)$ of the target object between two consecutive frames, and then obtain the 3D box $\mathcal{B}_t$ at t-th frame by transforming $\mathcal{B}_{t-1}$ with the translation and rotation angle.

Overall Tracking Framework

Figure 2 illustrates the overall framework of our SCVTrack. It mainly consists of the quality-aware shape completion, voxelized relation modeling, and box refinement modules. SCVTrack maintains a synthetic target representation $\mathcal{T}$ and performs tracking with it between two consecutive frames. Herein, the synthetic target representation is composed of dense and complete point clouds depicting the target shape precisely. We construct it by adaptively fusing the points belonging to the target from historical frames.

Suppose that the point clouds of the template and search area from two consecutive frames are denoted as $\mathcal{P}_{t-1} \in \mathbb{R}^{N_{t-1}\times 3}$ and $\mathcal{P}_t \in \mathbb{R}^{N_t\times 3}$, where N_{t-1} and N_t are the numbers of points. To localize the target in $\mathcal{P}_t$, our SCVTrack first completes $\mathcal{P}_{t-1}$ with the synthetic target representation $\mathcal{T}$ via quality-aware shape completion, yielding a completed template $\hat{\mathcal{P}}_{t-1} \in \mathbb{R}^{N'_{t-1}\times 3}$ with dense target points. Note that N'_{t-1} is usually much larger than both N_{t-1} and N_t due to the shape completion. With $\hat{\mathcal{P}}_{t-1}$ and $\mathcal{P}_t$, SCVTrack adopts a shared backbone to extract their point features $\hat{\mathcal{F}}_{t-1} \in \mathbb{R}^{N'_{t-1}\times C}$ and $\mathcal{F}_t \in \mathbb{R}^{N_t\times C}$ without downsample, respectively, where C is the feature dimension. Then SCVTrack voxelizes these point features and performs relation modeling between them to propagate the tracked target information from the template to the search area, generating the enhanced feature $\tilde{\mathcal{F}}_t$. An MLP regression head is constructed on top of $\tilde{\mathcal{F}}_t$ to predict a coarse box $\hat{\mathcal{B}}_t$. SCVTrack then performs box refinement with the guidance of $\mathcal{T}$ to obtain the refined box $\mathcal{B}_t$. After the tracking process, we use the new target points in $\mathcal{B}_t$ to update the synthetic target representation $\mathcal{T}$ via quality-aware shape completion.

Note that we opt to complete $\mathcal{P}_{t-1}$ with $\mathcal{T}$ to obtain a dense template instead of directly using T as the template. The rationale behind this design is that the target state in $\mathcal{P}_{t-1}$ is most similar to that in $\mathcal{P}_t$ in general, and completing $\mathcal{P}_{t-1}$ with $\mathcal{T}$ can not only obtain a dense template but also leverage all target points in $\mathcal{P}_{t-1}$ for tracking.

Quality-Aware Shape Completion

The quality-aware shape completion module aims to adaptively fuse the source point cloud $\mathcal{P}_{src}$ with the point cloud $\mathcal{P}_{tgt}$ to be completed. For generating a dense template, $\mathcal{T}$ is treated as the source point cloud, and $\mathcal{P}_{t-1}$ is treated as the point cloud to be completed. In turn, for updating $\mathcal{T}$, $\mathcal{P}_t$ is treated as the source point cloud, and $\mathcal{T}$ is treated as the point cloud to be completed. Figure 3 shows the shape completion process taking the completion for generating a dense template as an example. In the above two completion process, the source point cloud $\mathcal{P}_{src}$ is always obtained based on predicted target states, which inevitably contains noisy points. Directly using all points in $\mathcal{P}_{src}$ for shape completion will lead to error accumulation and even tracking collapse. To address this issue, we propose to evaluate the quality of the point clouds and perform selectively voxel-wise shape completion conditioned on the quality score.

Quality evaluator. To evaluate the quality of a point cloud, we design a quality evaluator consisting of a PointNet (Qi et al. 2017a) backbone and a three-layer MLP, which takes as input a point cloud and outputs a quality score. We formulate the quality evaluation task as a classification task. To be specific, we train the evaluator to differentiate the dense and well-aligned point clouds from the sparse and miss-aligned point clouds. To generate the required training samples, we first crop and center the points lying inside the object box from multiple frames. Then we concatenate these points to

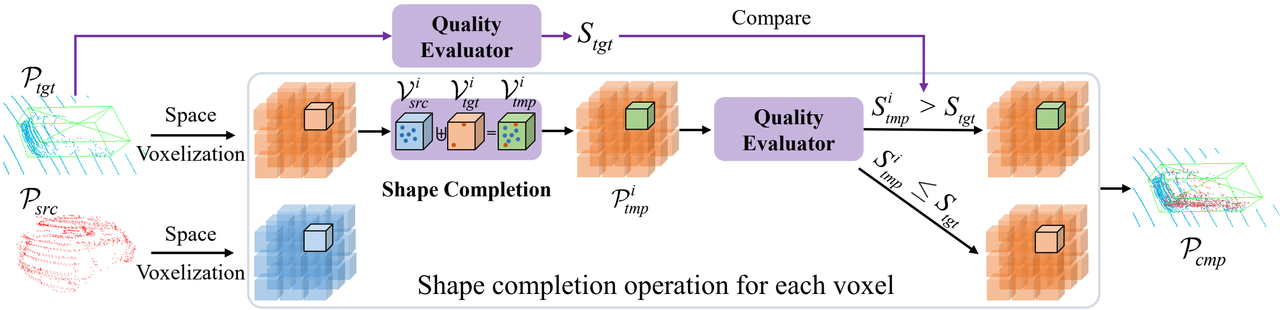


Figure 3: Illustration of the quality-aware shape completion module. $\uplus$ denotes the concatenation operation. This module performs selectively voxel-wise shape completion based on the output of the quality evaluator.

generate a dense and well-aligned point cloud as the positive sample. The negative sample is obtained by adding random position disturbance during concatenation or directly selecting a sparse point cloud from a certain frame. We use binary cross-entropy to train the quality evaluator. After training, the output logit is used as the quality score.

Voxel-wise shape completion. With the quality evaluator, we design a voxel-wise shape completion strategy to selectively complete different parts of $\mathcal{P}_{tgt}$ to alleviate the adverse effect of the noisy points. To this end, we voxelize the 3D space and assign the points in $\mathcal{P}_{tgt}$ and $\mathcal{P}_{src}$ into the corresponding voxel bin, as shown in Figure 3. The points of $\mathcal{P}_{tgt}$ and $\mathcal{P}_{src}$ lying inside the i-th voxel bin are denoted by $\mathcal{V}^i_{tgt}$ and $\mathcal{V}^i_{src}$. Before shape completion, we first evaluate the quality of $\mathcal{P}_{tgt}$, obtaining its quality score S_{tgt} as a reference. To complete the shape in the i-th voxel, we concatenate $\mathcal{V}^i_{src}$ with $\mathcal{V}^i_{tgt}$, yielding a dense point cloud $\mathcal{V}^i_{tmp}$ in the i-th voxel. We refer to the point cloud $\mathcal{P}_{tgt}$ whose i-th voxel is replaced with $\mathcal{V}^i_{tmp}$ as $\mathcal{P}^i_{tmp}$. Then we evaluate the quality of $\mathcal{P}^i_{tmp}$, obtaining a quality score S^i_{tmp}. After that, we compare S^i_{tmp} with S_{tgt} to judge whether the above completion in the i-th voxel improves the quality of the point cloud $\mathcal{P}_{tgt}$. Only when the quality is improved, we will update the points in the i-th voxel with $\mathcal{V}^i_{tmp}$. The above completion operation can be formulated as:

$$\begin{aligned} \mathcal{V}^i_{tmp} &= \mathcal{V}^i_{tgt} \uplus \mathcal{V}^i_{src}; \\ S^i_{tmp} &= \phi_{quality}(\mathcal{P}^i_{tmp}); \\ \mathcal{V}^i_{cmp} &= \begin{cases} \mathcal{V}^i_{tmp}, & if\ S^i_{tmp} > S_{tgt}; \\ \mathcal{V}^i_{tgt}, & else. \end{cases} \end{aligned} \tag{1}$$

Herein, $\uplus$ denotes the concatenation operation, $\phi_{quality}$ refers to the quality evaluator, $\mathcal{V}^i_{cmp}$ denotes the points in the i-th voxel of the final completed point cloud $\mathcal{P}_{cmp}$. Note that the voxel-wise shape completion can be done in a single forward propagation, as the above completion operation for different voxels can be performed in parallel.

Voxelized Relation Modeling

Taking as input the point features $\hat{\mathcal{F}}_{t-1}$ and $\mathcal{F}_t$, relation modeling aims to propagate the target information from the previous frame to the current one, generating the enhanced feature $\tilde{\mathcal{F}}_t$ for localizing the target. As above-mentioned, the motivations that we opt for voxelized relation modeling lie in eliminating the imbalance between $\hat{\mathcal{F}}_{t-1}$ and $\mathcal{F}_t$ and exploiting the neighbor relation explicitly. To this end, we first voxelize the point feature and then perform relation modeling between the voxelized feature, as shown in Figure 4.

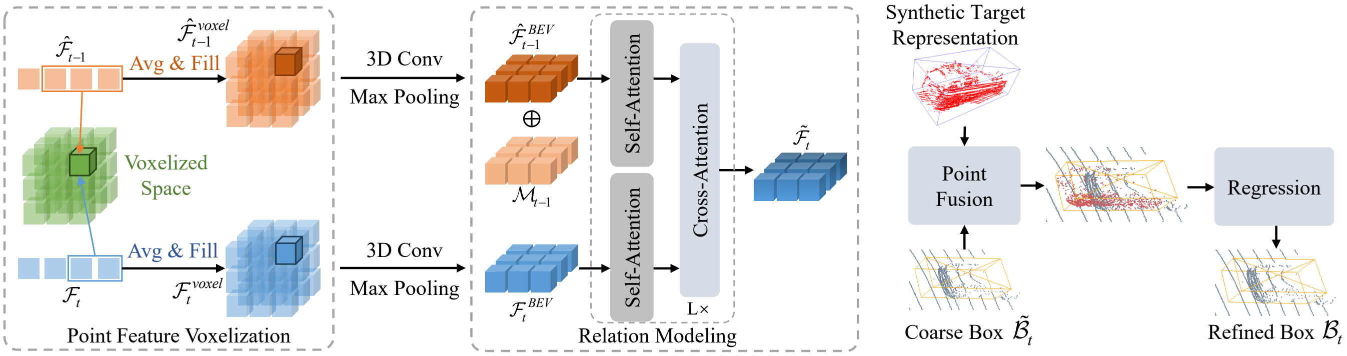
Point feature voxelization. We convert the point features $\hat{\mathcal{F}}_{t-1}$ and $\mathcal{F}_t$ into the voxelized representations $\hat{\mathcal{F}}^{vxl}_{t-1}$ and $\mathcal{F}^{vxl}_t$, respectively, by averaging the features of the points lying inside the same voxel bin. Then we apply shared 3D convolution layers to aggregate the shape information in the adjacent feature voxels to enhance the voxelized feature representations. Similar to (Hui et al. 2021), we perform max-pooling on these voxelized features along the z-axis to obtain the dense bird's eye view (BEV) features $\hat{\mathcal{F}}^{bev}_{t-1} \in \mathbb{R}^{H\times W\times C}$ and $\mathcal{F}^{bev}_t \in \mathbb{R}^{H\times W\times C}$ to alleviate the adverse effect the empty voxels.

Relation modeling. Considering that $\mathcal{P}_{t-1}$ contains both the target and background points, we introduce a learnable target mask $\mathcal{M}_{t-1} \in \mathbb{R}^{H\times W\times C}$ to embed the target state information into $\hat{\mathcal{F}}^{bev}_{t-1}$ before relation modeling. Technically, we introduce three learnable vectors indicating the three different positional states of a voxel, which are lying inside the box, outside the box, and across the box boundary. Then we generate the mask $\mathcal{M}_{t-1}$ according to the 2D projection box (along the z-axis) of the 3D box $\mathcal{B}_{t-1}$.

Inspired by recent advances (Ye et al. 2022; Zhou et al. 2023) in 2D tracking, we adopt an attention-based method to propagate the target information from $\hat{\mathcal{F}}^{bev}_{t-1}$ to $\mathcal{F}^{bev}_t$. As shown in Figure 2, a shared self-attention layer is first employed to model the intra-frame voxel relation. Then a cross-attention is used to model the cross-frame voxel relation, where the feature of the current frame is used as query and the feature of the previous frame is used as key and value. This process can be formulated as:

$$\tilde{\mathcal{F}}_t = \psi_{ca}(\psi_{sa}(\hat{\mathcal{F}}^{bev}_{t-1} \oplus \mathcal{M}_{t-1} \oplus \mathcal{E}), \psi_{sa}(\mathcal{F}^{bev}_t \oplus \mathcal{E})), \tag{2}$$

where ψ_{sa} and ψ_{ca} denote the self-attention and cross-attention, respectively. $\oplus$ means element-wise summation, and $\mathcal{E}$ refers to the position embedding. Note that we omit the flatten and reshape operation in Eq. 2, and this attention architecture is repeated L times.



(a) Voxelized Relation Modeling

(b) Box Refinement

Figure 4: Illustration of the voxelized relation modeling and box refinement modules. ⊕ denotes element-wise summation.

Box Refinement

The box refinement module aims to refine the coarse box $\tilde{\mathcal{B}}_t$ with the guidance of the dense geometric information in $\mathcal{T}$. To this end, we first fuse the dense points in $\mathcal{T}$ into the coarse box $\tilde{\mathcal{B}}_t$ by coordinate transform, obtaining a new point could $\hat{\mathcal{P}}_t$ depicting the target object with dense points. The offset between the coarse box $\tilde{\mathcal{B}}_t$ and the real target object will affect the smoothness of $\hat{\mathcal{P}}_t$. Based on this principle, we deploy a PointNet backbone following an MLP on top of $\hat{\mathcal{P}}_t$ to regress the above-mentioned offset to refine the target box.

End-to-end Modeling Learning

Our framework consists of two learnable parts: the quality evaluator and the remaining tracking model, which are trained separately. The quality evaluator is end-to-end trained as above-mentioned. The tracking model is end-to-end trained with pairs of consecutive frames. We impose smooth-$l1$ loss (Girshick 2015) on both the coarse and refined boxes to supervise the learning of the tracking model. Note that the synthetic target representation used in tracking model learning is pre-calculated with grounding truth.

Experiments

Experimental Setup

Implementation details. We use a modified PointNet++ (Qi et al. 2017b) as our backbone, which is tailored to contain three set-abstraction (SA) layers and three feature propagation (FP) layers. In the three SA layers, the sample radiuses are set to 0.3, 0.5, and 0.7, and the points are randomly sampled to 512, 256, and 128 points, respectively. Similar to (Zheng et al. 2022), we enlarge the target box predicted in the previous frame by 2 meters to obtain the search area in the current frame. We utilize the targetness prediction operation (Zheng et al. 2022) as a pre-process in our tracking framework. At the beginning of tracking, we use the target points lying inside the given box to initialize $\mathcal{T}$.

Benchmarks and metrics. We evaluate our algorithm on KITTI (Geiger, Lenz, and Urtasun 2012), NuScenes (Caesar et al. 2020), and Waymo Open Dataset (WOD) (Sun et al. 2020). KITTI consists of 21 training and 29 test sequences. We split the training set into train/validation/test splits as the test labels are inaccessible, following (Giancola, Zarzar, and Ghanem 2019; Zheng et al. 2022). NuScenes comprises 1000 scenes, which are divided into train/validation/test sets. Following (Zheng et al. 2021, 2022), we use the "train_track" split of the train set to train our model and test it on the validation set. WOD contains 1150 scenes, of which 798/202/150 scenes are used for training/validation/testing, respectively. We evaluate our method on WOD following two protocols: Protocol I (Xu et al. 2023a), where we directly test the KITTI pre-trained model on the validation set to evaluate generalization; Protocol II (Zheng et al. 2022), in which the model is trained on the training set and evaluated on the validation set. We use success and precision as metrics and report the Area Under Curve (AUC).

Ablation Studies

To analyze the effect of each component in SCVTrack, we conduct ablation experiments with six variants of SCVTrack: 1) the baseline (BL) model removing the shape completion mechanism and box refinement module from SCVTrack; 2) the variant using a naive shape completion mechanism without considering the point cloud quality into BL; 3) the variant performing quality-aware shape completion based on BL; 4) the variant that adopts the box refinement module based on the second variant; 5) our intact model; 6) the variant performing tracking with point features instead of voxelized features. This variant directly uses the attention-based method to process the point features and adopts an MLP head to regress the target box based on the output point features. Table 1 presents the experimental results.

Effect of the shape completion mechanism. The comparisons between the first three variants show that both the naive shape completion and quality-aware shape completion mechanisms can boost tracking performance. It manifests that performing shape completion in the raw point cloud space is an effective way to address the challenges of sparsity and incompleteness.

Effect of the quality evaluator. The performance gaps between the second and third variants and between the fourth and fifth variants demonstrate that the quality evaluator can substantially improve the quality of the synthetic target representation and further improve tracking performance.

Variants	Car	Cyclist	Van
1) BL	63.0 | 78.6	72.5 | 93.3	51.9 | 68.1
2) BL+NSC	65.2 | 78.0	73.6 | 93.5	54.9 | 70.1
3) BL+QASC	66.7 | 79.2	75.1 | 93.8	56.1 | 71.9
4) BL+NSC+BR	67.0 | 79.6	75.3 | 93.9	57.8 | 72.1
5) BL+QASC+BR	**68.7** | **81.9**	**77.4** | **94.4**	**58.6** | **72.8**
6) Ours w/o Vox.	64.5 | 79.6	74.5 | 93.6	55.2 | 71.0

Table 1: Ablation study results on the car, cyclist, and van categories. BL refers to the baseline model. NSC denotes naive shape completion. QASC is quality-aware shape completion. BR refers to box refinement. Vox. means voxelization. The best and second-best scores are marked in bold and underline, respectively. Success | Precision are reported.

	Car	Pedestrian	Van	Cyclist
SC3D	41.3 | 57.9	18.2 | 37.8	40.4 | 47.0	41.5 | 70.4
P2B	56.2 | 72.8	28.7 | 49.6	40.8 | 48.4	32.1 | 44.7
LTTR	65.0 | 77.1	33.2 | 56.8	35.8 | 45.6	66.2 | 89.9
BAT	60.5 | 77.7	42.1 | 70.1	52.4 | 67.0	33.7 | 45.4
PTT	67.8 | 81.8	44.9 | 72.0	43.6 | 52.5	37.2 | 47.3
PTTR	65.2 | 77.4	50.9 | 81.6	52.5 | 61.8	65.1 | 90.5
V2B	70.5 | 81.3	48.3 | 73.5	50.1 | 58.0	40.8 | 49.7
CMT	70.5 | 81.9	49.1 | 75.5	54.1 | 64.1	55.1 | 82.4
STNet	72.1 | 84.0	49.9 | 77.2	58.0 | 70.6	73.5 | 93.7
M^{2}T	65.5 | 80.8	61.5 | 88.2	53.8 | 70.7	73.2 | 93.5
TAT	72.2 | 83.3	57.4 | 84.4	58.9 | 69.2	74.2 | 93.9
CXT	69.1 | 81.6	67.0 | 91.5	60.0 | 71.8	74.2 | 94.3
MTM	73.1 | 84.5	**70.4** | **95.1**	60.8 | **74.2**	76.7 | **94.6**
MBPT	**73.4** | **84.8**	68.6 | 93.9	**61.3** | 72.7	76.7 | 94.3
Ours	68.7 | 81.9	62.0 | 89.1	58.6 | 72.8	**77.4** | 94.4

Table 2: Experimental results on KITTI.

Effect of the box refinement. Compared with the second and third variants, the fourth and fifth variants obtain large performance gains in the car and cyclist categories, respectively. It demonstrates the effectiveness of the box refinement guided by the synthetic target representation $\mathcal{T}$.

Effect of the voxelized tracking pipeline. Compared with our intact model, performing relation modeling and tracking with the imbalanced point features instead of the voxelized features, i.e., the sixth variant, results in substantial performance drops on these three categories. It validates that our voxelized tracking pipeline can deal with the aforementioned imbalance issue successfully but the point-feature-based tracking pipeline cannot.

Quantitative Results

The trackers involved in the comparison include SC3D (Giancola, Zarzar, and Ghanem 2019), P2B (Qi et al. 2020), LTTR (Cui et al. 2021), PTT (Shan et al. 2021), PTTR (Zhou et al. 2022) V2B (Hui et al. 2021), BAT (Zheng et al. 2021), STNet (Hui et al. 2022), M^{2}T (Zheng et al. 2022), CMT (Guo et al. 2022), TAT (Lan, Jiang, and Xie 2022), CXT (Xu et al. 2023a), MTM (Li et al. 2023), and MBPT (Xu et al. 2023b). We discuss the results per dataset.

	Car	Pedestrian	Van	Cyclist
	$\leq$150	$\leq$100	$\leq$150	$\leq$100
SC3D	37.9 | 53.0	20.1 | 42.0	36.2 | 48.7	50.2 | 69.2
P2B	56.0 | 70.6	33.1 | 58.2	41.1 | 46.3	24.1 | 28.3
BAT	60.7 | 75.5	48.3 | 77.1	41.5 | 47.4	25.3 | 30.5
V2B	64.7 | 77.4	50.8 | 74.2	46.8 | 55.1	30.4 | 37.2
M^{2}T	61.7 | 75.9	58.3 | 85.4	50.2 | 68.5	68.9 | 91.2
Ours	**64.8** | **77.7**	**60.1** | **88.6**	**52.8** | **70.5**	**70.2** | **92.8**

Table 3: Experimental results on sparse scenes of KIITI.

	Vehicle	Pedestrian	Mean
BAT†	54.7 | 62.7	18.2 | 30.3	34.1 | 44.4
V2B†	57.6 | 65.9	23.7 | 37.9	38.4 | 50.1
STNet†	59.7 | 68.0	25.5 | 39.9	40.4 | 52.1
TAT†	58.9 | 66.7	26.7 | 42.2	40.7 | 52.8
CXT†	57.1 | 66.1	30.7 | 49.4	42.2 | 56.7
Ours†	**61.3** | **69.8**	**32.2** | **50.0**	**44.8** | **58.6**

Table 4: Experimental results of different methods on WOD following Protocol I. † denotes the model is pre-trained on KITTI and directly evaluated on WOD validation split. These tracking results measure the generalization ability.

KITTI. Table 2 reports the experimental results on KITTI. V2B and TAT opt for dense geometric feature learning and sparse feature aggregation to address the sparsity and incompleteness challenges, respectively. Compared with them, our algorithm achieves better tracking performance in most categories. Our SCVTrack also outperforms M^{2}T in all categories, demonstrating its effectiveness. CXT and MBPT are two recently proposed trackers with sophisticated transformer blocks for relation modeling and target localization and perform better than our approach.

WOD. We first evaluate our SCVTrack on WOD following Protocol I to evaluate its generalization ability. Table 4 reports the experimental results. Compared with TAT and CXT, our SCVTrack achieves performance gains of 2.6% in mean success and 1.9% in mean precision. This comparison shows that our method obtains stronger generalization ability. We also evaluate SCVTrack on WOD following Protocol II. As shown in Table 5, our SCVTrack achieves the best performance in both vehicle and pedestrian categories.

NuScenes. Table 5 reports the experimental results on NuScenes. Our SCVTrack achieves the best success and precision score in the five categories. Compared with M^{2}T, our SCVTrack achieves performance gains of 2.9% in mean success and 2.0% in mean precision, demonstrating the effectiveness of our SCVTrack.

Quantitative results in sparse scenes. To investigate the effectiveness of our method in sparse scenes, we follow V2B to evaluate the performance in the sparse scenes (Car $\leq$ 150, Pedestrian $\leq$ 100, Van $\leq$ 150, and Cyclist $\leq$ 100) of KITTI, as shown in Table 3. Our SCVTrack performs favorably against the other methods in all categories.

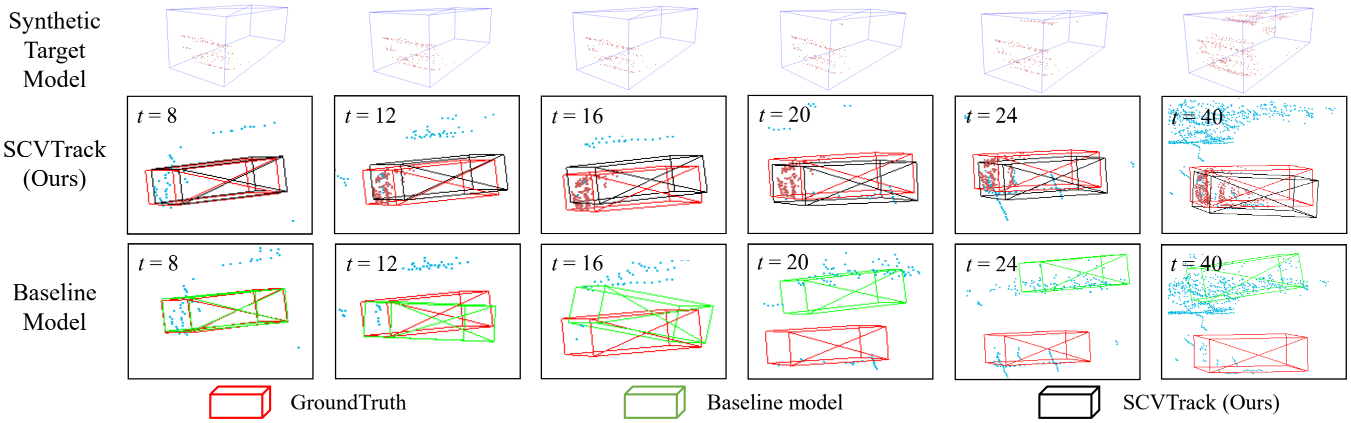


Figure 5: Qualitative comparisons between the variants w/ and w/o shape completion. Blue and red points refer to the raw points and fused points in each frame. We can observe that the shape completion mechanism helps SCVTrack successfully track the target in the extremely sparse scene, even though the synthetic target representation is not satisfactorily dense and complete.

	NuScenes						Waymo Open Dataset		
	Car	Pedestrian	Truck	Trailer	Bus	Mean	Vehicle	Pedestrian	Mean
SC3D	22.3 | 21.9	11.3 | 12.7	30.7 | 27.7	35.3 | 28.1	29.4 | 24.1	20.7 | 20.2	–	–	–
P2B	38.8 | 43.2	28.4 | 52.2	43.0 | 41.6	49.0 | 40.1	33.0 | 27.4	36.5 | 45.1	28.3 | 35.4	15.6 | 29.6	24.2 | 33.5
BAT	40.7 | 43.3	28.8 | 53.3	45.3 | 42.6	52.6 | 44.9	35.4 | 28.0	38.1 | 45.7	35.6 | 44.2	22.1 | 36.8	31.2 | 41.8
M^{2}T	55.9 | 65.1	32.1 | 60.9	57.4 | 59.5	57.6 | 58.3	51.4 | 51.4	49.2 | 62.7	43.6 | 61.6	42.1 | 67.3	43.1 | 63.5
Ours	**58.9 | 67.7**	**34.5 | 61.5**	**60.6 | 61.4**	**59.5 | 60.1**	**54.3 | 53.6**	**52.1 | 64.7**	**46.4 | 63.0**	**44.1 | 68.2**	**45.7 | 64.7**

Table 5: Experimental results of different methods on Nuscenes and WOD. These methods are trained on the training split of the Nuscenes or WOD benchmark and evaluated on the corresponding validation split.

	Pre-process	Shape completion	Pointnet++
Time	1.3 ms	11.1 ms	10.6 ms
	Voxelization	Relation modeling	Box refinement
Time	1.1 ms	5.6 ms	1.7 ms

Table 6: Inference time of each component of our model.

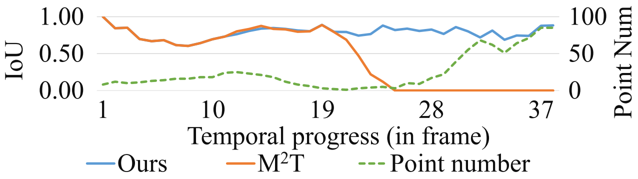


Figure 6: Results of M^{2}T and ours on a sparse scene.

Tracking speed. We measure the average tracking speed on Car of KITTI on an RTX3090 GPU, which is about 31 FPS. The average inference time per frame is 31.4 ms. Table 6 reports the detailed time consumption.

Qualitative Results

To further investigate the effectiveness of the shape completion mechanism, we visualize the tracking results of our SCVTrack and baseline model and the synthetic target representation in an extremely sparse scene, as shown in Figure 5. Although the synthetic target representation is not satisfactorily dense due to the extremely sparse point clouds, our SCVTrack keeps tracking the target successfully. By contrast, the baseline model loses the target then the point clouds become extremely sparse and incomplete. Figure 6 compares the tracking results of our method and M^{2}T (Zheng et al. 2022) in a sparse scene. They can both track the target at the beginning. M^{2}T loses the target at about the 20th frame (the target point cloud becomes quite sparse), while our method keeps tracking the target accurately.

Conclusion

In this work, we have presented a robust voxelized tracking framework with shape completion, named SCVTrack. Our SCVTrack constructs a dense and complete point cloud depicting the shape of the target precisely, i.e., the synthetic target representation, through shape completion and performs tracking with it in a voxelized manner. Specifically, we design a quality-aware shape completion mechanism, which can effectively alleviate the adverse effect of noisy historical predictions in shape completion. We also develop a voxelized relation modeling module and box refinement module to improve tracking performance. The proposed SCVTrack achieves favorable performance against state-of-the-art algorithms on three popular 3D tracking benchmarks.

Acknowledgements

This work was supported in part by the National Natural Science Foundation of China (Grant NO. U2013210, 62006060, 62372133, 62176077), in part by the Guangdong Basic and Applied Basic Research Foundation under Grant (Grant NO. 2022A1515010306), in part by Shenzhen Fundamental Research Program (Grant NO. JCYJ20220818102415032), in part by the Shenzhen Key Technical Project (NO. 2022N001, 2020N046), in part by the Guangdong International Science and Technology Cooperation Project (NO. 20220505), in part by the Guangdong Provincial Key Laboratory of Novel Security Intelligence Technologies (NO. 2022B1212010005), and in part by the Key Research Project of Peng Cheng Laboratory (NO. PCL2023A08).

References

Asvadi, A.; Girao, P.; Peixoto, P.; and Nunes, U. 2016. 3D object tracking using RGB and LIDAR data. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems*, 1255–1260. IEEE.

Bertinetto, L.; Valmadre, J.; Henriques, J. F.; Vedaldi, A.; and Torr, P. H. 2016. Fully-convolutional siamese networks for object tracking. In *European Conference on Computer Vision Workshops*, 850–865.

Bibi, A.; Zhang, T.; and Ghanem, B. 2016. 3d part-based sparse tracker with automatic synchronization and registration. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1439–1448.

Caesar, H.; Bankiti, V.; Lang, A. H.; Vora, S.; Liong, V. E.; Xu, Q.; Krishnan, A.; Pan, Y.; Baldan, G.; and Beijbom, O. 2020. nuscenes: A multimodal dataset for autonomous driving. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11621–11631.

Cui, Y.; Fang, Z.; Shan, J.; Gu, Z.; and Zhou, S. 2021. 3d object tracking with transformer. *arXiv preprint arXiv:2110.14921*.

Fang, Z.; Zhou, S.; Cui, Y.; and Scherer, S. 2020. 3d-siamrpn: An end-to-end learning method for real-time 3d single object tracking using raw point cloud. *IEEE Sensors Journal*, 21(4): 4995–5011.

Geiger, A.; Lenz, P.; and Urtasun, R. 2012. Are we ready for autonomous driving? the kitti vision benchmark suite. In *IEEE Conference on Computer Vision and Pattern Recognition*, 3354–3361. IEEE.

Giancola, S.; Zarzar, J.; and Ghanem, B. 2019. Leveraging shape completion for 3d siamese tracking. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1359–1368.

Girshick, R. 2015. Fast r-cnn. In *IEEE International Conference on Computer Vision*, 1440–1448.

Guo, Z.; Mao, Y.; Zhou, W.; Wang, M.; and Li, H. 2022. CMT: Context-Matching-Guided Transformer for 3D Tracking in Point Clouds. In *European Conference on Computer Vision*, 95–111. Springer.

Hui, L.; Wang, L.; Cheng, M.; Xie, J.; and Yang, J. 2021. 3D Siamese voxel-to-BEV tracker for sparse point clouds. *Advances in Neural Information Processing Systems*, 34: 28714–28727.

Hui, L.; Wang, L.; Tang, L.; Lan, K.; Xie, J.; and Yang, J. 2022. 3D Siamese transformer network for single object tracking on point clouds. In *European Conference on Computer Vision*, 293–310. Springer.

Lan, K.; Jiang, H.; and Xie, J. 2022. Temporal-Aware Siamese Tracker: Integrate Temporal Context for 3D Object Tracking. In *Asian Conference on Computer Vision*, 399–414.

Lang, A. H.; Vora, S.; Caesar, H.; Zhou, L.; Yang, J.; and Beijbom, O. 2019. Pointpillars: Fast encoders for object detection from point clouds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12697–12705.

Li, Y.; Bu, R.; Sun, M.; Wu, W.; Di, X.; and Chen, B. 2018. Pointcnn: Convolution on x-transformed points. *Advances in Neural Information Processing Systems*, 31.

Li, Z.; Lin, Y.; Cui, Y.; Li, S.; and Fang, Z. 2023. Motion-to-Matching: A Mixed Paradigm for 3D Single Object Tracking. *arXiv preprint arXiv:2308.11875*.

Liu, Y.; Jing, X.-Y.; Nie, J.; Gao, H.; Liu, J.; and Jiang, G.-P. 2018. Context-aware three-dimensional mean-shift with occlusion handling for robust object tracking in RGB-D videos. *IEEE Transactions on Multimedia*, 21(3): 664–677.

Liu, Z.; Tang, H.; Lin, Y.; and Han, S. 2019. Point-voxel cnn for efficient 3d deep learning. *Advances in Neural Information Processing Systems*, 32.

Qi, C. R.; Su, H.; Mo, K.; and Guibas, L. J. 2017a. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, 652–660.

Qi, C. R.; Su, H.; Nießner, M.; Dai, A.; Yan, M.; and Guibas, L. J. 2016. Volumetric and multi-view cnns for object classification on 3d data. In *IEEE Conference on Computer Vision and Pattern Recognition*, 5648–5656.

Qi, C. R.; Yi, L.; Su, H.; and Guibas, L. J. 2017b. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. *Advances in Neural Information Processing Systems*, 30.

Qi, H.; Feng, C.; Cao, Z.; Zhao, F.; and Xiao, Y. 2020. P2b: Point-to-box network for 3d object tracking in point clouds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6329–6338.

Shan, J.; Zhou, S.; Fang, Z.; and Cui, Y. 2021. PTT: Point-track-transformer module for 3D single object tracking in point clouds. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1310–1316. IEEE.

Sun, P.; Kretzschmar, H.; Dotiwalla, X.; Chouard, A.; Patnaik, V.; Tsui, P.; Guo, J.; Zhou, Y.; Chai, Y.; Caine, B.; et al. 2020. Scalability in perception for autonomous driving: Waymo open dataset. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2446–2454.

Wang, N.; Zhou, W.; Wang, J.; and Li, H. 2021. Transformer meets tracker: Exploiting temporal context for robust visual tracking. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1571–1580.

Xu, T.-X.; Guo, Y.-C.; Lai, Y.-K.; and Zhang, S.-H. 2023a. CXTrack: Improving 3D point cloud tracking with contextual information. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 1084–1093.

Xu, T.-X.; Guo, Y.-C.; Lai, Y.-K.; and Zhang, S.-H. 2023b. MBPTrack: Improving 3D Point Cloud Tracking with Memory Networks and Box Priors. In *IEEE/CVF International Conference on Computer Vision*, 9911–9920.

Yan, Y.; Mao, Y.; and Li, B. 2018. Second: Sparsely embedded convolutional detection. *Sensors*, 18(10): 3337.

Yang, Z.; Sun, Y.; Liu, S.; and Jia, J. 2020. 3dssd: Point-based 3d single stage object detector. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11040–11048.

Ye, B.; Chang, H.; Ma, B.; Shan, S.; and Chen, X. 2022. Joint feature learning and relation modeling for tracking: A one-stream framework. In *European Conference on Computer Vision*, 341–357. Springer.

Yin, T.; Zhou, X.; and Krahenbuhl, P. 2021. Center-based 3d object detection and tracking. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11784–11793.

Zhang, L.; Gonzalez-Garcia, A.; Weijer, J. V. D.; Danelljan, M.; and Khan, F. S. 2019. Learning the model update for siamese trackers. In *IEEE/CVF International Conference on Computer Vision*, 4010–4019.

Zheng, C.; Yan, X.; Gao, J.; Zhao, W.; Zhang, W.; Li, Z.; and Cui, S. 2021. Box-aware feature enhancement for single object tracking on point clouds. In *IEEE/CVF International Conference on Computer Vision*, 13199–13208.

Zheng, C.; Yan, X.; Zhang, H.; Wang, B.; Cheng, S.; Cui, S.; and Li, Z. 2022. Beyond 3d siamese tracking: A motion-centric paradigm for 3d single object tracking in point clouds. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8111–8120.

Zhou, C.; Luo, Z.; Luo, Y.; Liu, T.; Pan, L.; Cai, Z.; Zhao, H.; and Lu, S. 2022. Pttr: Relational 3d point cloud object tracking with transformer. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 8531–8540.

Zhou, L.; Zhou, Z.; Mao, K.; and He, Z. 2023. Joint Visual Grounding and Tracking With Natural Language Specification. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 23151–23160.

Zhou, Y.; and Tuzel, O. 2018. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *IEEE Conference on Computer Vision and Pattern Recognition*, 4490–4499.

Zhou, Z.; Pei, W.; Li, X.; Wang, H.; Zheng, F.; and He, Z. 2021. Saliency-Associated Object Tracking. In *IEEE/CVF international conference on computer vision*, 9866–9875.