MLVSNet: Multi-level Voting Siamese Network for 3D Visual Tracking

Zhoutao Wang [*1], Qian Xie [*1], Yu-Kun Lai[2], Jing Wu[2], Kun Long[1] and Jun Wang [†1]
[1]Nanjing University of Aeronautics and Astronautics
[2]Cardiff University

Abstract

Benefiting from the excellent performance of Siamese-based trackers, huge progress on 2D visual tracking has been achieved. However, 3D visual tracking is still under-explored. Inspired by the idea of Hough voting in 3D object detection, in this paper, we propose a Multi-level Voting Siamese Network (MLVSNet) for 3D visual tracking from outdoor point cloud sequences. To deal with sparsity in outdoor 3D point clouds, we propose to perform Hough voting on multi-level features to get more vote centers and retain more useful information, instead of voting only on the final level feature as in previous methods. We also design an efficient and lightweight Target-Guided Attention (TGA) module to transfer the target information and highlight the target points in the search area. Moreover, we propose a Vote-cluster Feature Enhancement (VFE) module to exploit the relationships between different vote clusters. Extensive experiments on the 3D tracking benchmark of KITTI dataset demonstrate that our MLVSNet outperforms state-of-the-art methods with significant margins. Code will be available at https://github.com/CodeWZT/MLVSNet.

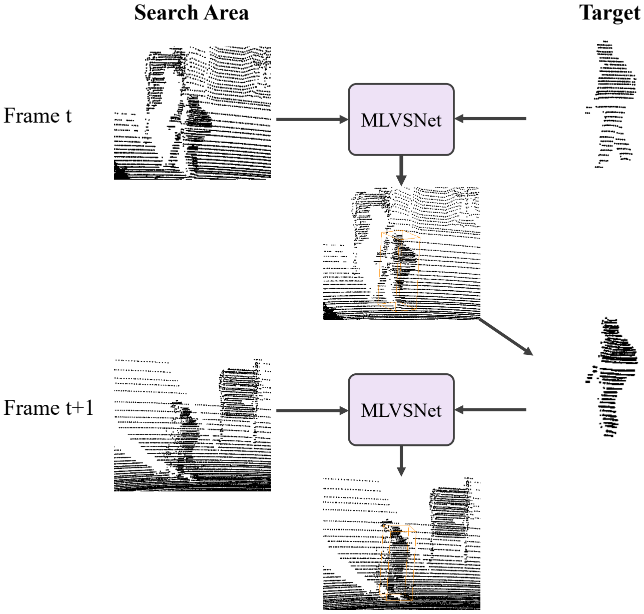


Figure 1. Illustration on how MLVSNet tracks a target on 3D point clouds. MLVSNet tracks the target object in the search area at each frame over a sequence of point clouds.

1. Introduction

Visual tracking aims to track a given target in every frame of a sequence. As shown in Fig. 1, a tracking algorithm takes as input a target and a search area, and outputs the location of the detected target in the search area, which also serves as the target for the next frame. Visual tracking is an indispensable part in robot vision and autopilot systems [43, 46, 8], and has long been a popular research topic in computer vision [23, 35]. Great progress has been made in 2D visual tracking community, benefiting from the excellent performance of Siamese based trackers [47, 11]. However, direct application of these trackers to 3D tracking is infeasible due to the different data structure. Compared to 2D visual tracking, 3D tracking uses point cloud data. It has the advantages of being more robust to illumination and appearance changes, but is also more challenging. Different from RGB images/videos, 3D point cloud data is irregular, noisy, and more sparse especially in outdoor environment. These impose challenges different from 2D visual tracking and require specific considerations in the algorithm design.

In this paper, we propose a novel Multi-level Voting Siamese Network (MLVSNet), an end-to-end 3D visual tracking method from point clouds in outdoor scenes. As illustrated in Fig. 1, at each frame, tracking is grounded on target detection. The recently proposed VoteNet [29] has demonstrated its effectiveness in detecting 3D objects from point clouds [42, 5, 33]. VoteNet is based on Hough voting, where the chosen points collect the geometric information from their surrounding points and then vote for the object centers. Inspired by the success of VoteNet, the proposed MLVSNet also employs the idea of Hough voting to locate targets in the search area. In addition to voting-based detection, the design of MLVSNet also takes into consideration

[*] These authors contributed equally to this work.
[†] Corresponding author: wjun@nuaa.edu.cn.

the following two challenges: 1) how to deal with the irregular, noisy and sparse 3D point cloud data; 2) how to efficiently transfer the information of the target to the search area for tracking.

To deal with the irregular data structure and noisy data capture, PointNet++ [30] was proposed recently and has shown great success in 3D object detection [40, 28, 41]. Its hierarchical feature learning effectively distills information and captures useful high-level features from the irregular and noisy point cloud data. However, associated with the high-level features is the reduced number of seed points, which worsens the already sparse points for representing targets in outdoor scene point clouds. We argue that both the number of seed points and the feature descriptive ability of points are important for target detection. To balance the two, in MLVSNet, we propose to make use of seed points and their representations at multiple levels, i.e., a multi-level voting strategy (MLV), to aggregate votes from seed points at multiple levels. We also argue that the multi-level aggregation captures information at different scales and actually improves the detection performance, as we will later show in experiments.

To transfer the target information for tracking, the state-of-the-art method [31] makes use of a series of MLP (Multi-Layer Perceptron) layers to embed target features into the feature map of the search area. However, such operations are both memory and time inefficient especially when combined with the multi-level voting strategy. Attention mechanism [38, 26] has been demonstrated an effective and efficient way to model relationship information [40]. We therefore propose a lightweight module, termed Target-Guided Attention (TGA) module, to establish the relationship between the search area and the target with much fewer parameters. Attention mechanism is also used in the proposed Vote-cluster Feature Enhancement (VFE) module to exploit the relationship between different vote clusters, which further improves the network performance.

All the above modules constitute the proposed Multi-level Voting Siamese Network (MLVSNet) for 3D tracking from point cloud sequences. Extensive experiments show that our model outperforms the state-of-the-art models [31, 14] by a large margin. In summary, this work makes the following contributions:

- We propose a multi-level voting (MLV) strategy to aggregate data and information at multiple levels for more effective target detection in sparse point clouds.

- We design a lightweight feature fusion module, named Target-Guided Attention (TGA), for efficient embedding of target information for tracking.

- We present the novel Multi-level Voting Siamese Network (MLVSNet) for 3D visual tracking on point clouds, which achieves the new state-of-the-art performance on benchmarks.

2. Related Work

In this section, we briefly introduce a recent wave of work related to our MLVSNet: 2D Siamese tracking, 3D visual tracking, and attention mechanism.

2.1. 2D Visual Tracking

Advances in 2D visual tracking approaches promote the development of 3D visual tracking in which the greatest influence is the Siamese-based tracker. Bertinetto *et al.* [2], for the first time, proposed SiamFC for image visual tracking. They employed a Siamese tracker to compute the similarity between the search area and targets. Then the work in [22] extended the SiamFC by introducing a region proposal network (RPN) into Siamese networks, denoted as Siamese-RPN. Benefiting from the RPN structure, Siamese-RPN can obtain more accurate bounding boxes and achieve high-speed processing. Li *et al.* [21] developed a SiamRPN++ that exploited a simple spatial aware sampling strategy to make the tracking network go deep. Zhang *et al.* [47] followed with interest in the shallow backbone problem, and proposed a residual module to overcome the negative impact of padding. Gao *et al.* [12] developed a Siamese lightweight hourglass network to achieve high performance and efficiency in real-world scenarios. At present, 2D tracking networks based on Siamese structure are still competitive [48, 37, 15, 9, 47, 16].

2.2. 3D Visual Tracking

The existing 3D visual tracking methods can be divided into two categories according to the form of input data: The first category of methods [27, 1, 19, 3, 18, 24] relies on RGB-D information. Pieropan *et al.* [27] developed a tracking method that can learn the appearance of unknown objects and track their positions and full 3D poses. The work in [1] introduced a tracking algorithm that employs RGB information, 3D point clouds, and localization data to predict the location of the target. Kart *et al.* [19] proposed a long-term RGB-D tracker to overcome the out-of-plane rotation challenge when modeling appearance changes. Relying on RGB-D data, this category of methods is thus not robust to changes in illumination and appearance. The second category of methods only takes point clouds as input. To the best of our knowledge, there are only a few works. Shape Completion 3D (SC3D) network [14] is the pioneering work in this category. It contains a Siamese tracker to encode targets and search areas into a latent representation for similarity computation. Despite the efforts, SC3D still has several limitations. First, SC3D needs to pre-train the shape completion network, which limits its convenience and

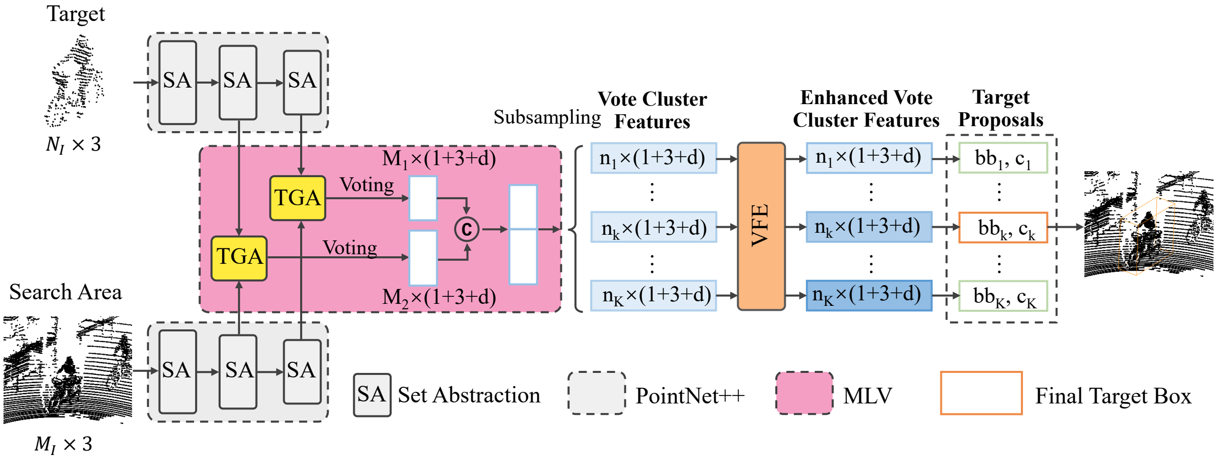


Figure 2. Network architecture of the proposed MLVSNet. MLVSNet is mainly composed of three modules: 🟨 TGA module is used to embed target information into the search area to obtain embedded features. 🟪 MLV module performs multi-level voting for effectively utilizing information at different levels. 🟧 VFE module is used to enhance features of vote clusters so that vote clusters can perceive each other. We select the box bb_k with the highest targetness confidence c_k as the final target box.

generality. Second, some intermediate processes, e.g. 3D search and proposal reasoning, limit its speed and accuracy. Point-to-Box (P2B) network [31] was the first breakthrough for end-to-end 3D visual tracking on point clouds. In order to transfer the target information to the search area, P2B proposed a Target-Specific Feature Augmentation (TSFA) module, composed of a series of MLP layers. Although it can effectively embed the target information, this architecture limits the efficiency and makes it difficult to further expand the framework. In addition to the above methods, Zarzar *et al.* [45] developed an efficient 3D Siamese tracking network focusing on 3D vehicle tracking. It takes birds-eye view images and 3D point clouds as input and employs an RPN network to achieve efficient search processing. P2B is the most pertinent to our method. Our method also takes point clouds as input and trains the tracking network end-to-end. Different from P2B, our method fuses the target information in a lightweight manner and aggregates information from multiple levels, which further improves the efficiency and performance of 3D tracking.

2.3. Attention Mechanism

Attention mechanism is often suggested as a means of improving deep neural network performance. It can imitate the processing mechanism of human visual system for massive data, i.e., focusing on the important part and discarding the rest. The work in [36, 17] employed an attention mechanism to improve the performance of large-scale image classification. Specifically, in [36], an encoder-decoder attention module was utilized to refine feature maps. In [17], the Squeeze-and-Excitation module was proposed, which adopted a compact module to enhance the channel relationship. Inspired by [17], Woo *et al.* [38] proposed a lightweight attention module called Convolutional Block Attention Module (CBAM) for convolutional neural networks (CNNs). CBAM utilized both spatial and channel-wise attention to refine intermediate feature maps. Owing to its lightweight and generality, it can be embedded in any CNNs and supports end-to-end training.

Attention mechanism is also widely used in 2D visual tracking [44, 10, 4, 7, 25, 6]. Yu *et al.* [44] proposed deformable self-attention and cross-attention to learn context information and contextual interdependencies from the target template and search image. Du *et al.* [10] proposed Correlation-Guided Attention (CGA) module for corner detection of target bounding boxes. CGA module mainly improves the performance of corner detection and achieves the high-speed processing at 70 FPS. In this work, we exploit the attention mechanism to embed target information into the search area to achieve efficient 3D visual tracking.

3. Network Architecture

Fig. 2 shows the network architecture of MLVSNet. MLVSNet is a Siamese network combined with Hough voting. It mainly consists of three modules: 1) Target-Guided Attention (TGA) module, 2) Multi-level Voting (MLV) module, and 3) Vote-cluster Feature Enhancement (VFE) module. It first takes as input the point clouds of a target and a search area and obtains their multi-level seeds through the PointNet++ backbone. Then, the TGA module embeds information of the target seeds into the seed features of the search area. After that, the embedded features at multiple levels vote for object centers through Hough voting. We select K voting points from all levels to form vote clusters, and utilize the VFE module to enhance their features. Finally, the bounding box of the target is inferred from the

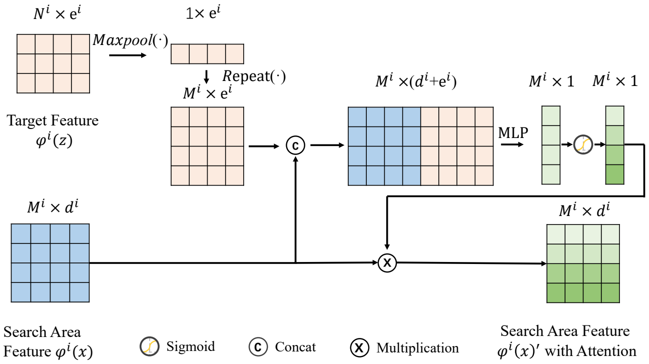


Figure 3. Illustration of Target-Guided Attention (TGA) module in MLVSNet. Our method employs an attention mechanism to embed target information into the search area in a lightweight manner.

vote clusters. We will discuss the details in the following subsections.

3.1. Target-Guided Attention

Target-Guided Attention (TGA) module provides a lightweight way to embed target information into the search area. We denote the target input as z and the search area input as x. We use PointNet++ [30] as the backbone, which consists of three Set Abstraction layers (as shown in Fig. 2). The backbone network shares parameters so that the two branches of input (x and z) are implicitly encoded by the same transformation. The structure of the TGA module is shown in Fig. 3. It is motivated by the Cross-frame Global Attention (CGA) in [34], where the attention mask for the current frame is generated from the information in the last frame. In our work, since our aim is to locate the target in the search area, features of the search area that match the target features should be highlighted, which we achieve through generating an attention mask.

Suppose at level i, we have M^i seed points $p^i(x) \in \mathbb{R}^{M^i \times 3}$ in the search area and N^i seed points $p^i(z) \in \mathbb{R}^{N^i \times 3}$ in the target. $\varphi^i(z) \in \mathbb{R}^{N^i \times e^i}$ and $\varphi^i(x) \in \mathbb{R}^{M^i \times d^i}$ are their corresponding features (d^i and e^i are the feature dimensions of $\varphi^i(x)$ and $\varphi^i(z)$, respectively). We first generate a vector $w^i \in \mathbb{R}^{M^i \times 1}$ from point features, i.e.,

$$w^i = MLP_{TGA}(Concat(\varphi^i(x), Repeat(Maxpool(\varphi^i(z))))). \tag{1}$$

As shown in Fig. 3, we adopt the maxpooling layer $Maxpool(\cdot)$ to change the size of target feature from $N^i \times e^i$ to $1 \times e^i$. Then, we combine the target feature with the search area feature through $Repeat(\cdot)$ and $Concat(\cdot)$. Finally, we use a three-layer MLP (MLP_{TGA}) to transform the size of merged feature from $M^i \times (d^i + e^i)$ into $M^i \times 1$. The TGA module aims to highlight important (i.e., the target) points in the search area. We therefore generate the point-wise attention map as:

$$Sig(w^i) = \frac{1}{1 + e^{-w^i}}. \tag{2}$$

where $Sig(\cdot)$ denotes the $Sigmod$ activation function to make the output normalized in $(0, 1)$. Then the features in the search area are enhanced by

$$\varphi^i(x)' = Sig(w^i) \otimes \varphi^i(x). \tag{3}$$

where $\otimes$ is the point-wise multiplication. TGA provides a lightweight way to find points that match the target via assigning different importance weights to points in the search area. That is, the TGA module aims to figure out which points in $\varphi^i(x)$ are important.

Permutation-invariance. For the target feature, we adopt symmetric functions (i.e. $Maxpool(\cdot)$) to ensure permutation-invariance. The order of w^i is consistent with search area points, so the point-wise multiplication is not affected by the order of search area points. Therefore, the proposed TGA module is permutation-invariant.

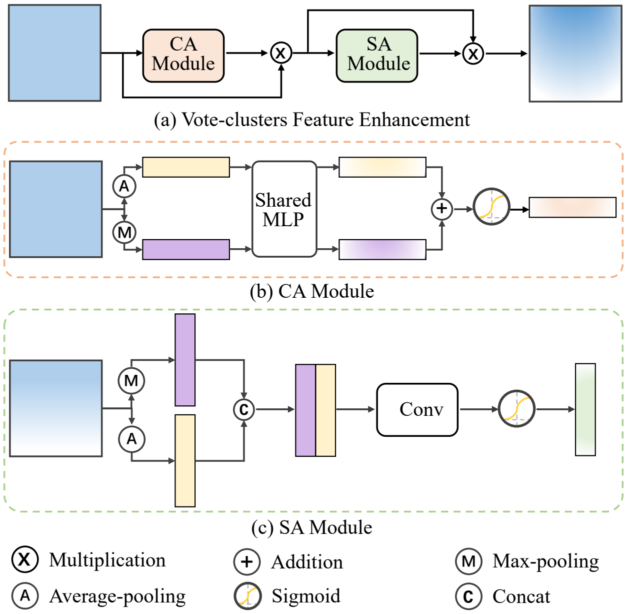
Comparison to TSFA. The TSFA module in P2B [31] also embeds target information into the search area. However, TSFA applies cosine distance to compute the point-wise similarity and obtains a similarity map Map_{sm} of size $[M^i \times N^i, 1]$. Moreover, a series of MLP layers and a Max-pooling operation are used to augment $\varphi^i(z)$ and $\varphi^i(x)$ in order to get the target-specific features. Although TSFA has achieved high performance, these operations have made it inefficient. In experiments, we compared the time and memory efficiency of TSFA with the proposed TGA (see Table 2).

3.2. Multi-level Voting

In the TGA module, we use target features $\varphi^i(z)$ to enhance the search features $\varphi^i(x)$. Now, our goal is to track the target based on the seed points $p^i(x)$ and their enhanced features $\varphi^i(x)'$. We can regard the subsequent task as a 3D object detection task. Here, we follow the idea of Hough voting in VoteNet [29] to detect the target. However, the multiple subsampling-based set abstraction operations in PointNet++ significantly reduce the number of seed points and make it difficult to get enough meaningful votes in the relatively sparse outdoor point clouds. Thus, we propose to perform the Hough voting at multiple levels, on the one hand, to bring in more seed points; on the other hand, to make use of multi-level features. The multi-level voting (MLV) is formulated as:

$$\begin{aligned} [\triangle p^{v,i}, \triangle f^{v,i}] &= HoughVoting([p^i(x), \varphi^i(x)']) \\ [p^{v,i}, f^{v,i}] &= [p^{v,i}(x), \varphi^i(x)'] + [\triangle p^{v,i}, \triangle f^{v,i}]. \end{aligned} \tag{4}$$

where $HoughVoting(\cdot)$ is realized with an MLP network with batch normalization and ReLU. It predicts the feature



(a) Vote-clusters Feature Enhancement

(b) CA Module

(c) SA Module

Figure 4. The structure of Vote-clusters Feature Enhancement (VFE) in MLVSNet. We use the modified CBAM module [38] to enhance the features of vote clusters. VFE module contains two sub-modules for channel attention (CA) and spatial attention (SA).

residual $\triangle f^{v,i}$ for $\varphi^i(x)'$ and the coordinate offset $\triangle p^{v,i}$ for $p^i(x)$. Then the coordinates $p^{v,i} \in \mathbb{R}^{M^i \times 3}$ and the features $f^{v,i} \in \mathbb{R}^{M^i \times d^i}$ are obtained. d^i is the feature dimension. M_i is the point number at the i-th level.

Multi-level votes $\{v_j = [p^v{}_j; f^v{}_j] \in \mathbb{R}^{3+d}\}_{j=1}^{\sum_i^I M^i}$ contain rich context information from the features at different levels. Gathering votes from all levels gives a relatively large number of seed points. Then, we sample a subset ($K = 64$) from $\{v_j\}_{j=1}^{\sum_i^I M^i}$ using random sampling. Following VoteNet, we employ the ball query to obtain sample points within radius r to the vote center to form vote clusters C_k:

$$C_k = \{v_j^{(k)} | \|v_j - v_{j_k}\| \leq r\}. \tag{5}$$

where $k = 1, ..., K$, $j = 1, ..., \sum_i^I M^i$. After that, we employ an MLP layer to update the feature of vote cluster. $C' = \{[p_k^v; f_k^{v\prime}] \in \mathbb{R}^{3+d}\}_{k=1}^K$ contains the coordinates and features for vote clusters. Instead of directly predicting the bounding boxes from C', we follow a common strategy that considers the relationships between the vote clusters for the final prediction, which is described in the following subsection.

3.3. Vote-cluster Feature Enhancement

We propose a Vote-cluster Feature Enhancement module, which is again based on an attention mechanism, to capture the relationships between vote clusters and enhance their features. By establishing the relationships, the final tracking box is determined not only by its individual vote cluster, but also the related vote clusters. In practice, the relationships are established by assigning weights between vote clusters. Specifically, we make use of the CBAM module [38] to enhance the perception between vote clusters. The original CBAM module is mainly used for images, so we modify it to make it suitable for enhancing point cloud features. Fig. 4 depicts the computation process in detail. We first utilize the channel attention to enhance the features of the vote clusters, and then employ the spatial attention to establish the relationships between vote clusters. The encoding of vote clusters relationships can be expressed as:

$$\begin{aligned} f^{v\prime\prime} &= A_c(f^{v\prime}) \otimes f^{v\prime} \\ f^{v\prime\prime\prime} &= A_s(f^{v\prime\prime}) \otimes f^{v\prime\prime}. \end{aligned} \tag{6}$$

where A_c, A_s respectively denote the channel attention map and the spatial attention map. $\otimes$ is the element-wise multiplication. Finally we send $C'' = [p_k^v; f_k^{v\prime\prime\prime}]_{k=1}^K$ into an MLP network to get the final prediction:

$$\{bb, c\} = MLP_{VFE}([p_k^v; f_k^{v\prime\prime\prime}]_{k=1}^K). \tag{7}$$

where bb represents the target proposal with proposal targetness confidence c.

3.4. Loss

In the MLV module, an MLP network is used to obtain a seed-wise targetness score $s^{i,s}$ for each $\varphi^i(x)'$. Therefore, the size of votes expands to $M^i \times (1 + 3 + d)$. We employ a standard binary cross entropy loss L_{cls} for $s^{i,s}$ at each level. Then the loss for multi-level voting is:

$$L_{\text{reg}} = \sum_i \left(\frac{1}{M^i} \sum_j \left\| \Delta p_j^{v,i} - \Delta gt_j \right\| \cdot \mathbb{I}[p_j^{v,i} \text{ on target}]\right). \tag{8}$$

where $\mathbb{I}[p_j^{v,i} \text{ on target}]$ is the indicator of whether $p_j^{v,i}$ is on the surface based on the ground truth. M^i denotes the seed point number at the i-th level, and Δgt_j denotes the ground-truth offset from $p_j^{v,i}$ to the target center. The other loss terms L_{pro}, L_{box} are similar to P2B [31]. L_{pro} is a cross entropy loss for c in Eq. 7. L_{box} is a Huber (smooth-L1) loss for bb in Eq. 7. Finally, we combine all the above losses as the final loss L_{final}:

$$L_{final} = \lambda_1 L_{pro} + \lambda_2 L_{box} + \lambda_3 L_{cls} + L_{reg}. \tag{9}$$

where $\lambda_1, \lambda_2, \lambda_3$ are hyper-parameters to balance the different losses (In our experiments, we empirically set $\lambda_1 = 1.5$, $\lambda_2 = 0.2$, $\lambda_3 = 0.2$).

4. Experiments and Results

MLVSNet is evaluated on the challenging 3D visual tracking benchmanrk of KITTI dataset [13]. We first introduce the KITTI tracking dataset in Sec. 4.1. In Sec. 4.2,

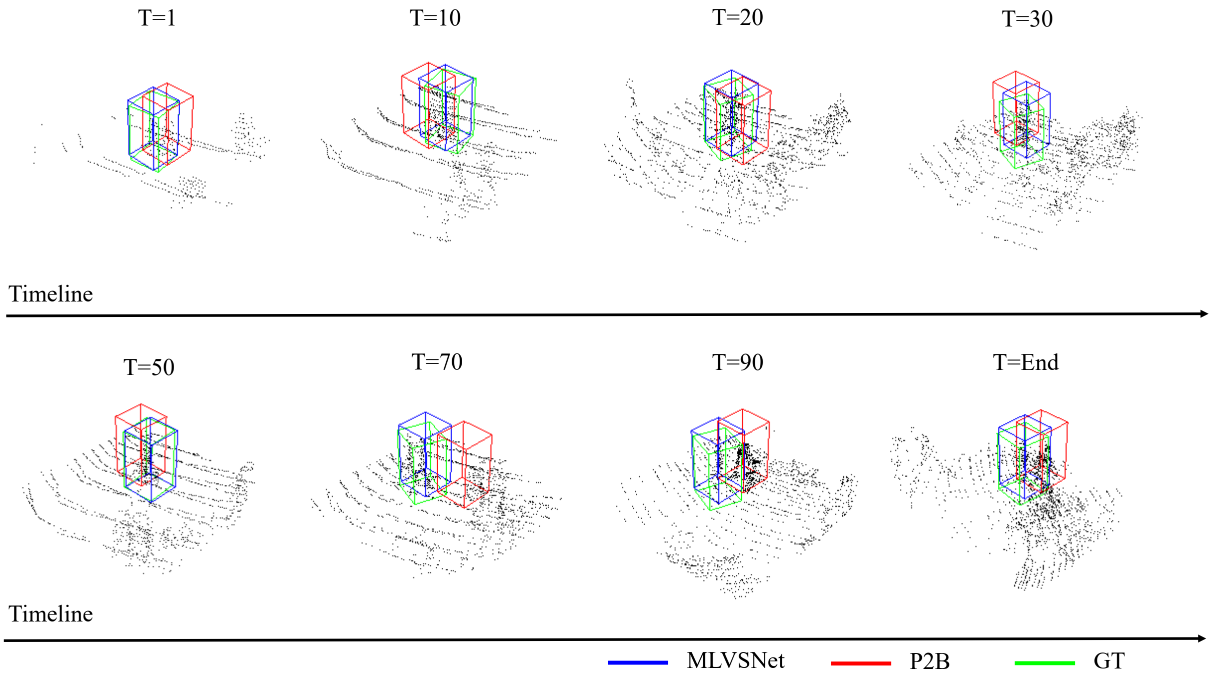


Figure 5. Visual comparison between P2B and MLVSNet in a noisy scene. Compared with P2B, MLVSNet is closer to ground truth even with the interference of background noise.

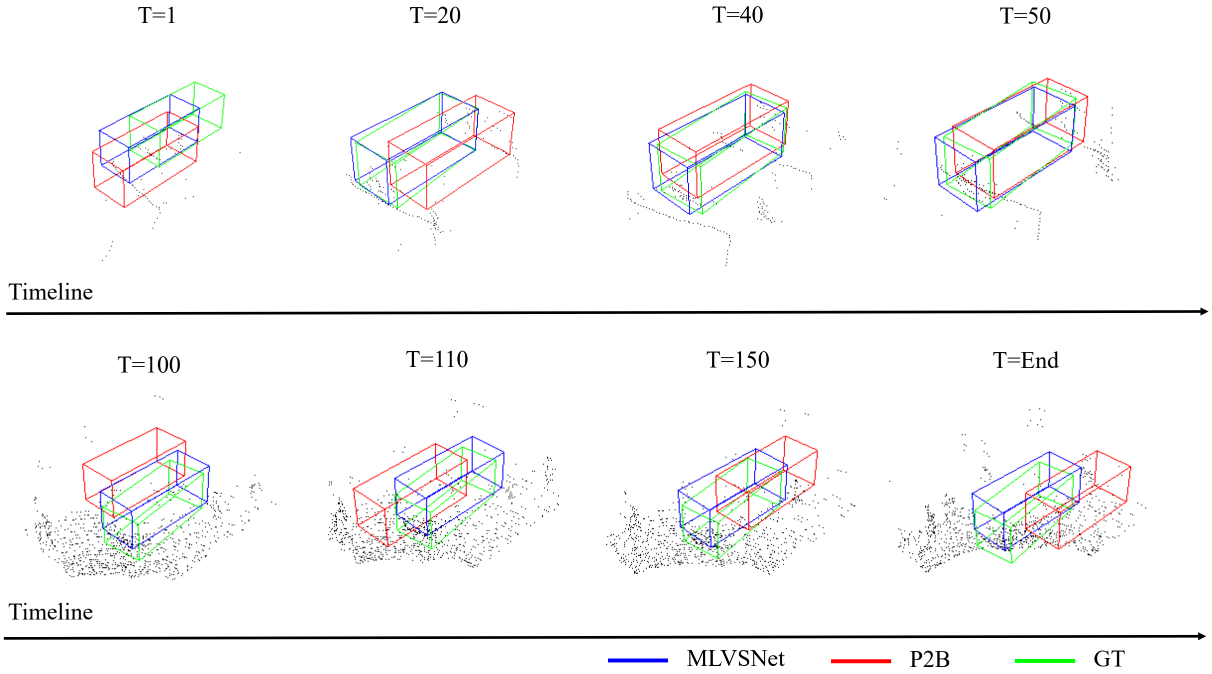


Figure 6. Visual comparison between P2B and MLVSNet in a scene with sparse object point clouds. MLVSNet achieves better performance than P2B.

we describe the implementation details of MLVSNet. In Sec. 4.3, we compare the results of MLVSNet with state-of-the-art 3D visual tracking frameworks. In Sec. 4.4, we conduct extensive ablation studies to analyze MLVSNet.

4.1. Dataset

The original KITTI tracking dataset consists of 21 training scenes and 29 test scenes. We follow the standard setting in [31] and [14] to split the training set as follows:

scenes 0-16 are used for training, 17-18 for validation, and 19-20 for testing. The sparse point clouds of some categories carry little semantic information, which makes it a great challenge for 3D visual tracking. Hence, recent works mainly focused on tracking cars, the category that has the largest quantity and diversity of samples. By contrast, we conduct experiments on all categories (Car, Pedestrian, Van, Cyclist) using MLVSNet, and compare the results with other state-of-the-art methods.

4.2. Implementation Details

Given the point clouds of a target, we first normalized the number of points to $N_I = 512$ by randomly repeating or removing points. We did the same for the point clouds of the search area and set the number of points to $M_I = 1024$. For the backbone network, we follow the architecture of [31], where three set-abstraction (SA) layers are used to sub-sample the target points into groups with sizes [256, 128, 64], and to sub-sample the points of the search area into groups with sizes [512, 256, 128]. The feature channel sizes of the three SA layers for both the target and the search area are [128, 256, 256]. We use features from the second and third layers for multi-level voting, the detailed analysis of this setting can be seen in Sec. 4.4.

Our MLVSNet is implemented in Python 3.6 and PyTorch 1.2. All experiments in this paper are carried out on a PC with an Intel i7-7700K CPU and a GeForce GTX 1080ti GPU. We train the entire network end-to-end with the Adam optimizer [20]. With a batch size of 8, the learning rate is 10^{-3} initially and reduced by a factor of 5 after 10 epochs. During training, we use the same parameters for all types of targets. For other data processing settings, we follow the settings in [31].

4.3. Comparison with the State-of-the-art

Evaluation Metric. To quantitatively compare the tracking performance of different methods, we adopted the One Pass Evaluation (OPE) [39] to compute Success and Precision. Under this rule, "Success" measures the IOU (Intersection Over Union) between the ground-truth bounding box and the predicted bounding box. "Precision" indicates the percentage of frames whose predicted box centers are within the given distance range (0 to 2m) from the ground truth.

Quantitative Comparison. We evaluate our model against the state-of-the-art methods, P2B [31] and SC3D [14]. Table 1 shows the results on the KITTI tracking dataset. It can be seen that the proposed MLVSNet achieves the best mean Success and Precision. Especially for vans, MLVSNet achieves 11.2% and 13% improvements over P2B, the second best, on Success and Precision respectively. For pedestrians, MLVSNet also achieves significant improvements. For cyclists, SC3D achieves better performance than the other two methods. We believe this is probably due to the small number of cyclist samples in the training set. P2B and MLVSNet cannot learn effective target features through insufficient training samples. However, SC3D requires less data to learn the similarity between two regions [31, 14]. Hence, SC3D has achieved an advantage in Cyclist evaluation. Finally, in the mean evaluation score, our method achieves the best performance (compared with P2B, 3.3% higher on Success and 6.6% higher on Precision).

	Method	Car	Pedestrian	Van	Cyclist	Mean
	TN	6424	6088	1248	308	14068
Success	SC3D [14]	41.3	18.2	40.4	**41.5**	31.2
	P2B [31]	**56.2**	28.7	40.8	32.1	42.4
	MLVSNet	56.0	**34.1**	**52.0**	34.3	**45.7**
Precision	SC3D [14]	57.9	37.8	47.0	**70.4**	48.5
	P2B [31]	72.8	49.6	48.4	44.7	60.0
	MLVSNet	**74.0**	**61.1**	**61.4**	44.5	**66.6**

Table 1. Extensive comparisons with state-of-the-art methods. The right five columns exhibit the results in different target types and their mean. TN represents the number of test samples. Our method (MLVSNet) outperforms the state-of-the-art methods by a large margin.

Module	Model size ↓	Training ↓	Testing ↑
P2B+TSFA [31]	5.4MB	21 min	84 fps
P2B+our TGA	**5.3**MB	**16** min	**93** fps
MLVSNet	7.6MB	19 min	70 fps

Table 2. Model size, training time and test speed.

Visual Comparison. We use Mayavi [32] to generate our visual results. Fig. 5 and Fig. 6 show representative examples of MLVSNet tracking results on Pedestrian and Car, respectively. As can be seen in Fig. 5, in street scenes with pedestrians walking, there are a lot of noisy points in the search area, which pose a great challenge for tracking. As can be seen, the tracking results of P2B are affected with large deviations from the ground-truth, while MLVSNet achieves accurate tracking consistently through the frames, demonstrating its robustness to noise. Fig. 6 shows the tracking results of P2B and MLVSNet in a scene with sparse points. In frames from $T = 1$ to $T = 50$, where the points are extremely few, both P2B and MLVSNet perform poorly with the results of MLVSNet slightly closer to the ground-truth. When the number of object points increases (from $T = 100$ to $T = End$), MLVSNet locates targets more accurately, much closer to the ground-truth than P2B.

Speed Comparison. Here, we compare the computational efficiency of the TSFA module in P2B and the proposed TGA module in MLVSNet. We have adopted the same data processing strategy as P2B, so the speed comparison focuses on network efficiency. As shown in Table 2, we evaluate module efficiency in three metrics. The training time is measured as the average time to train the network for one epoch (trained for the car category). The test speed is calculated as the number of samples processed by the model in one second during testing. To evaluate the TSFA module, we used the default settings in P2B. To evaluate the TGA module, we directly replaced the TSFA module in P2B with the TGA module to test the training/testing time for a fair comparison. In all metrics, the efficiencies of the TGA module are better than the TSFA module. It is precisely due to the lightweight of the TGA module that when we combine all the proposed modules (TGA, MLV and VFE) into MLVSNet, the whole model can still achieve 70 FPS at testing. Moreover, our method requires less training time than P2B.

4.4. Ablation Study

In this section, we conduct experiments to analyze the effectiveness of different modules in MLVSNet. P2B is employed as the baseline, and we analyze the effects of using different module settings on network performance. We first replace the TSFA module of P2B with the TGA module while keeping other parts unchanged. As is shown in Table 3, using the TGA module alone gains higher Precision and slightly lower Success, compared with the baseline (2.3% higher on Precision, 0.1% lower on Sucess). Based on the TGA module, we further verify the effectiveness of the MLV module. Initially, the Hough voting is taken from seed points at all layers. However, we find that using seed points from all layers is actually detrimental to the network performance. Table 4 shows the results of using seed points from different combinations of layers. The best results are achieved when aggregating the points from the last two layers (3 and 2) for Hough voting. This confirmed our point that both the number of seed points and the feature descriptive ability of points are important for target detection. Although there are many shallow seed points, the descriptive ability of shallow features is insufficient. When the framework introduces too many shallow seed points for voting, the following sampling would have an imbalanced preference for these shallow points due to their large quantity. Therefore, in implementation of MLVSNet, we only combine the last two layers for the following voting. Table 5 shows the performance of MLVSNet with and without the VFE module for different target types. The performance of using VFE module has obvious improvements on all target types. The proposed VFE module enables different voting clusters to perceive each other, which further improves the tracking performance.

Framework	TGA	MLV	VFE	Metric	
				Sucess	Precision
Baseline				42.4	60.0
MLVSNet	✓			42.3	62.3
MLVSNet	✓	✓		44.4	64.9
MLVSNet	✓	✓	✓	**45.7**	**66.6**

Table 3. Effectiveness of different sub-module of MLVSNet.

Metric	Voting layers	Car	Pedestrian	Van	Cyclist	Mean
Success	3	54.1	29.8	47.1	25.3	42.3
	3+2	**55.2**	**32.4**	**50.9**	**29.6**	**44.4**
	3+2+1	49.8	25.3	33.4	24.2	37.2
Precision	3	70.5	56.3	56.7	33.7	62.3
	3+2	**72.7**	**58.9**	**60.3**	**39.9**	**64.9**
	3+2+1	69.2	44.8	43.6	33.9	55.6

Table 4. Effectiveness of different voting strategies. MLV has the best performance using the last two features for Hough voting.

Metric	Module	Car	Pedestrian	Van	Cyclist	Mean
Success	w/o VFE	55.2	32.4	50.9	29.6	44.4
	w/ VFE	**56.0**	**34.1**	**52.0**	**34.3**	**45.7**
Precision	w/o VFE	72.7	58.9	60.3	39.9	64.9
	w/ VFE	**74.0**	**61.1**	**61.4**	**44.5**	**66.6**

Table 5. Effectiveness of VFE module for all target types and their Mean.

5. Conclusion

In this paper, we present a new network, MLVSNet, for 3D visual tracking on point clouds. Specifically, we propose a lightweight Target-Guided Attention module to embed the target information into the search area. Based on TGA, we further propose a multi-level voting strategy to fuse data and information from different layers for voting, in order to improve the tracking performance on sparse point clouds in outdoor scenes. A feature enhancement module is also proposed to enhance features for tracking by considering relationships between vote clusters. MLVSNet achieves the new state-of-the-art performance and runs at 70 FPS, demonstrating both its effectiveness and efficiency for 3D visual tracking. In the future, we plan to improve the tracking performance in sparse point cloud scenes. One possible solution is to add RGB images as network input. RGB images can provide rich texture and color information, which can increase the difference between the target and the noise points.

Acknowledgment

This work was supported in part by the National Key Research and Development Program of China (2019YFB1707504), National Natural Science Foundation of China under Grant 61772267, and the Natural Science Foundation of Jiangsu Province under Grant BK20190016.

References

[1] Alireza Asvadi, Pedro Girão, Paulo Peixoto, and Urbano Nunes. 3D object tracking using rgb and lidar data. In *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, pages 1255–1260. IEEE, 2016. 2

[2] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *European conference on computer vision*, pages 850–865. Springer, 2016. 2

[3] Adel Bibi, Tianzhu Zhang, and Bernard Ghanem. 3D part-based sparse tracker with automatic synchronization and registration. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1439–1448, 2016. 2

[4] Boyu Chen, Peixia Li, Chong Sun, Dong Wang, Gang Yang, and Huchuan Lu. Multi attention module for visual tracking. *Pattern Recognition*, 87:80–93, 2019. 3

[5] Jintai Chen, Biwen Lei, Qingyu Song, Haochao Ying, Danny Z Chen, and Jian Wu. A hierarchical graph network for 3D object detection on point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 392–401, 2020. 1

[6] Xuesong Chen, Xiyu Yan, Feng Zheng, Yong Jiang, Shu-Tao Xia, Yong Zhao, and Rongrong Ji. One-shot adversarial attacks on visual tracking with dual attention. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10176–10185, 2020. 3

[7] Jongwon Choi, Hyung Jin Chang, Jiyeoup Jeong, Yiannis Demiris, and Jin Young Choi. Visual tracking using attention-modulated disintegration and integration. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4321–4330, 2016. 3

[8] Gioele Ciaparrone, Francisco Luque Sánchez, Siham Tabik, Luigi Troiano, Roberto Tagliaferri, and Francisco Herrera. Deep learning in video multi-object tracking: A survey. *Neurocomputing*, 381:61–88, 2020. 1

[9] Xingping Dong and Jianbing Shen. Triplet loss in siamese network for object tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 459–474, 2018. 2

[10] Fei Du, Peng Liu, Wei Zhao, and Xianglong Tang. Correlation-guided attention for corner detection based visual tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6836–6845, 2020. 3

[11] Heng Fan and Haibin Ling. Siamese cascaded region proposal networks for real-time visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7952–7961, 2019. 1

[12] Peng Gao, Ruyue Yuan, Fei Wang, Liyi Xiao, Hamido Fujita, and Yan Zhang. Siamese attentional keypoint network for high performance visual tracking. *Knowledge-based systems*, 193:105448, 2020. 2

[13] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012. 5

[14] Silvio Giancola, Jesus Zarzar, and Bernard Ghanem. Leveraging shape completion for 3D siamese tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1359–1368, 2019. 2, 6, 7

[15] Qing Guo, Wei Feng, Ce Zhou, Rui Huang, Liang Wan, and Song Wang. Learning dynamic siamese network for visual object tracking. In *Proceedings of the IEEE international conference on computer vision*, pages 1763–1771, 2017. 2

[16] Anfeng He, Chong Luo, Xinmei Tian, and Wenjun Zeng. A twofold siamese network for real-time object tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4834–4843, 2018. 2

[17] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018. 3

[18] Ugur Kart, Joni-Kristian Kamarainen, and Jiri Matas. How to make an rgbd tracker? In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 0–0, 2018. 2

[19] Ugur Kart, Alan Lukezic, Matej Kristan, Joni-Kristian Kamarainen, and Jiri Matas. Object tracking by reconstruction with view-specific discriminative correlation filters. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1339–1348, 2019. 2

[20] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7

[21] Bo Li, Wei Wu, Qiang Wang, Fangyi Zhang, Junliang Xing, and Junjie Yan. Siamrpn++: Evolution of siamese visual tracking with very deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4282–4291, 2019. 2

[22] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8971–8980, 2018. 2

[23] Peixia Li, Dong Wang, Lijun Wang, and Huchuan Lu. Deep visual tracking: Review and experimental comparison. *Pattern Recognition*, 76:323–338, 2018. 1

[24] Ye Liu, Xiao-Yuan Jing, Jianhui Nie, Hao Gao, Jun Liu, and Guo-Ping Jiang. 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, 2018. 2

[25] Zheng Pan, Shuai Liu, Arun Kumar Sangaiah, and Khan Muhammad. Visual attention feature (VAF): a novel strategy for visual tracking based on cloud platform in intelligent surveillance systems. *Journal of Parallel and Distributed Computing*, 120:182–194, 2018. 3

[26] Jongchan Park, Sanghyun Woo, Joon-Young Lee, and In So Kweon. Bam: Bottleneck attention module. *arXiv preprint arXiv:1807.06514*, 2018. 2

[27] Alessandro Pieropan, Niklas Bergström, Masatoshi Ishikawa, and Hedvig Kjellström. Robust 3D tracking of unknown objects. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2410–2417. IEEE, 2015. 2

[28] Charles R Qi, Xinlei Chen, Or Litany, and Leonidas J Guibas. Imvotenet: Boosting 3D object detection in point clouds with image votes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4404–4413, 2020. 2

[29] Charles R Qi, Or Litany, Kaiming He, and Leonidas J Guibas. Deep hough voting for 3D object detection in point clouds. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9277–9286, 2019. 1, 4

[30] Charles Ruizhongtai Qi, Li Yi, Hao Su, and Leonidas J Guibas. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. In *Advances in neural information processing systems*, pages 5099–5108, 2017. 2, 4

[31] Haozhe Qi, Chen Feng, Zhiguo Cao, Feng Zhao, and Yang Xiao. P2b: Point-to-box network for 3D object tracking in point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6329–6338, 2020. 2, 3, 4, 5, 6, 7

[32] Prabhu Ramachandran and Gaël Varoquaux. Mayavi: 3D visualization of scientific data. *Computing in Science & Engineering*, 13(2):40–51, 2011. 7

[33] Xiaoke Shen and Ioannis Stamos. Frustum voxnet for 3D object detection from rgb-d or depth images. In *The IEEE Winter Conference on Applications of Computer Vision*, pages 1698–1706, 2020. 1

[34] Hanyu Shi, Guosheng Lin, Hao Wang, Tzu-Yi Hung, and Zhenhua Wang. Spsequencenet: Semantic segmentation network on 4D point clouds. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4574–4583, 2020. 4

[35] Paul Voigtlaender, Jonathon Luiten, Philip HS Torr, and Bastian Leibe. Siam r-cnn: Visual tracking by re-detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6578–6588, 2020. 1

[36] Fei Wang, Mengqing Jiang, Chen Qian, Shuo Yang, Cheng Li, Honggang Zhang, Xiaogang Wang, and Xiaoou Tang. Residual attention network for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2017. 3

[37] Qiang Wang, Zhu Teng, Junliang Xing, Jin Gao, Weiming Hu, and Stephen Maybank. Learning attentions: residual attentional siamese network for high performance online visual tracking. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4854–4863, 2018. 2

[38] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018. 2, 3, 5

[39] Yi Wu, Jongwoo Lim, and Ming-Hsuan Yang. Online object tracking: A benchmark. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2411–2418, 2013. 7

[40] Qian Xie, Yu-Kun Lai, Jing Wu, Zhoutao Wang, Yiming Zhang, Kai Xu, and Jun Wang. Mlcvnet: Multi-level context votenet for 3D object detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10447–10456, 2020. 2

[41] Qian Xie, Yu-Kun Lai, Jing Wu, Zhoutao Wang, Yiming Zhang, Kai Xu, and Jun Wang. Vote-based 3D object detection with context modeling and SOB-3DNMS. *International Journal of Computer Vision*, 129(6):1857–1874, 2021. 2

[42] Zetong Yang, Yanan Sun, Shu Liu, and Jiaya Jia. 3dssd: Point-based 3D single stage object detector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11040–11048, 2020. 1

[43] Alper Yilmaz, Omar Javed, and Mubarak Shah. Object tracking: A survey. *Acm computing surveys (CSUR)*, 38(4):13–es, 2006. 1

[44] Yuechen Yu, Yilei Xiong, Weilin Huang, and Matthew R Scott. Deformable siamese attention networks for visual object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6728–6737, 2020. 3

[45] Jesus Zarzar, Silvio Giancola, and Bernard Ghanem. Efficient tracking proposals using 3D siamese networks on lidar. *arXiv preprint arXiv:1903.10168*, 2019. 3

[46] Jianming Zhang, Xiaokang Jin, Juan Sun, Jin Wang, and Arun Kumar Sangaiah. Spatial and semantic convolutional features for robust visual object tracking. *Multimedia Tools and Applications*, 79(21):15095–15115, 2020. 1

[47] Zhipeng Zhang and Houwen Peng. Deeper and wider siamese networks for real-time visual tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4591–4600, 2019. 1, 2

[48] Zheng Zhu, Qiang Wang, Bo Li, Wei Wu, Junjie Yan, and Weiming Hu. Distractor-aware siamese networks for visual object tracking. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 101–117, 2018. 2