P2B: Point-to-Box Network for 3D Object Tracking in Point Clouds

Haozhe Qi, Chen Feng, Zhiguo Cao,∗ Feng Zhao, and Yang Xiao

National Key Laboratory of Science and Technology on Multi-Spectral Information Processing, School of Artificial Intelligence and Automation, Huazhong University of Science and Technology, Wuhan 430074, China
qihaozhe, chen_feng, zgcao@hust.edu.cn, fzhao@alumni.hust.edu.cn, Yang_Xiao@hust.edu.cn

Abstract
Towards 3D object tracking in point clouds, a novel point-to-box network termed P2B is proposed in an end-to-end learning manner. Our main idea is to first localize potential target centers in 3D search area embedded with target information. Then point-driven 3D target proposal and verification are executed jointly. In this way, the time-consuming 3D exhaustive search can be avoided. Specifically, we first sample seeds from the point clouds in template and search area respectively. Then, we execute permutation-invariant feature augmentation to embed target clues from template into search area seeds and represent them with target-specific features. Consequently, the augmented search area seeds regress the potential target centers via Hough voting. The centers are further strengthened with seed-wise targetness scores. Finally, each center clusters its neighbors to leverage the ensemble power for joint 3D target proposal and verification. We apply PointNet++ as our backbone and experiments on KITTI tracking dataset demonstrate P2B’s superiority (∼10%’s improvement over state-of-the-art). Note that P2B can run with 40FPS on a single NVIDIA 1080Ti GPU. Our code and model are available at https://github.com/HaozheQi/P2B.

1. Introduction
3D object tracking in point clouds is essential for applications in autonomous driving and robotics vision [25, 26, 7]. However, point clouds’ sparsity and disorder imposes great challenges on this task, and leads to the fact that, well-established 2D object tracking approaches (e.g., Siamese network [3]) cannot be directly applied. Most existing 3D object tracking methods [1, 4, 24, 16, 15] inherit 2D’s experience and rely heavily on RGB-D information. But they may fail when RGB visual information is degraded with luminational change or even inaccessible. We hence focus on 3D object tracking using only point clouds. The first pioneer effort on this topic appears in [11]. It mainly executes 3D template matching using Kalman filtering [12] to generate bunches of 3D target proposals. Meanwhile, it uses shape completion to regularize feature learning on point set. Nevertheless, it tends to suffer from four main defects: 1) its tracking network cannot be end-to-end trained; 2) 3D search with Kalman filtering consumes much time; 3) each target proposal is represented with only one-dimensional global feature, which may lose fine local geometric information; 4) shape completion network brings strong class prior which weakens generality.

Towards the above concerns, we propose a novel point-to-box network termed P2B for 3D object tracking which can be end-to-end trained. Differing from the intuitive 3D search with box in [11], we turn to addressing 3D object tracking by first localizing potential target centers and then executing point-driven target proposal and verification jointly. Our intuition lies in two folders. First, the point-wise tracking paradigm may help better exploit 3D local geometric information to characterize target in point clouds.
Secondly, formulating 3D object tracking task in an end-to-end manner is of stronger ability to fit target’s 3D appearance variation during tracking.

We exemplify how P2B works in Fig. 1. We first feed template and search area into backbone respectively and obtain their seeds. The search area seeds will consequently predict potential target centers for joint target proposal and verification. Then the search area seeds are augmented with target-specific features, yielding three main components: 1) their 3D position coordinates to retain spatial geometric information, 2) their point-wise similarity with template seeds to mine resembling patterns and reveal the local tracking clue, and 3) encoded global feature of target from template. This augmentation is invariant to seeds’ permutation and yields consistent target-specific features. After that, the augmented seeds are projected to the potential target centers via Hough voting [28]. Meanwhile, each seed is assessed with its targetness to regularize earlier feature learning; the result targetness score further strengthens its predicted target center’s representation. Finally, each potential target center clusters the neighbors to leverage the ensemble power for joint target proposal and verification.

Experiments on KITTI tracking dataset [10] demonstrate that, P2B significantly outperforms the state-of-the-art method [11] by large a margin (~10% on both Success and Precision). Note that P2B can run with about 40FPS on a single NVIDIA 1080Ti GPU.

Overall, the main contributions of this paper include

- P2B: a novel point-to-box network for 3D object tracking in point clouds, which can be end-to-end trained;
- Target-specific feature augmentation to include global and local 3D visual clues for 3D object tracking;
- Integration of 3D target proposal and verification.

2. Related Works

We briefly introduce the works most related to our P2B: 3D object tracking, 2D Siamese tracking, deep learning on point set, target proposal and Hough voting.

3D object tracking. To the best of our knowledge, 3D object tracking using only point clouds has seldom been studied before the recent pioneer attempt [11]. Earlier related tracking methods [24, 16, 15, 27, 1, 4] generally resort to RGB-D information. Though with the paid efforts from different theoretical aspects, they may suffer from two main defects: 1) they rely on RGB visual clue and may fail if it is degraded or even inaccessible. This limits some real applications; 2) they have no networks designed for 3D tracking, which may limit the representative power. Besides, some of them [24, 16, 15] focus on generating 2D boxes. The above concerns are addressed in [11]. Leveraging deep learning on point set and 3D target proposal, it achieves the state-of-the-art result on 3D object tracking using only point clouds. However, it still suffers from some drawbacks as in Sec. 1, which motivates our research.

2D Siamese tracking. Numerous state-of-the-art 2D tracking methods [33, 3, 34, 13, 42, 35, 20, 8, 40, 36, 21] are built upon Siamese network. Generally, Siamese network has two branches for template and search area with shared weights to measure their similarity in an implicitly embedded space. Recently, [21] unites region proposal network and Siamese network to boost performance. Hence, time-consuming multi-scale search and online fine-tuning are both avoided. Afterwards, many efforts [42, 20, 40, 36, 8] follow this paradigm. However, the above methods are all driven by 2D CNN which is inapplicable to point clouds. We hence aim to extend the Siamese tracking paradigm to 3D object tracking with effective 3D target proposal.

Deep learning on point set. Recently, deep learning on point set draws increasing research interests [5, 30]. To address point clouds’ disorder, sparsity and rotation variance, the paid efforts have facilitated the research in 3D object recognition [18, 23], 3D object detection [28, 29, 32, 39], 3D pose estimation [22, 9, 6], and 3D object tracking [11]. However, the 3D tracking network in [11] cannot ex-
Figure 3. The idea of permutation-invariance. To represent \( r_j \), we first compute point-wise similarity \( \text{Sim}_{i,j} \), between \( r_j \) and all template seeds \( Q = \{q_i\}_{i=1}^3 \). However, \( \text{Sim}_{i,j} \) keeps changing due to \( Q \)'s disorder (\( Q \)'s order can change irregularly). This motivates our feature augmentation for consistent (i.e., permutation-invariant) \( f_{\text{Sim}}^i \), “1, 2, 3” denote dimensions in \( \text{Sim}_{i,j} \), and \( f_{\text{Sim}}^j \).

**3.2. Target-specific feature augmentation**

Here we aim to merge template’s target information into search area seed to include both global target clue and local tracking clue. We first feed template and search area respectively into feature backbone and obtain their seeds. Then the template seeds help augment the search area seeds with target-specific features. After that, these augmented search area seeds are projected to potential target centers via Hough voting. Seed-wise targetness scores are also calculated to regularize feature learning and strengthen the discriminative power of these potential target centers. Then each potential target center clusters its neighbors for 3D target proposal. Proposal with the maximal proposal-wise targetness score is verified as the final result. We will detail them as follows. Main symbols within P2B are defined in Table 1. For easy comprehension, we also sketch the detailed technical flow in Algorithm 1.

**3.3. P2B: A Novel Network on Point Set for 3D Object Tracking**

**3.1. Overview**

In 3D object tracking, we focus on localizing the target (defined by template) in search area frame by frame. We aim to embed template’s target clue into search area to predict potential target centers, and execute joint target proposal and verification in an end-to-end manner. P2B has two main parts (Fig. 2): 1) target-specific feature augmentation, and 2) 3D target proposal and verification. We first feed template and search area respectively into backbone and obtain their seeds. Then the template seeds help augment the search area seeds with target-specific features. After that, these augmented search area seeds are projected to potential target centers via Hough voting. Seed-wise targetness scores are also calculated to regularize feature learning and strengthen the discriminative power of these potential target centers. Then each potential target center clusters its neighbors for 3D target proposal. Proposal with the maximal proposal-wise targetness score is verified as the final result. We will detail them as follows. Main symbols within P2B are defined in Table 1. For easy comprehension, we also sketch the detailed technical flow in Algorithm 1.
**Permutation-invariant target-specific feature augmentation.** To embed $Q$’s target information into $R$, a natural idea is to compute point-wise similarity $Sim$ (of size $M_2 \times M_1$) between $Q$ and $R$, e.g., using cosine distance:

$$Sim_{j,i} = \frac{f_{q_i}^T \cdot f_{r_j}}{\|f_{q_i}\|_2 \cdot \|f_{r_j}\|_2}, \forall q_i \in Q, r_j \in R.$$ (1)

Note that $Sim_{j,:}$ (row $j$ in $Sim$) denotes similarity between $r_j$ and all seeds in $Q$. We may first consider $Sim_{j,:}$ as $r_j$’s target-specific feature. However, as in Fig. 3, $Sim_{j,:}$ keeps unstable due to $Q$’s disorder. This contradicts our need for a consistent feature, *i.e.*, a feature invariant to $Q$’s inside permutation. We accordingly apply symmetric functions (specifically, Maxpool) to ensure permutation-invariance. As in Fig. 4, we first augment each $Sim_{j,:}$ (local tracking clue) with $Q$’s spatial coordinates and features (global target clue), yielding a tensor of size $M_1 \times (1+3+d_1)$. Then we feed the tensor into network $\Phi$ (MLP-Maxpool-MLP) and obtain $r_j$’s target-specific feature, $f_{r_j}^s \in \mathbb{R}^{d_2}$. $r_j$ is finally represented with $[x_{r_j}^i; f_{r_j}^s] \in \mathbb{R}^{3+d_2}$ (where $x_{r_j}^i = x_{r_j} + \Delta x_{r_j}$ and $f_{r_j}^c = f_{r_j}^i + \Delta f_{r_j}^c$).

The loss for $\Delta x_j$ is termed as

$$L_{\text{reg}} = \frac{1}{M_{ts}} \sum_j \|\Delta x_j - \Delta gt_j\| \cdot [r_j \text{ on target}].$$ (2)

Here, $\Delta gt_j$ denotes the ground-truth offset from $r_j$ to the target center; $[\cdot]$ indicates that we only train those seeds located on the surface of ground-truth target; $M_{ts}$ denotes the number of trained seeds.

**Clustering and Target proposal.** For each $c_j$, we use ball query [30] to generate cluster $T_j^s$ with radius $R$: $T_j^s = \{c_k \mid \|c_k - c_j\|_2 < R\}$. Since neighboring clusters may capture similar region-level context, we sample a subset of size $K$ in all potential target centers as cluster centroids for efficiency. In Sec. 4.3.3, P2B turns robust to a wide range of $K$s. Finally we feed each $T_j^s$ into $\Theta$ (MLP-Maxpool-MLP) and obtain target proposal $p_j^s$ with proposal-wise targetness score $s_j^p$ (totally $K$ proposals are generated):

$$\{p_j^s, s_j^p\} = \Theta(T_j^s).$$ (3)

$p_j^s$ has 4 parameters: offsets for 3D position and rotation in X-Y plane. We will detail how to learn $\Theta$ in Sec. 3.5.

### 3.3. Target proposal based on potential target centers

Embedded with target clue, each $r_j$ can directly predict one target proposal. But our intuition is that, individual seed can only capture limited local clue, which may not suffice the final prediction. We follow the idea within VoteNet [28] to 1) regress the search area seeds into potential target centers via Hough voting, and 2) cluster neighboring centers to leverage the ensemble power and obtain target proposals.

**Potential target center generation.** Each seed $r_j$ with feature $f_{r_j}^i$ can roughly predict a potential target center $c_j$ via Hough voting. Following VoteNet [28], the voting module applies MLP to predict the coordinate offset $\Delta x_j$ between $r_j$ and ground-truth target center and the residual $\Delta f_{r_j}^c$ for $f_{r_j}^c$. Hence, $c_j$ is represented with $[x_{c_j}; f_{c_j}] \in \mathbb{R}^{3+d_2}$ where $x_{c_j} = x_{r_j} + \Delta x_{r_j}$ and $f_{c_j} = f_{r_j}^i + \Delta f_{r_j}^c$. We may first consider $\Delta x_j$’s inside $T_j^s$ and all seeds in $Q$ as candidates for $c_j$. We accordingly apply symmetric functions (specifically, Maxpool) to ensure permutation-invariance.

**Improved target proposal with seed-wise targetness score.** We consider each seed with target-specific feature can be directly assessed with its targetness to 1) regularize earlier feature learning and 2) strengthen the representation of its predicting potential target center. Therefore, we can obtain target proposals with higher quality.

**Seed-wise targetness score $s^p$.** We learn a MLP to generate $s_j^p$ for each $r_j$. Those search area seeds located on the surface of ground-truth target are regarded as positives, and the extra as negatives. We use a standard binary cross entropy loss $L_{\text{cla}}$ for $s^p$. Since $s_j^p$ tightly relates to $f_{r_j}^i$, $L_{\text{cla}}$ can explicitly constrain the point feature learning and consequent target-specific feature augmentation.

**Improved target proposal.** Inheriting more discriminative power from $s_j^p$, we update $c_j$’s representation with...
We consider that, $s^a$ can implicitly help pick out representative potential target centers to benefit final target proposal.

### 3.5. Final target verification

With $K$ proposals generated from above (refer to $\Theta$ in Equation (3)), proposal with the highest proposal-wise targetness score is verified as the final tracking result.

We follow VoteNet [28] to learn $\Theta$. Specifically, we consider proposals whose centers near the target center (within 0.3 meters) as positives and those faraway (by more than 0.6 meters) as negatives. Other proposals are left unpunalized. We use a standard binary cross entropy loss $L_{prop}$ for $s_j^p$. As for $p_j^p$, only the positives’ box parameters are supervised via Huber (smooth-L1 [31]) loss $L_{box}$. We aggregate all the mentioned losses as our final loss $L$:

$$L = L_{reg} + \gamma_1 L_{cla} + \gamma_2 L_{prop} + \gamma_3 L_{box}. \quad (4)$$

Here $\gamma_1 = 0.2$, $\gamma_2 = 1.5$ and $\gamma_3 = 0.2$ are used to normalize all the component losses to be of the same scale.

### 4. Experiments

We applied KITTI tracking dataset [10] (with point clouds scanned using lidar) as benchmark. We followed settings in [11] (shortened as SC3D by us for simplicity) in data split, tracklet generation¹ and evaluation metric for fair comparisons. Since cars in KITTI appear in largest quantity and diversity, we mainly focused on car tracking and perform ablation study on it as in SC3D. We also did extensive experiments with other three target types (Pedestrian, Van, Cyclist) for better comparisons.

#### 4.1. Experimental setting

##### 4.1.1 Dataset

Since ground truth for test set in KITTI is inaccessible offline, we used its training set to train and test our P2B. This tailored dataset had 21 outdoor scenes and 8 types of targets. We generated tracklets for target instances within all videos and split the dataset as follows: scenes 0-16 for training, 17-18 for validation, and 19-20 for testing.

**Point cloud’s sparsity.** Though each frame reports an average of 120k points, we suppose the points on target might be quite sparse with general occlusion and lidar’s defect on distant objects. To validate our idea, we counted the number of points on KITTI’s cars in Fig. 5. We can observe that about 34% cars held fewer than 50 points. The situation may be worse on smaller-size pedestrians and cyclists. This sparsity imposes great challenge onto point cloud based 3D object tracking.

¹Frames containing the same target instance, e.g., a car, are concatenated by time order to form a tracklet.

### 4.1.2 Evaluation metric

We used One Pass Evaluation (OPE) [38] to measure Success and Precision of different methods. “Success” is defined as IOU between predicted box and ground-truth (GT) box. “Precision” is defined as AUC for errors (distance between two boxes’ centers) from 0 to 2m.

#### 4.1.3 Implementation details

**Template and search area.** For template², we collected and normalized its points to $N_1 = 512$ ones with randomly abandoning or duplicating. For search area, we similarly collected and normalized the points to $N_2 = 1024$ ones. The ways to generate template and search area differ in training and testing as detailed below.

**Network architecture.** We adopted PointNet++ [30] as our backbone. We tailored it to contain three set-abstraction (SA) layers, with receptive radius of 0.3, 0.5, 0.7 meters, and 3 times of half-size down-sampling. This yielded $M_1 = 64 (= N_1/2^3)$ template seeds and $M_2 = 128 (= N_2/2^3)$ search area seeds. We applied random sampling, and removed up-sampling layers in PointNet++ due to points’ sparsity. The output feature was of $d_1 = 256$ dimensions.

Throughout our method, all used MLPs had three layers. The size of these layers was 256 (hence $d_2 = 256$) except that of the last layers ($size_{ly}$) in following MLPs:

- For MLP to predict $s^p$, $size_{ly} = 1$.
- For $\Theta$ to predict $s^p$ and $p^t$, $size_{ly} = 5$.

**Clustering.** $K = 64$ randomly sampled potential target centers clustered the neighbors within $R = 0.3$ meters.

**Training.** 1) Data Augmentation: we applied random offset on previous GT and fused point clouds within the result box and the first GT for more template samples; we enlarged the current GT by 2 meters to include background (negative seeds), applied similar random offset and collected inside point cloud for more search area samples. 2) We trained P2B from scratch with the augmented samples.

²Template and search area are in forms of point clouds. GT and result are in forms of 3D boxes.
We applied Adam optimizer [17]. Learning rate was initially 0.001 and decreased by 5 times after 10 epochs. Batch size was 32. In practice, we observed P2B converged to a satisfying result after about 40 epochs.

**Testing.** We used the trained P2B to infer 3D bounding boxes within tracklets frame by frame. For the current frame, template initially adopted the first GT’s point cloud and then fusion of the first GT’s and previous result’s point clouds. We enlarged previous result by 2 meters in current frame and collected inside point cloud to obtain search area.

### 4.2. Comprehensive comparisons

We only compared our P2B with SC3D [11], the first and only work on point cloud based 3D object tracking. We reported results for 3D car tracking in Table 2.

We generated search area centered on previous result, previous GT or current GT. Using previous result as the search center meets the requirement of real scenarios, while using previous GT helps approximately assess short-term tracking performance. For the two situations, SC3D applies Kalman filtering to generate proposals. Using current GT is unreasonable, but is considered in SC3D to approximate exhaustive search and assess SC3D’s discriminative power. Specifically, SC3D conducts grid search around target center to include GT box in generated proposals. However, P2B clusters potential target centers to generate proposals without explicit dependence on GT box. *I.e.*, P2B may adapt to various scenarios while SC3D could degrade when the GT boxes are removed as demonstrated in Table 2. Comprehensive, P2B outperformed SC3D by a large margin. All later experiments adopted the more realistic setting of using previous result (“Testing” in Sec. 4.1.3).

**Extensive comparisons.** We further compared P2B with SC3D on Pedestrian, Van, and Cyclist (Table 3). P2B outperformed SC3D by $\sim$10% on average. P2B’s advantage turned significant on data-rich Car and Pedestrian. But P2B degraded when training data decreased as was the case for Van and Cyclist. We conjecture that P2B may rely on more data to learn better networks especially when regressing potential target centers. Comparatively, SC3D needs relatively less data to suffice similarity measuring between two regions. To validate this, we used the model trained on data-rich Car to test Van, with the belief that car resembles van and contains potentially transferable information. As expected, the Success/Precision result of P2B showed an improved 49.9/59.9 (original: 40.8/47.0), while SC3D reported a declined 37.2/45.9 (original: 40.4/47.0).

### 4.3. Ablation study

#### 4.3.1 Ways for target-specific feature augmentation

Besides our default setting in P2B (Sec. 3.2), there are another four possible ways for feature augmentation: removing (the duplication of) template features, removing the similarity map, using search area feature A and B (Fig. 6).

We compared the five settings in Table 4. Here removing template features or similarity map degraded by about 1% or 3%, which validates the contributions of these two parts in our default setting. Search area feature A and B did not improve or even harm the performance. Note that we already combined template features in both conditions. This may reveal that search area features only capture spatial context rather than target clue, and hence turns useless for target-specific feature augmentation. In comparison, our default setting brings with richer target clue from template seeds to yield a more “directed” proposal generation.
Figure 7. Illustration of seed-wise targetness scores and potential target centers. Green lines show projection from seeds (colored points in the first row) to potential target centers (colored points in the second row). We marked those informative points, i.e., with higher targetness scores, in red and opposite in yellow. Paired seed and potential center were marked in the same color to show correlation.

Table 5. Effectiveness of seed-wise targetness.

<table>
<thead>
<tr>
<th>Ways for using $s^t$</th>
<th>Success</th>
<th>Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Our default setting</td>
<td>56.2</td>
<td>72.8</td>
</tr>
<tr>
<td>Without concatenation</td>
<td>55.1</td>
<td>70.8</td>
</tr>
<tr>
<td>Without the whole branch of $s^t$</td>
<td>52.6</td>
<td>67.4</td>
</tr>
</tbody>
</table>

Table 6. Different ways for template generation. “First & Previous” denotes “The first GT and Previous result”.

within the first GT and previous result to update template for efficiency. Here we reported results with four settings for template generation: the first GT, the previous result, the fusion of the first GT and previous result, and all previous results. Results in Table 6 show P2B’s consistent advantage over SC3D in all settings, even in “All previous shapes” where P2B reported degraded result. We attribute the degradation to that 1) we did not include shape completion [11] and 2) we did not train P2B with all previous results while SC3D considered both.

4.4. Qualitative analysis

4.4.1 Advantageous cases

We first exemplified our target-specific feature’s discriminative power in Fig. 7. The first row visualizes seeds’ targetness scores to demonstrate their possibility of belonging to the target (Car). We can observe that P2B had learnt to discriminate the target seeds from the background ones. The second row visualizes how P2B projects seeds to potential target centers. We can observe that the potential centers with more target information gathered tightly around GT target center, which further validates our discriminative target-specific features. Besides, P2B can address the occlusion because it can generate groups of informative potential target centers for final prediction.

We then visualize P2B’s advantage over SC3D to address point cloud’s sparsity in Fig. 9. We can observe that in the sparse scenarios where SC3D tracked off course or even failed, our predicted box held tight to the target center.
4.4.2 Failure cases

Here we searched for tracklets where P2B failed and found that most failure cases arose when initial template in the first frame was too sparse and hence yielded little target information. As exemplified in Fig. 10, when P2B faced such case and tracked off course with cluttered background, points from the initial template cannot modify current erroneous predictions and re-obtain an informative template. This failure may also reveal that P2B inherits target information from template instead of search area.

We believe that when fed with more points containing potentially rich target information, P2B could generate proposals with higher quality to yield better results. Our intuition is validated in Fig. 11.

4.5. Running speed

Here we averaged the running time of all test frames for car to measure P2B’s speed. P2B achieved 45.5 FPS, including 7.0 ms for processing point cloud, 14.3 ms for network forward propagation and 0.9 ms for post-processing, on a single NVIDIA 1080Ti GPU. SC3D in default setting ran with 1.8 FPS on the same platform.

5. Conclusions

In this work we propose a novel point-to-box (P2B) network for 3D object tracking. We focus on embedding the target information within template into search space and formulate an end-to-end method for point-driven target proposal and verification jointly. P2B operates on sampled seeds instead of 3D boxes to reduce search space by a large margin. Experiments justify our proposition’s superiority.

The experiments also reveal that P2B needs more data to obtain satisfying result. Hence, we could expect a less data-dependent P2B while we could also collect more data to handle the issue under this big-data era. Besides, we could seek better ways for feature augmentation in search area and test our method on more challenging scenarios.

Acknowledgements This work is jointly supported by the National Natural Science Foundation of China (Grant No. U1913602, 61876211 and 61502187), Equipment Pre-research Field Fund of China (Grant No. 61403120405), National Key Laboratory Open Fund of China (Grant No. 6142113180211), and the Fundamental Research Funds for the Central Universities (Grant No. 2019kfyXKJC024).
References


[29] Charles R Qi, Wei Liu, Chenzia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-


