Towards Generic 3D Tracking in RGBD Videos: Benchmark and Baseline

Jinyu Yang[1,2], Zhongqun Zhang[2], Zhe Li[1], Hyung Jin Chang[2], Aleš Leonardis[2], and Feng Zheng[1]⋆

[1] Department of Computer Science and Engineering,
Southern University of Science and Technology, Shenzhen, China
[2] University of Birmingham, Birmingham, United Kingdom

Abstract. Tracking in 3D scenes is gaining momentum because of its numerous applications in robotics, autonomous driving, and scene understanding. Currently, 3D tracking is limited to specific model-based approaches involving point clouds, which impedes 3D trackers from applying in natural 3D scenes. RGBD sensors provide a more reasonable and acceptable solution for 3D object tracking due to their readily available synchronised color and depth information. Thus, in this paper, we investigate a novel problem: is it possible to track a generic (class-agnostic) 3D object in RGBD videos and predict 3D bounding boxes of the object of interest? To inspire research on this topic, we newly construct a standard benchmark for generic 3D object tracking, *'Track-it-in-3D'*, which contains 300 RGBD video sequences with dense 3D annotations and corresponding evaluation protocols. Furthermore, we propose an effective tracking baseline to estimate 3D bounding boxes for arbitrary objects in RGBD videos, by fusing appearance and spatial information effectively. Resources are available on `https://github.com/yjybuaa/Track-it-in-3D`.

Keywords: Object tracking, 3D object tracking, RGBD data

1 Introduction

Object tracking is to distinguish an arbitrary object from a video, given only the object location in the first frame. 3D object tracking, which can estimate not only the location but also the 3D size of objects, has a broader spectrum of practical applications involving augmented reality [27], autonomous driving [20], scene understanding [32] and robotic manipulation [7,21].

However, current state-of-the-art 3D trackers are mostly point cloud-based and highly rely on geometric information to estimate the shape of objects. In fact, LiDAR sensors are quite expensive, and the sparsity and disorder of the point cloud impose great challenges on identifying target objects from backgrounds. Whilst, compared with point clouds, the ignored color cues are more

⋆ Corresponding author.

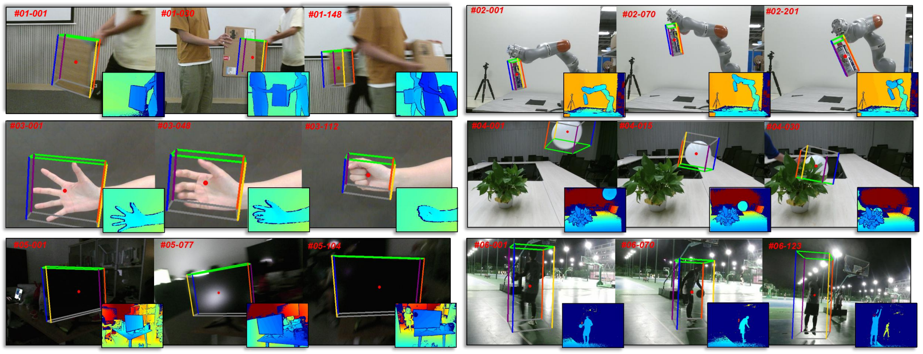

Fig. 1. Examples of RGBD videos in our benchmark dataset. Each video is annotated with the object's per-frame 3D bounding box. Video sequences are captured towards 3D tracking challenges, *e.g.*, (1) similar objects and occlusion; (2) small-sized object; (3) deformation; (4) symmetric object and partial occlusion; (5) dark scene and camera motion; (6) outdoor scenario.

informative for computing appearance features which are widely used to distinguish the target object from backgrounds. In addition, similar to LiDAR, depth information captured by low-cost sensors such as Kinect can also provide geometric information to estimate the shape of targets for most natural tracking scenarios. Moreover, it is easy to get the synchronised color channels from such cameras. Even for modelling target appearance, the depth information can be used to resolve tracking failures in cases of, *e.g.,* distractors or rotation [3,12], due to its insensitivity to the variations in color, illumination, rotation angle, and scale. Therefore, a RGB+D fusion framework is a more reasonable and acceptable solution for 3D object tracking. On the one hand, appearance information in RGB channels and geometry information from the depth channel are two complementary data sources. On the other hand, the 3D coordinate of the object, with the spatial information given by depth information in 3D scenes, is more practical on real-world applications.

In addition, current state-of-the-art 3D tracking methods are mostly model-based: the trackers can track the target due to their discriminative ability to recognise targets' categories. For instance, P2B [24] trains the network on human and vehicle data to handle the challenges dedicated in human and vehicle categories respectively. However, object tracking is in essence a class-agnostic task that should track anything regardless of the object category. Moreover, in autonomous driving applications, the target objects are mostly rigid and placed on the ground so that 3D BBox is set as 4DoF (Degree-of-Freedom) or 7DoF for convenience. As a result, the precise 3D description of arbitrary objects is still unavailable which is desirable for generic 3D object tracking.

To this end, in this paper, we propose a novel task for 3D object tracking: given the real 3D BBox description of the target object in the first frame of RGBD videos, we aim to estimate the 3D BBox of it in the subsequent frames. To ensure the generic characteristic of object tracking, we collect a diverse RGBD video dataset for this task. The proposed *Track-it-in-3D* contains 300 video sequences with per-frame 3D annotations. The targets and scenarios are designed with a diverse range to avoid the semantic classification of specific targets. Specifically, the 3D BBox is freely rotating to fit the object's shape and orientation, which breaks the limitation of application scenarios. We provide some representative examples in Fig. 1. In addition, providing the input of RGB and depth data jointly provides new inspirations on how to leverage multi-modal information. Therefore, we propose a strong baseline, which for the first time realises tracking by 3D cross-correlation through dedicated RGBD fusion.

Our contributions are three-fold:

- We propose generic 3D object tracking in RGBD videos for the first time, which aims to realise class-agnostic 3D tracking in complex scenarios.
- We generate the benchmark *Track-it-in-3D*, which is, to the best of our knowledge, the first benchmark for generic 3D object tracking. With dense 3D BBox annotations and corresponding evaluation protocols provided, it contains 300 RGBD videos covering multiple tracking challenges.
- We introduce a strong baseline, *TrackIt3D*, for generic 3D object tracking, which handles 3D tracking difficulties by RGBD fusion and 3D cross-correlation. Extensive evaluations are given for in-depth analysis.

2 Related Work

3D single object tracking. In 3D tracking, the task is defined as getting a 3D BBox in a video sequence given the object template of the first frame. In general, 3D single object tracking is still constrained by tracking on raw point clouds. SC3D [11] extends the 2D to 3D Siamese tracker on point clouds for the first time, in which exhaustive search is used to generate candidates. P2B [24] is proposed to solve the drawbacks of SC3D by importing VoteNet to construct the point-based correlation. Also, the 3D region proposal network (RPN) is utilised to obtain the object proposals. However, the ambiguities among part-aware features weaken the tracking performance severely. After that, BAT [33] is proposed to directly infer the BBox by box-aware feature enhancement, which is the first to use box information. Recent works make multiple attempts with the image prior[34], multi-level features[29], or transformers [8] to handle these problems, but the performances remain low with only point cloud provided. On the other hand, current RGBD tracking follows 2D BBox settings [14,15,16], while there were works devoted to predicting the 2D BBox in 3D view. In 2016, Bibi *et al.* developed 3D-T [3] which used 3D BBox with particle filter for RGBD tracking. In 2018, OTR [12] generated 3D BBox to model appearance changes during out-of-plane rotation. But they only generated incomplete 3D BBoxes in a rough level and served for 2D predictions.

Related datasets and benchmarks. There are four publicly available RGBD video datasets for tracking: *Princeton Tracking Benchmark* (PTB) [26], *Spatio-Temporal Consistency* dataset (STC) [30], *Color and Depth Tracking Benchmark* (CDTB) [19] and *DepthTrack* [31]. We observe that they strictly follow the 2D mode with both input and output as axis-aligned BBoxes. Whereas in 3D tracking, LiDAR is the most popular sensor due to distant view and insensitivity to ambient light variations. The commonly used benchmarks on the 3D tracking task are *KITTI* [10] and *NuScenes* [4]. KITTI contains 21 outdoor scenes and 8 types of targets. NuScenes is more challenging, containing 1000 driving scenes across 23 object classes with annotated 3D BBoxes. With respect to their volume, the data diversity remains poor with focusing on driving scenarios and restraining methods to track objects in point clouds.

3 Proposed Benchmark: *Track-it-in-3D*

3.1 Problem Formulation

In current 3D tracking [24,33], the 3D BBox is represented as $(x, y, z, w, h, l, \theta) \in R^7$, in which (x, y, z) represents the target center and (w, h, l) represents the target size. There is only one parameter θ indicating rotation because the roll and pitch deviations are usually aligned to the road in autonomous driving scenarios. Notice that any BBox is amodal (covering the entire object even if only part of it is visible). The current 3D tracking task is to compare the point clouds of the given template BBox (P_t) with that of the search area candidates (P_s) and get the prediction of BBox. Therefore, the tracking process is formulated as:

$$Track: \ (P_t, P_s) \ \rightarrow \ (x, y, z, \theta).$$

In most cases, because the target size is fixed, the final output only gives a prediction of the target center (x, y, z) and rotation angle θ.

Differing from the existing 3D tracking in point clouds, we explore a more flexible and generic 3D tracking mode. We formulate the new task as:

$$Track: \ B_t \ \rightarrow \ (x, y, z, w, h, l, \alpha, \beta, \gamma),$$

in which B_t is the template 3D BBox given in the first frame, (x, y, z) indicates the target position, (w, h, l) indicates the target scale, and (α, β, γ) indicates the target rotation angle. Specifically, this tracking problem predicts a rotated 3D BBox to best match the initial target.

3.2 Dataset Construction

Video collection. We collect the videos with *Microsoft Kinect V2* and *Intel RealSense SR300* for different depth ranges. We aim to provide a diverse set of groundtruthed synchronised color and depth sequences for generic 3D tracking, in which diversity is of priority. To this end, we carefully inspect each sequence

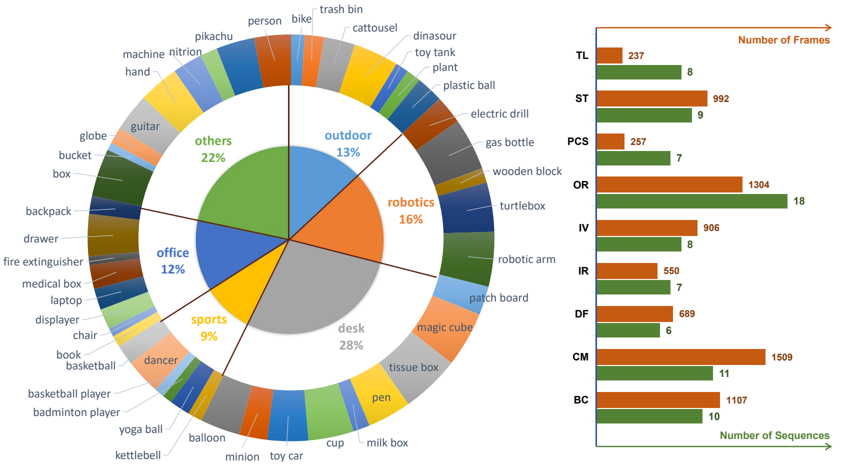

Fig. 2. Distribution of the object, scenarios and challenges in all test frames. Left: The inner pie-chart shows the distribution of the scenarios; The outside ring graph shows our target objects. Right: Brown histogram shows the attribute distribution on frame level; Green histogram shows the attribute distribution on sequence level.

among all candidate data for the availability and challenge for generic 3D tracking. Examples of some representative sequences are shown in Fig. 1. Finally, *Track-it-in-3D* comprises a total of 300 sequences with the data split as such: 250 sequences (32,343 frames) for training, and 50 sequences (6,224 frames) for testing. All the videos are captured at $30fps$. We do not provide a further partition to leave users with the freedom of the training/validation split. We provide the distribution of scenarios and objects in our test set in Fig. 2. We keep our test set compact but diverse for a fair and effective evaluation.

Attribute definition. Based on characteristics of the aforementioned problem, we annotate all the frames with 9 attributes to analyse how different kinds of challenges influence the tracking performance: Background Clutter (BC), Camera Motion (CM), Deformation (DF), In-plane Rotation (IR), Illumination Variation (IV), Out-of-plane Rotation (OR), Similar Targets (ST), Target Loss (TL) and Point Cloud Sparsity (PCS). Among them, background clutter, similar targets, and illumination variation are close related to depth favorable scenarios. In addition, point cloud sparsity, in-plane rotation and out-of-plane rotation are specifically challenging to 3D scenes. Unlike existing attributes in 3D tracking datasets, we are the first 3D dataset to provide detailed visual attributes according to both objects and scenarios. Distribution of attributes is given in Fig. 2. For detailed description of the attributes, please refer to the supplementary.

Data annotation. For annotation, we manually annotate each target object in the video sequences with per-frame rotated 3D BBox on our modified version

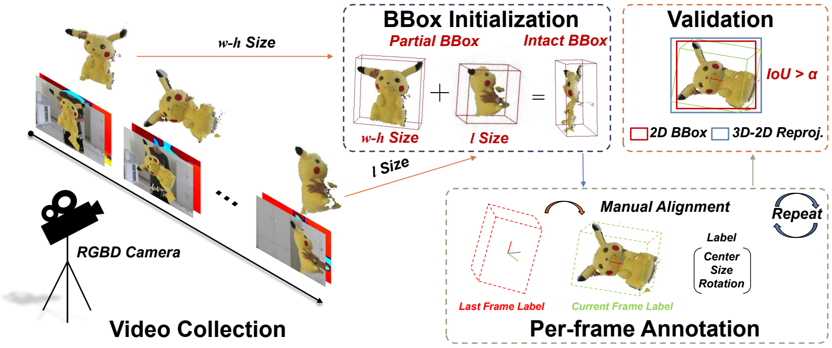

Fig. 3. Steps of our data annotation strategy. *BBox Initialisation:* We complete the size of the initial BBox from multi-view partial BBoxes. *Per-frame Annotation:* Similar to the tracking pipeline, annotators align the last-frame BBox with the current-frame object and record the label. *Validation:* We re-project the 3D BBox to image and generate 2D BBox. By computing the IoU between the re-projected 2D BBox with annotated 2D BBox, the accuracy of 3D annotation can be verified.

of SUSTechPoints tool [17]. We follow this principle for data annotation: given an initial target description (3D BBox) in a video, if the target appears in the subsequent frames, we will edit the 3D BBox to tightly covering the whole target; otherwise, we will maintain the BBox state from the adjacent frame, and annotate the current frame with a "target loss" label. To guarantee annotation accuracy, we adopt a three-stage annotation strategy: 1) *BBox initialisation:* we firstly go through the whole sequences to best describe the target size (w, h, l) and give an initial 3D BBox. For example, we may not get precise length l_p in the first frame, but we can get precise width w_p and height h_p with an estimated length l_e of the target. Then we will go through the whole video to find the frame best showing the precise length l_p of the target, duplicate the 3D box to the first frame, and finally fine-tune the 3D box to get a precise length l_p for the target. 2) *Per-frame annotation:* an annotator edits the initial BBox in the subsequent frames to make the BBox best fit the target; the annotator can change the BBox's location and angle, and size if necessary (for cases like deformable objects) in this stage; 3) *Validation:* the authors finally check the annotation frame by frame to verify the annotation accuracy. The annotation workflow is shown in Fig. 3, which ensures high-quality annotation BBoxes in 3D scenes. Under such strategy, we can obtain the intact target BBox of the target in the specific frame, while it is tightest to fit the object with containing the real target size information in 3D space. We also evaluate our annotation accuracy with projection and sampling, please refer to the supplementary material.

3.3 Evaluation Protocols

To judge the quality of 3D tracking, measures are designed to reflect the 3D BBox tracking performance. Therefore, we follow the One Pass Evaluation (OPE) and the standard evaluation protocols to calculate the object center bias and 3D IoU accuracy. In the following, we present our evaluation protocols.

Precision plot. One widely used evaluation metric for object tracking is the center bias, which is used to measure the Euclidean distance between the centers of predicted BBox and groundtruth BBox. We present the precision plots of the trackers averaged over all sequences with the threshold from $0m$ to $0.5m$. We obtain the area-under-curve (AUC) of a tracker's precision plot as its "Precision".

Success plot. As we propose the rotated 3D BBox description in the 3D tracking scenes, 3D Intersection-over-Union (IoU) is essential to measure the tracking accuracy. According to [9], we provide the IoU measure for general 3D-oriented boxes based on the Sutherland-Hodgman Polygon clipping algorithm. We firstly clip each face as the convex polygon between the predicted box and the groundtruth box. Then, the IoU is computed from the volume of the intersection and the volume of the union of two boxes by swapping the two boxes. AUC in success plot of IoU between groundtruth and predicted BBox is defined as "Success". For details, we refer readers to [9,1].

3.4 Comparison with related tasks

As shown in Fig. 4, we compare our 3D object tracking in RGBD videos with related tasks [10,25,31,28]. Compared to current *3D object tracking in point clouds* [10], we provide corresponding synchronised color information besides point clouds. Furthermore, instead of tracking with (x, y, z, θ), which only describes the location of the target center and one-dimensional rotation, we require a more flexible bounding box to better fit the object. Similarly, *3D object detection* [25] is to classify objects in image level, which also places all objects on the plane and cannot give a precise description for genric objects *e.g.*, suspended or sloping objects. Compared to *RGBD tracking*, [31] which remains on tracking the object within 2D settings, our proposed task requires a more detailed description of the object in the spatial domain. In addition, *6D pose tracking* [28] focuses on describing the pose of specific objects, which is heavily model-based. Different from existing tasks, 3D single object tracking (SOT) in RGBD videos is more challenging, in which the objects, scenarios, and annotations are more diverse and flexible. A detailed comparison of the proposed *Track-it-in-3d* with representative datasets from related tasks is summarised in Table 1. Although the proposed dataset is not prominent on volume compared to existing datasets, it can represent characteristics of the 3D tracking more effectively: 1) It achieves a high diversity for class-agnostic 3D tracking with covering indoor and outdoor scenarios, class-agnostic target objects and freely rotated 3D target annotation. 2) It provides a more effective way to track objects in 3D scenes with providing synchronised RGB and depth information.

Table 1. Comparison with related datasets. I=Indoor, O=Outdoor. We are the first dataset that provides 3D annotations for dynamic objects to realise generic 3D single object tracking in natural scenes.

Dataset	Type	Task	Modality	Sequence	Frame	Label	Class	Scenario	Dynamic
DepthTrack[31]	Video	RGBD Tracking	RGB+D	200	294K	2D	46	I,O	✓
SUN-RGBD[25]	Image	3D Detection	RGB+D	-	10K	3D	63	I	×
Objectron[1]	Video	3D Detection	RGB	14,819	4M	3D	9	I,O	×
NOCS[28]	Image	Pose Tracking	RGB+D	-	300K	3D	6	I,O	×
KITTI[10]	Video	3D Tracking	PC	21	15K	3D	8	O	✓
NuScenes[4]	Video	3D Tracking	PC	1,000	40K	3D	23	O	✓
Track-it-in-3D	Video	3D Tracking	RGB+D	300	36K	3D	144	I,O	✓

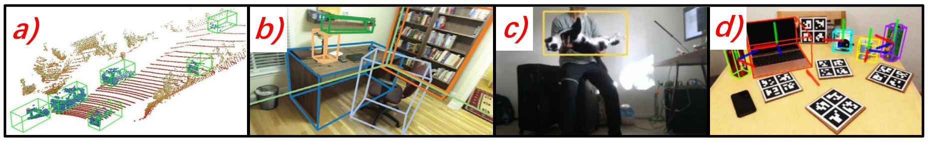

Fig. 4. Samples from related tasks and corresponding datasets, which basically show the object/scenario/annotation styles. a) KITTI [10], b) SUN-RGBD [25], c) DepthTrack [31], d) NOCS [28].

4 Proposed Baseline: *TrackIt3D*

Sole RGB based and point cloud based trackers already exist, and they perform well in specific cases respectively. Here, we propose a generic 3D tracker, namely *TrackIt3D*, which fuses the RGB and depth information in a seamless way. In this section, we first describe the overall network architecture, including the main components, then illustrate our implementation details.

4.1 Network Architecture

The input of our network is two frames from an RGBD video, defined as a target template frame and a search area frame respectively. The goal is simplified to localise the template target in the search area per frame. Our network consists of three main modules as shown in Fig. 5. We first design a Siamese RGBD Fusion Network to fuse the surface information (RGB Info.) and the spatial information (XYZ Info.) together. Next, the 3D Cross-Correlation Network is proposed to merge the template information into the search area. Finally, the fused feature is fed into the VoteNet module [22] to yield 3D BBox and confidence scores via the proposed BBox Loss and IoU Loss.

Siamese RGBD fusion network. The key idea of our fusion network is to enable surface information and spatial information to complement each other. To better exploit the spatial information of the depth map, we convert the depth image to a point cloud. Given the RGBD template t and search area s, our network

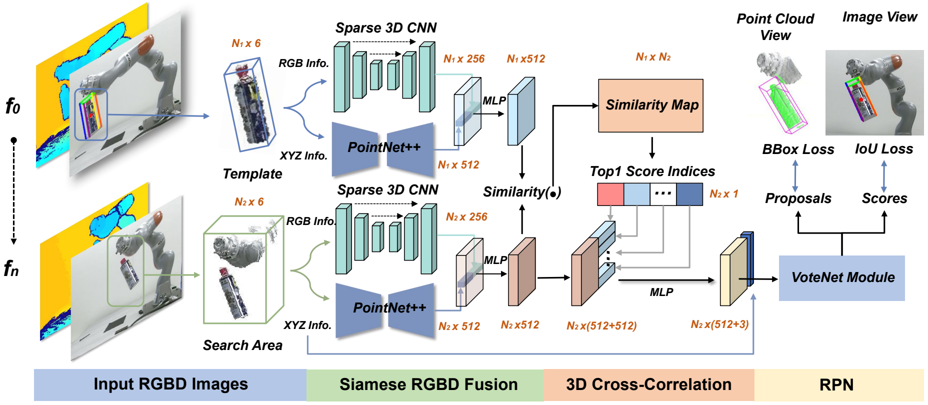

Fig. 5. Overview of our baseline TrackIt3D. The target gas bottle is moving with the robotic arm, tied by a transparent rope. The inputs are pixels and points of the template and search area, with the number of $N1$ and $N2$ respectively. The Siamese RGBD Fusion Network fuses the surface information (RGB Info.) and the spatial information (XYZ Info.). The Cross-Correlation Network learns the similarity between the template and the search area features. We use the BBox Loss and IoU Loss to enforce the VoteNet module [22] to yield the 3D BBox and corresponding confidence scores.

first associates each point to its corresponding image pixel based on projection onto the image plane using the known camera intrinsic parameters. The obtained pairs P of template and search area are then downsampled to $P^t \in \mathbb{R}^{N_1 \times 6}$ and $P^s \in \mathbb{R}^{N_2 \times 6}$ separately. Every pair P is represented as (x, y, z, R, G, B), in which (x, y, z) indicates the target spatial information and (R, G, B) indicates the surface information. We adopt an encoder-decoder structure with skip connections constructed by sparse 3D CNN [6], to extract the pixel-wise feature map $f^t_{rgb} \in \mathbb{R}^{N_1 \times 256}$ and $f^s_{rgb} \in \mathbb{R}^{N_2 \times 256}$ from the sparse surface pixels. We also implement a variant of the PointNet++ [23] architecture, with adding a decoder with skip connections to generate dense point-wise feature maps $f^t_{xyz} \in \mathbb{R}^{N_1 \times 512}$ and $f^s_{xyz} \in \mathbb{R}^{N_2 \times 512}$. The output feature maps of sparse 3D CNN and Pointnet++ are then concatenated and fed to a MLP network to generate the fused feature maps $f^t \in \mathbb{R}^{N_1 \times 512}$ and $f^s \in \mathbb{R}^{N_2 \times 512}$.

3D cross-correlation network. Learning to track arbitrary objects can be addressed by similarity matching [2]. Following this, our 3D cross-correlation network learns to conduct a reliable similarity between the template features and the search area features. Different from unordered point sets [24], our points are in order because of pixel and point alignment, so that we can do similarity matching directly over 3D feature maps. As shown in Fig. 5, after obtaining the fused feature maps of the template and search area, we can compute the similarity map $Sim \in \mathbb{R}^{N_1 \times N_2}$ between f^t and f^s using the cosine distance. The column i in Sim means the similarity score of each feature in f^t to the i^{th} feature in f^s. We then find the top score of i column, which represents the most similar template feature to the i^{th} search feature. After getting all top score indices, we search the template feature by the index in f^t and then concatenate it with the corresponding feature in f^s, yielding a feature map of size $N_2 \times (512+512)$. Then we feed it into an MLP network to obtain the final feature map $f \in \mathbb{R}^{N_1 \times 512}$. The point-wise feature map f and the corresponding 3D position of each point are fed to the VoteNet module to obtain the final 3D BBox.

Loss function. We train our network with the following loss function:

$$\mathcal{L}_{\text{total}} = \lambda_1 \mathcal{L}_{\text{reg}} + \lambda_2 \mathcal{L}_{\text{bbox}} + \lambda_3 \mathcal{L}_{\text{IoU}}. \tag{1}$$

Following [22], a shared voting module is used to predict the coordinate offset between points and target center. The predicted 3D offset is supervised by Vote loss $\mathcal{L}_{\text{reg}}$, which enforces the network to produce potential centers of the object. BBox loss $\mathcal{L}_{\text{bbox}}$ is designed to pull the K proposal BBoxes closer to the groundtruth BBox. Our 3D groundtruth BBox is defined by $\bar{B} = \left[\bar{x}, \bar{y}, \bar{z}, \bar{w}, \bar{h}, \bar{l}, \bar{q}\right]$, in which quaternion q represents the rotation. The BBox loss is computed via Huber (smooth-L1) loss:

$$\mathcal{L}_{\text{bbox}} = \frac{1}{K} \sum_{i}^{K} \left\| B_i - \bar{B}_i \right\|_1 . \tag{2}$$

IoU loss $\mathcal{L}_{\text{IoU}}$ aims to ensure that the confidence score S_k approximates the IoU between proposals and groundtruth BBox. Following [9], we compute the IoU between the two 3D BBoxes based on the Sutherland-Hodgman Polygon clipping algorithm. The loss function is written as follow:

$$\mathcal{L}_{\text{IoU}} = \frac{1}{K} \sum_{i=1}^{K} \left\| IoU_k - S_k \right\|_1 . \tag{3}$$

4.2 Implementation Details

Architecture. For our network, we downsample the points and pixels for template and search area to $N_1 = 512$ and $N_2 = 1024$. The cluster parameter in the VoteNet module is $K = 64$. The coefficients for the loss terms are $\lambda_1 = 1$, $\lambda_2 = 0.5$ and $\lambda_3 = 0.5$.

Training phase. We train our model using the training set discussed in Sec. 3.2 which consists of RGBD videos and 3D object bounding box annotations. 1) *Template and Search Area:* we randomly sample RGBD image pairs from all the videos with a maximum gap of 10 frames. In each pair, the first image will serve as the template and the second will be the search area. The template is generated by cropping pixels and points inside the first given 3D BBox and we enlarge the second BBox by 4 times in each direction and collect pixels and points inside to generate the search area. 2) *3D Deformation:* to handle the shape variation of the target, we generate the augmented data for each pair by enlarging, shrinking, or changing some part of the point cloud following [5]. 3) The learning rate is 0.001, the batch size is 50, Adam [13] is adopted as an optimiser and trained for a total of 120 epochs. The learning rate decreased by 5 times after 50 epochs.

Inference phase. During the inference, we also use the proposed dataset in Sec. 3.2. Different from the training phase, we track a target across all RGBD frames in a video. The given 3D BBox will be used to crop the template area, and the search area of the current frame is generated by enlarging (by 4 times in each direction) the predicted 3D BBox in the last frame and collecting the pixels and points in it.

5 Experiments

5.1 Benchmark Settings

As our proposed *TrackIt3D* is the first tracker designed for generic 3D tracking, we evaluate some representative 3D trackers based on point clouds for comparison. The compared trackers are SC3D[11], P2B[24], and BAT[33]. For model-based 3D trackers, we evaluate their default pre-trained models and the models finetuned on our proposed training set (if the model is trainable). Experiments are run on a single NVIDIA Tesla V100S GPU 32GB.

5.2 Benchmark Results

Overall results. Table 2 gives the comparison results of 3D trackers. Our method achieves the highest score compared to the existing ones, in terms of both Success (31.1%) and Precision (35.0%). With dedicated combination of color and depth modalities, TrackIt3D is capable to distinguish the object in the RGB domain and makes good predictions of 3D BBox in the point cloud domain. It is worth noting that the SC3D, which performs worse on KITTI compared with P2B and BAT, shows a better performance on our test set even without finetuning on the proposed training set. The reason is that SC3D aims to compare the similarity between the template and 3D target proposals, while P2B and BAT utilise VoteNet to vote an object center, which tends to learn the center location based on strong category-related priors. We use their car-based model for testing. Therefore, when facing the class-agnostic tracking sequences, the sole VoteNet is not enough for center prediction. The P2B and BAT show remarkable improvements after finetuning on our training set. However, they still suffer low scores because the threshold of the center error is around 0.5m in our proposed dataset, while it is 2m in KITTI [10]. In addition, they can only regress an axis-aligned BBox while we get a 9DoF BBox which contributes to a higher IoU score. We show the precision and success plots in Fig. 6.

Table 2. Quantitative comparison between our method and state-of-the-art methods. Our method outperforms the compared models by a large margin on our *Track-it-in-3d* test set. Speed is also listed and "_ft" means the method is finetuned on our training dataset. **Bold** denotes the best performance.

Tracker	SC3D[11]	P2B[24]	P2B_ft[24]	BAT[33]	BAT_ft[33]	TrackIt3D
Success	9.2%	4.2%	9.4%	2.5%	2.5%	**31.1%**
Precision	6.8%	1.1%	8.4%	0.8%	4.7%	**35.0%**
Speed(FPS)	0.51	23.78	21.25	**28.17**	25.08	6.95

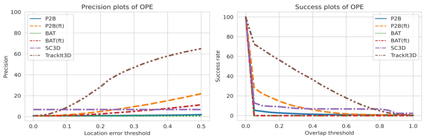

Fig. 6. The Success and Precision plots of the compared trackers and the proposed *TrackIt3D*.

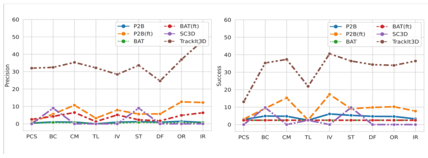

Fig. 7. Optimal Precision (left) and Success (right) scores over the visual attributes.

Fig. 8 shows several representative samples of results comparing our *TrackIt3D* with finetuned P2B. As shown, unlike P2B which only gives an axis-aligned estimation of the target object, our *TrackIt3D* can also distinguish the target orientation and track the target rotation. Specifically, row a) shows a scene with similar objects, in which P2B fails in total while our method can accurately track the target object. Besides, our method is more robust to challenging cases like object rotation and deformation, as shown in rows b) and c), due to its strong discriminative ability based on RGBD fusion. Moreover, row d) gives an outdoor scenario under low illumination, where it is difficult to locate the object, but our method shows a good estimation. The last row gives a failed case in which the target is severely occluded by a plant, both *TrackIt3D* and P2B fail due to their lack of a re-detection mechanism.

Attribute-based results. Per-attribute results are reported in Fig. 7. Although the overall performance is low, we can obtain informative analysis from the per-attribute result. Our method obviously outperforms the compared models in all attributes, especially in in-plane rotation and illumination variation. Clearly, the superior performance of our RGBD fusion over point cloud is evident. However, *TrackIt3D*'s success score degrades severely on the point cloud sparsity and target loss, indicating that it still need improvement on long-term discriminative ability and target localisation under little spatial information. Despite that, it is worth noting that the finetuned P2B performs well under in-camera motion and illumination variation, while SC3D beats the other trackers on background clutter and similar targets.

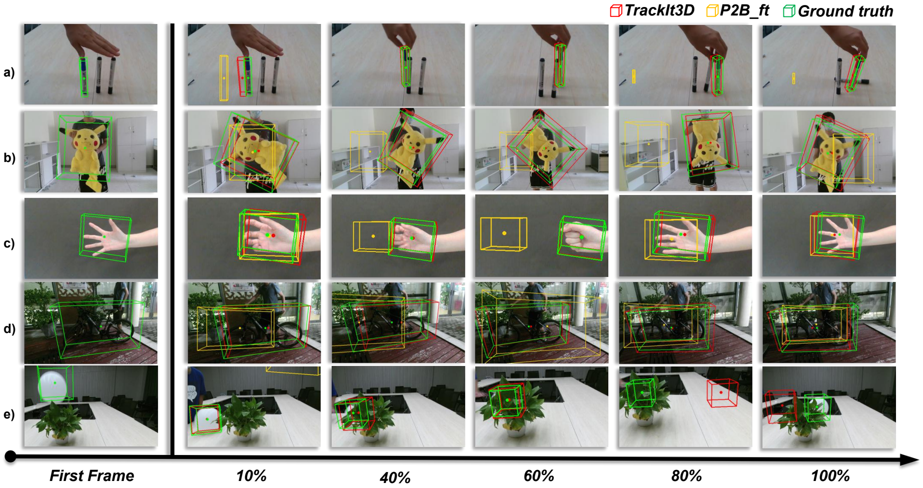

Fig. 8. Qualitative results of our baseline *TrackIt3D* compared with the fine-tuned *P2B*. We can observe our baseline's advantage over P2B in many challenge scenarios, *e.g.*, a) similar objects, b) rotation, c) deformation and d) dark scene. The last row is a failed case when the object is fully occluded.

Table 3. Performance of the RGBD variant of original 3D point cloud tracker, and P2B++ and BAT++ have been finetuned on our training dataset.

Tracker	P2B++ [24]	BAT++ [33]	TrackIt3D
Success	24.5%	18.1%	31.1%
Precision	28.2%	26.0%	35.0%

5.3 Ablation Study

Effectiveness of RGBD fusion. To validate the effectiveness of the proposed RGBD fusion on 3D tracking, we apply it on P2B and BAT to instead their original heads and obtains corresponding variants P2B++ and BAT++. Table 3 shows the comparison between the variants with the RGBD fusion head and our *TrackIt3D*. Specifically, there are striking improvements (at least 15.1% and 19.8%) in terms of Success and Precision compared with the finetuned P2B and BAT, which proves that the RGBD fusion boosts the performance of point cloud voting models. Also, performance of BAT++ is lower that the P2B++ due to its strong object prior with fixed size.

Different ways for 3D cross-correlation. Besides our default settings in Sec. 4.1, we consider other possible ways for 3D cross-correlation, *e.g.*, 2D correlation [18], which is commonly used in 2D tracking, instead of 3D correlation. The left section in Fig. 9 shows how we implement 2D correlation. Surprisingly,

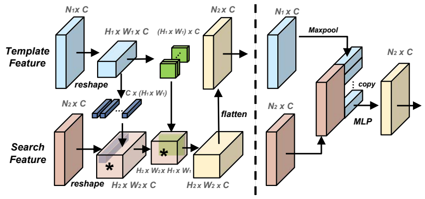

Fig. 9. Different ways for 3D cross-correlation. The left part is following 2D tracking pipeline. The right part is without calculating similarity map. $*$ means convolution operation.

Ways for 3D xcorr.	Success	Precision
our default setting	**31.1%**	35.0%
w/ 2D xcorr. setting	28.3%	**38.4%**
w/o similarity map	30.9%	33.1%
w/o template feature	7.0%	5.0%

Table 4. Different ways for 3D cross-correlation (xcorr.). Methods for similarity learning between search features and template following 2D tracking method are illustrated in Fig. 9.

results in Table. 4 show that the 2D correlation setting outweighs our 3D correlation on Precision, although it gives a lower Success. This may reveal that the 2D-based method is more robust to estimate an accurate target center, while it is weaker on 3D BBox prediction as it omits the spatial correlation in 3D space. We also try to remove the similarity map and template feature, as shown in the right part of Fig. 9. The performance degrades without using the two parts. Specifically, once removing the template feature, Success and Precision degrade with 7% and 5%, which proves that the tracker loses the discriminative ability without the reference feature.

6 Conclusions

In this paper, we investigate a novel topic to track generic objects with 3D rotated BBox in RGBD videos. We first construct a novel benchmark *Track-it-in-3D* with 300 RGBD videos for training and testing, which covers diverse objects and challenging scenarios in 3D scenes. Also, this benchmark enables generic 3D tracking in complex scenarios with novel target annotation and performance evaluation. Furthermore, we propose an end-to-end method *TrackIt3D* for tracking class-agnostic 3D objects. With effective RGBD fusion and 3D cross-correlation, our baseline shows superior performance on this challenging task. We hope this work will facilitate further research on generic 3D tracking.

Acknowledgements

This work is supported by the National Natural Science Foundation of China under Grant No. 61972188 and 62122035. Z.Z. was supported by China Scholarship Council (CSC) Grant No. 202208060266. H.C. was supported by Institute of Information and communications Technology Planning and evaluation (IITP) grant funded by the Korea government (MSIT) (2021-0-00537, Visual common sense through self-supervised learning for restoration of invisible parts in images). A.L. was supported in part by the Engineering and Physical Sciences Research Council (grant number EP/S032487/1).

References

1. Ahmadyan, A., Zhang, L., Ablavatski, A., Wei, J., Grundmann, M.: Objectron: A large scale dataset of object-centric videos in the wild with pose annotations. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 7822–7831 (2021)
2. Bertinetto, L., Valmadre, J., Henriques, J.F., Vedaldi, A., Torr, P.H.: Fully-convolutional siamese networks for object tracking. In: European conference on computer vision. pp. 850–865. Springer (2016)
3. Bibi, A., Zhang, T., Ghanem, B.: 3d part-based sparse tracker with automatic synchronization and registration. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1439–1448 (2016)
4. Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G., Beijbom, O.: nuscenes: A multimodal dataset for autonomous driving. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 11621–11631 (2020)
5. Chen, W., Jia, X., Chang, H.J., Duan, J., Shen, L., Leonardis, A.: Fs-net: Fast shape-based network for category-level 6d object pose estimation with decoupled rotation mechanism. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1581–1590 (2021)
6. Choy, C., Gwak, J., Savarese, S.: 4d spatio-temporal convnets: Minkowski convolutional neural networks. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3075–3084 (2019)
7. Comport, A.I., Marchand, É., Chaumette, F.: Robust model-based tracking for robot vision. In: 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)(IEEE Cat. No. 04CH37566). vol. 1, pp. 692–697. IEEE (2004)
8. Cui, Y., Fang, Z., Shan, J., Gu, Z., Zhou, S.: 3d object tracking with transformer. arXiv preprint arXiv:2110.14921 (2021)
9. Ericson, C.: Real-time collision detection. Crc Press (2004)
10. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: 2012 IEEE conference on computer vision and pattern recognition. pp. 3354–3361. IEEE (2012)
11. Giancola, S., Zarzar, J., Ghanem, B.: Leveraging shape completion for 3d siamese tracking. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
12. Kart, U., Lukezic, A., Kristan, M., Kamarainen, J.K., Matas, J.: Object tracking by reconstruction with view-specific discriminative correlation filters. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 1339–1348 (2019)
13. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
14. Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder, R., Kamarainen, J.K., Čehovin Zajc, L., Danelljan, M., Lukezic, A., Drbohlav, O., He, L., Zhang, Y., Yan, S., Yang, J., Fernandez, G., et al.: The eighth visual object tracking vot2020 challenge results (2020)
15. Kristan, M., Matas, J., Leonardis, A., Felsberg, M., Pflugfelder, R., Kamarainen, J.K., Cehovin Zajc, L., Drbohlav, O., Lukezic, A., Berg, A., et al.: The seventh visual object tracking vot2019 challenge results. In: Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops (2019)

16. Kristan, M., Matas, J., Leonardis, A., Felsberg, M., Pflugfelder, R., Kamarainen, J.K., Chang, H.J., Danelljan, M., Cehovin, L., Lukezic, A., et al.: The ninth visual object tracking vot2021 challenge results. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 2711–2738 (2021)
17. Li, E., Wang, S., Li, C., Li, D., Wu, X., Hao, Q.: Sustech points: A portable 3d point cloud interactive annotation platform system. In: 2020 IEEE Intelligent Vehicles Symposium (IV). pp. 1108–1115 (2020). https://doi.org/10.1109/IV47402.2020.9304562
18. Liao, B., Wang, C., Wang, Y., Wang, Y., Yin, J.: Pg-net: Pixel to global matching network for visual tracking. In: European Conference on Computer Vision. pp. 429–444. Springer (2020)
19. Lukezic, A., Kart, U., Kapyla, J., Durmush, A., Kamarainen, J.K., Matas, J., Kristan, M.: Cdtb: A color and depth visual object tracking dataset and benchmark. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10013–10022 (2019)
20. Luo, W., Yang, B., Urtasun, R.: Fast and furious: Real time end-to-end 3d detection, tracking and motion forecasting with a single convolutional net. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. pp. 3569–3577 (2018)
21. Machida, E., Cao, M., Murao, T., Hashimoto, H.: Human motion tracking of mobile robot with kinect 3d sensor. In: 2012 Proceedings of SICE Annual Conference (SICE). pp. 2207–2211. IEEE (2012)
22. Qi, C.R., Litany, O., He, K., Guibas, L.J.: Deep hough voting for 3d object detection in point clouds. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 9277–9286 (2019)
23. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. arXiv preprint arXiv:1706.02413 (2017)
24. Qi, H., Feng, C., Cao, Z., Zhao, F., Xiao, Y.: 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 (CVPR) (June 2020)
25. Song, S., Lichtenberg, S.P., Xiao, J.: Sun rgb-d: A rgb-d scene understanding benchmark suite. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 567–576 (2015)
26. Song, S., Xiao, J.: Tracking revisited using rgbd camera: Unified benchmark and baselines. In: Proceedings of the IEEE international conference on computer vision. pp. 233–240 (2013)
27. Taylor, C., McNicholas, R., Cosker, D.: Towards an egocentric framework for rigid and articulated object tracking in virtual reality. In: 2020 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW). pp. 354–359 (2020). https://doi.org/10.1109/VRW50115.2020.00077
28. Wang, H., Sridhar, S., Huang, J., Valentin, J., Song, S., Guibas, L.J.: Normalized object coordinate space for category-level 6d object pose and size estimation. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
29. Wang, Z., Xie, Q., Lai, Y.K., Wu, J., Long, K., Wang, J.: Mlvsnet: Multi-level voting siamese network for 3d visual tracking. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 3101–3110 (2021)
30. Xiao, J., Stolkin, R., Gao, Y., Leonardis, A.: Robust fusion of color and depth data for rgb-d target tracking using adaptive range-invariant depth models and spatio-temporal consistency constraints. IEEE transactions on cybernetics **48**(8), 2485–2499 (2017)

31. Yan, S., Yang, J., Kapyla, J., Zheng, F., Leonardis, A., Kamarainen, J.K.: Depthtrack: Unveiling the power of rgbd tracking. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 10725–10733 (2021)
32. Yan, X., Zheng, C., Li, Z., Wang, S., Cui, S.: Pointasnl: Robust point clouds processing using nonlocal neural networks with adaptive sampling. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5589–5598 (2020)
33. Zheng, C., Yan, X., Gao, J., Zhao, W., Zhang, W., Li, Z., Cui, S.: Box-aware feature enhancement for single object tracking on point clouds. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 13199–13208 (2021)
34. Zou, H., Cui, J., Kong, X., Zhang, C., Liu, Y., Wen, F., Li, W.: F-siamese tracker: A frustum-based double siamese network for 3d single object tracking. In: 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). pp. 8133–8139. IEEE (2020)