Segment as Points for Efficient Online Multi-Object Tracking and Segmentation

Zhenbo Xu[1,2][0000−0002−8948−1589], Wei Zhang[2], Xiao Tan[2], Wei Yang[1*][0000−0003−0332−2649], Huan Huang[1], Shilei Wen[2], Errui Ding[2], and Liusheng Huang[1]

[1] University of Science and Technology of China
[2] Department of Computer Vision Technology (VIS), Baidu Inc., China
[*] Corresponding Author. E-mail: qubit@ustc.edu.cn

Abstract. Current multi-object tracking and segmentation (MOTS) methods follow the tracking-by-detection paradigm and adopt convolutions for feature extraction. However, as affected by the inherent receptive field, convolution based feature extraction inevitably mixes up the foreground features and the background features, resulting in ambiguities in the subsequent instance association. In this paper, we propose a highly effective method for learning instance embeddings based on segments by converting the compact image representation to un-ordered 2D point cloud representation. Our method generates a new tracking-by-points paradigm where discriminative instance embeddings are learned from randomly selected points rather than images. Furthermore, multiple informative data modalities are converted into point-wise representations to enrich point-wise features. The resulting online MOTS framework, named PointTrack, surpasses all the state-of-the-art methods including 3D tracking methods by large margins (5.4% higher MOTSA and 18 times faster over MOTSFusion) with the near real-time speed (22 FPS). Evaluations across three datasets demonstrate both the effectiveness and efficiency of our method. Moreover, based on the observation that current MOTS datasets lack crowded scenes, we build a more challenging MOTS dataset named APOLLO MOTS with higher instance density. Both APOLLO MOTS and our codes are publicly available at https://github.com/detectRecog/PointTrack.

Keywords: Motion and Tracking, Tracking, Vision for Robotics

1 Introduction

Multi-object tracking (MOT) is a fundamental task in computer vision with broad applications such as autonomous driving and video surveillance. Recent MOT methods [4, 6, 41] mainly adopt the tracking-by-detection paradigm which links detected bounding boxes across frames via data association algorithms. Since the performance of association highly depends on robust similarity measurements, which is widely noticed difficult due to the frequent occlusions among

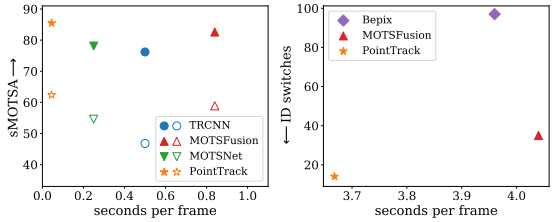

Fig. 1. Comparison between our PointTrack and the state-of-the-art MOTS methods on sMOTSA (Left) and id switches (Right). On the left subfigure, the filled symbols and the hollow symbols denote the results for cars and for pedestrians respectively. On the right subfigure, all methods perform tracking on the same segmentation result, which takes 3.66s.

targets, challenges remain in MOT especially for crowded scenes [2]. More recently, the task of multi-object tracking and segmentation (MOTS) [34] extends MOT by jointly considering instance segmentation and tracking. As instance masks precisely delineate the visible object boundaries and separate adjacency naturally, MOTS not only provides pixel-level analysis, but more importantly encourages to learn more discriminative instance features to facilitate robust similarity measurements than bounding box (bbox) based methods.

Unfortunately, how to extract instance feature embeddings from segments have rarely been tackled by current MOTS methods. TRCNN [34] extends Mask RCNN by 3D convolutions and adopts ROI Align to extract instance embeddings in bbox proposals. To focus on the segment area in feature extraction, Porzi *et al.* [27] propose mask pooling to replace ROI Align. Nevertheless, as affected by the receptive field of convolutions, the foreground features and the background features are still mixed up, which is harmful for learning discriminative feature. Therefore, though current MOTS methods adopt advanced segmentation networks to extract image features, they fail to learn discriminative instance embeddings which are essential for robust instance association, resulting in limited tracking performances.

In this paper, we propose a simple yet highly effective method to learn instance embeddings on segments. Inspired by the success of PointNet [28] which enables feature aggregations directly from irregular formatted 3D point clouds, we regard 2D image pixels as un-ordered 2D point clouds and learn instance embeddings in a point cloud processing manner. Concretely, for each instance, we build two separate point clouds for the foreground segment and the surrounding area respectively. In each point cloud, we further propose to combine different modalities of point-wise features to realize a unified and context-aware instance embedding. In this way, the novel tracking-by-points paradigm can be easily established by combining our proposed instance embedding with any instance segmentation method. The effectiveness of our proposed instance embedding method is examined through a comparison with current MOTS approaches based on the same segmentation results. As shown in the right subfigure of Fig. 1, our

method reduces id switches significantly. Evaluations across different datasets (see PointTrack* in Table 3,5) also prove the strong generalization ability of our proposed instance embedding. Besides, to enable the practical utility of MOTS, we enhance the state-of-the-art one-stage instance segmentation method SpatialEmbedding [23] for temporal coherence and build up a novel MOTS framework named PointTrack. Our proposed framework first achieves nearly real-time performance while out-performs all the state-of-the-art methods including 3D tracking methods on KITTI MOTS by large margins (see the left subfigure of Fig. 1).

Moreover, to facilitate better evaluations, we construct a more crowded thus more challenging MOTS dataset named APOLLO MOTS based on the public ApolloScape dataset [13]. APOLLO MOTS has a similar number of frames with KITTI MOTS but two times more tracks and car annotations (see Table 1). We believe APOLLO MOTS can further help promote researches in MOTS.

We summarize our main contributions as follows:

- We propose a highly effective method for learning discriminative instance embeddings on segments by breaking the compact image representation into un-ordered 2D point clouds.
- A novel online MOTS framework named PointTrack is introduced, which is more efficient and more effective than the state-of-the-art methods.
- We build APOLLO MOTS, a more challenging dataset with 68% higher instance density over KITTI MOTS.
- Evaluations across three datasets show that PointTrack outperforms all existing MOTS methods by large margins. Also, PointTrack can reduce id switches significantly and generalizes well on instance embedding extraction.

2 Related Work

Tracking-by-Detection. Detection based MOT approaches first detect objects of interests and then link objects into trajectories via data association. The data association can be accomplished on either the 2D image plane [4, 6, 7, 14, 32, 41, 37] or the 3D world space [1, 8, 10, 20, 24, 38]. ATOM [7] introduces a novel tracking architecture, which consists of dedicated target estimation and classification components, by predicting the overlap between the target object and an estimated bounding box. FAMNet [6] develops an end-to-end tracking architecture where feature extraction, affinity estimation and multi-dimensional assignment are jointly optimized. Most 3D tracking methods [24, 31] merge track-lets based on 3D motion clues. Other approaches [10, 22, 18] further perform 3D reconstruction for objects to improve the tracking performance.

Tracking-by-Segmentation. Unlike 2D bounding boxes which might overlap heavily in crowded scenes, per-pixel segments locate objects precisely. Recently instance segments have been exploited for improving the tracking performance [19, 26, 25, 12, 27]. In [25], Osep *et al.* present a model-free multi-object tracking approach that uses a category-agnostic image segmentation method to track objects. Track-RCNN [34] extends Mask-RCNN with 3D convolutions to

incorporate temporal information and extracts instance embeddings for tracking by ROI Align. MOTSNet [27] proposes a mask pooling layer to Mask-RCNN to improve object association over time. STE [12] introduces a new spatial-temporal embedding loss to generate temporally consistent instance segmentation and regard the mean embeddings of all pixels on segments as the instance embedding for data association. As features obtained by 2D or 3D convolutions are harmful for learning discriminative instance embeddings, different from previous methods, our PointTrack regards 2D image pixels as un-ordered 2D point clouds and learn instance embeddings in a point cloud processing manner.

MOTS Datasets. KITTI MOTS [34] extends the popular KITTI MOT dataset with dense instance segment annotations. Except for KITTI MOTS, popular datasets (like the ApolloScape dataset [13]) also provide video instance segmentation labels, but the instances are not consistent in time. Compared with KITTI MOTS, ApolloScape provides more crowded scenes which are more challenging for tracking. Based on this observation, we build Apollo MOTS in a semi-automatic annotation manner with the same metric as KITTI MOTS.

3 Method

In this section, we first formulate how PointTrack converts different data modalities into a unified per-pixel style and learns context-aware instance embeddings M on 2D segments. Then, details about instance segmentation are introduced.

3.1 Context-aware instance embeddings extraction

For an instance C with its segment C_s and its smallest circumscribed rectangle C_b, we enlarge C_b to $\hat{C}_b$ by extending its border in all four directions (top, down, left, and right) by a scale factor k ($k = 0.2$ by default). Both C_s and $\hat{C}_b$ are visualized in dark green in the lower-left corner of Fig. 2. Then, we regard the foreground segment as a 2D point cloud and denote it as F. Similarly, we regard the other area in $\hat{C}_b$ the environment point cloud and denote it as E. Each point inside $\hat{C}_b$ has six dimensional data space $(u, v, R, G, B, \mathcal{C})$ that contains the coordinate (u, v) in the image plane, the pixel color (R, G, B), and which class $\mathcal{C}$ the pixel belongs to.

For the foreground point cloud F, we uniformly random sample N_F points ($N_F = 1000$ by default) for feature extraction. As shown in Fig. 2, N_F points are enough to evenly cover a relatively large instance. For the environment point cloud E, N_E points ($N_E = 500$ by default) are randomly selected. The coordinate of the foreground point F_i is denoted as (u_i^F, v_i^F) and the coordinate of the environment point E_i is denoted as (u_i^E, v_i^E). The center point $P(u_c^F, v_c^F)$ is computed by averaging the coordinates of selected foreground points $\{F_i | i = 1, ..., N_F\}$ in the image plane. P is highlighted in blue in the foreground point cloud (see Fig. 2).

Previous works [31, 36, 8] have demonstrated that features concerning position, appearance, scale, shape, and nearby objects, are useful for tracking.

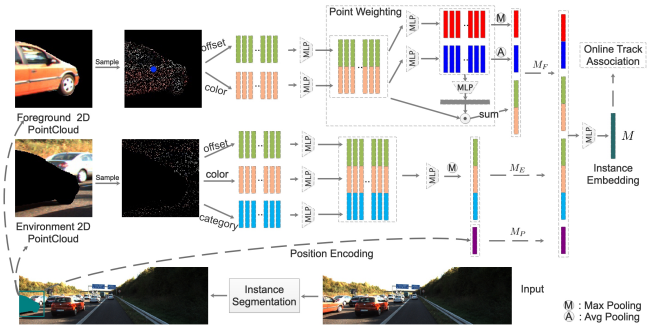

Fig. 2. Overview of PointTrack. For an input image, PointTrack obtains instance segments by an instance segmentation network. Then, PointTrack regards the segment and its surrounding environment as two 2D point clouds and learn features on them separately. MLP stands for multi-layer perceptron with Leaky ReLU.

Intuitively, PointTrack can summarize all the above features by learning the following data modalities: (i) Offset; (ii) Color; (iii) Category; (iv) Position. In the following, we formulate these data modalities and show how PointTrack learns context-aware embeddings from them.

Offset. We define the offset data of each foreground point F_i and each environment point E_i as follows:

$$O_{F_i} = (u_i^F - u_c^F, v_i^F - v_c^F), O_{E_i} = (u_i^E - u_c^F, v_i^E - v_c^F) \tag{1}$$

The offset data, which are formulated as vectors from the instance center P to themselves, represent the relative locations inside the segment. Offset vectors of foreground points provide essential information concerning both the scale and the shape of instances.

Color. We consider RGB channels and formulate the color data as follows:

$$C_{F_i} = (R_i^F, G_i^F, B_i^F), C_{E_i} = (R_i^E, G_i^E, B_i^E) \tag{2}$$

When the color data combine with the offset data, the discriminative appearance features can be learned from foreground points and the surrounding color distribution can be learned from environment points. The ablation study (see Table 6) shows that the color data are critical for accurate instance association.

Category. To further incorporate the environmental context into point-wise features, we encode all semantic class labels including the background class (suppose Z classes include the background) into fixed-length one-hot vectors $\{H_j | j = 1, ..., Z\}$. Then, for selected environment points E_i, the one-hot category vector are also gathered for feature extraction. Suppose that E_i belongs to the category $\mathcal{C}_i$, the category data are formulated as follows:

$$Y_{E_i} = H_{\mathcal{C}_i}, \mathcal{C}_i \in [1, Z] \tag{3}$$

Strong context features can be learned by PointTrack by jointly learning from the category data and the offset data. When the current instance is adjacent to other instances, for E_i lying on the nearby instances, the category data Y_{E_i} together with the offset data O_{E_i} tell PointTrack both the relative position and the semantic class of nearby instances, which serve as strong clues for instance association. Visualizations (see Fig. 5) also confirm that environment points on nearby instances matter in learning discriminative instance embeddings.

Position. Since previous three data modalities focus on extracting features around C_b regardless of the position of C_b in the image plane, we encode the position of C_b into the position embedding M_P. Following [33], we embed the position of C_b (4-dim) into a high-dimensional vector (64-dim) to make it easier for learning by computing *cosine* and *sine* functions of different wavelengths.

Based on the above four data modalities, PointTrack learns the foreground embeddings M_F and the environment embeddings M_E in separate branches. As shown in Fig. 2, the environment embeddings M_E are learned by first fusing (O_E, C_E, Y_E) for all E_i and then applying the max pooling operation to the fused features. As aforementioned, by fusing (O_E, Y_E), PointTrack learns strong context clues concerning nearby instances from M_E. For the foreground point cloud F, M_F is learned by fusing (O_F, C_F). Based on the intuition that more prominent points should have higher weights for differentiating instances, and other points should also be considered, but have lower weights, we introduce the point weighting layer to actively weight all foreground points and sum the features of all points. Different from Max-Pooling which only selects features of prominent points and Average-Pooling which blindly averages features of all points, the point weighting layer learns to summarize the foreground features by learning to weight points. Visualizations (see Fig. 5) demonstrate that the point weighting layer learns to give informative areas higher weights. Afterward, as shown in Fig. 2, M_F, M_E, and the position embeddings M_P are concatenated for predicting the final instance embeddings M as follows:

$$M = \mathbf{MLP}(M_F + M_E + M_P) \tag{4}$$

where + represents concatenation and **MLP** denotes multi-layer perceptron.

Instance association. To produce the final tracking result, we need to perform instance association based on similarities. Given segment C_{s_i} and segment C_{s_j}, and their embeddings M_i and M_j, the similarity S is formulated as follows:

$$S(C_{s_i}, C_{s_j}) = -D(M_i, M_j) + \alpha * U(C_{s_i}, C_{s_j}) \tag{5}$$

where D denotes the Euclidean distance and U represents the mask IOU. α is set to 0.5 by default. If an active track does not update for the recent β frames, we end this track automatically. For each frame, we compute the similarity between the latest embeddings of all active tracks and embeddings of all instances in the current frame according to Eq. (5). Following [34], we set a similarity threshold γ for instance association and instance association is allowed only when the similarity is greater than γ. The Hungarian algorithm [17] is exploited to perform

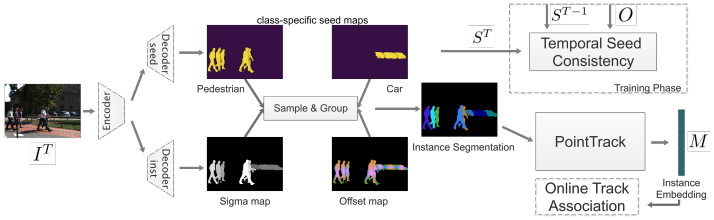

Fig. 3. Segmentation network of PointTrack.

instance matching. After instance association, unassigned segments will start new tracks. By default, β and γ are set to 30 and -8.0 respectively.

3.2 Instance segmentation with Temporal Seed Consistency

Different from previous methods [34, 27] which put great efforts to adapt mask RCNN into the MOTS frameworks, PointTrack builds on a state-of-the-art one-stage instance segmentation method named SpatialEmbedding [23]. SpatialEmbedding performs instance segmentation without bbox proposals, and thus runs much faster than two-stage methods. As shown in Fig. 3, SpatialEmbedding follows an encoder-decoder structure with two separate decoders: (i) the seed decoder; (ii) the inst decoder. Given an input image I^T at time T, the seed decoder predicts seed maps S^T for all semantic classes. Besides, the inst decoder predicts a sigma map denoting the pixel-wise cluster margin and an offset map representing the vector pointing to the corresponding instance center. Afterward, instance centers are sampled from S^T and pixels are grouped into segments according to the learned clustering margin for each instance. When applied to MOTS, by studying the segmentation failure cases, we find that the seed map predictions are not consistent between consecutive frames, which results in many false positives and false negatives. Therefore, we introduce the temporal consistency loss in the training phase to improve the quality of seed map prediction as follows. First, we also feed the input image I^{T-1} at time $T-1$ to SpatialEmbedding to predict the seed maps S^{T-1}. Then, optical flow O between I^{T-1} and I^T is estimated by VCN[3] [39]. Subsequently, we synthesize the warped seed maps $\hat{S^T} = O(S^{T-1})$ by exploiting O to warp S^{T-1}. Our temporal consistency loss is formulated as: $L_{tc} = \frac{1}{N}\sum_i^N ||\hat{S_i^T} - S_i^T||^2$. where N is the number of foreground pixels and i denotes the i-th foreground pixel. Evaluations (see Table 2) demonstrate that our temporal consistency loss improves the instance segmentation performance.

[3] We exploit the pre-trained model provided at *https://github.com/gengshan-y/VCN* for optical flow estimation.

	Frames	Tracks	Annotations	Car density	Crowd cars	Frames per second
APOLLO MOTS	11488	1530	64930	5.65	36403	10
KITTI MOTS (Car)	8008	582	26899	3.36	14509	7

Table 1. Comparison between APOLLO MOTS and KITTI MOTS on their respective train/validation sets.

4 Apollo MOTS Dataset

Tracking becomes more challenging with the increase of instances. However, for KITTI MOTS, the instance density is limited (only 3.36 cars per frame on average) and crowded scenes are also insufficient. Based on these observations, we present our Apollo MOTS dataset. We first briefly overview the dataset. Thereafter, the annotation procedures are introduced.

4.1 Overview

We build APOLLO MOTS on the public ApolloScape dataset [13] which contains video instance segment labels for 49,287 frames. As there are barriers on both sides of the road where data were collected, pedestrians are much fewer than cars in the ApolloScape dataset. Therefore, we focus on cars. As the ApolloScape dataset calibrates the camera for each frame, APOLLO MOTS can serve as a challenging MOTS dataset for both 2D tracking and 3D tracking.[4]

Detailed comparisons between our APOLLO MOTS and KITTI MOTS on their respective train/validation sets are shown in Table 1. APOLLO MOTS contains 22480 video frames including the testing set. We divide the train set, validation set, and test set according to the proportions of 30%, 20%, and 50%. Scenes in these three sets have similar tracking difficulties. The original image resolution in the ApolloScape dataset is 3384 (width) x 2710 (height). We crop it to 3384 x 1604 to remove the sky area and down-sample it to a lower and more suitable resolution 1692 x 802. As shown in Table 1, though APOLLO MOTS has a similar number of frames, we have two times more tracks and car annotations. We define the car density as the number of cars per frame. The average car density of APOLLO MOTS is 5.65, which is much higher than that of KITTI MOTS. Moreover, as tracking becomes more challenging when cars are overlapped, we count the number of crowded cars for both APOLLO MOTS and KITTI MOTS. A car is considered crowded if and only if its segment is adjacent to any other car. Our APOLLO MOTS has 2.5 times more crowded cars than KITTI MOTS.

4.2 Annotation

We annotate all video frames in the ApolloScape dataset. If consecutive frames contain no cars or are too easy for tracking, the entire video is removed. The

[4] Sample videos are provided in the supplementary material.

resulting 22480 frames represent the most difficult tracking scenes in the ApolloScape dataset. We annotate APOLLO MOTS in the following three steps.

(1) In-complete instance segment removal. For cars occluded by the fence, the ground-truth instance segment is always in-complete in the ApolloScape dataset. We manually traverse all frames to remove these in-complete instances by setting the bounding box area compassing this instance to the 'Dontcare' category. The 'Dontcare' area will be ignored in the evaluation process. After this step, only instances with complete segments will be preserved.

(2) Semi-automatic tracking annotation. We incorporate PointTrack trained on KITTI MOTS into our data annotation tool for automatic instance association. For each frame, the tracking results generated by PointTrack are manually reviewed and corrected. Moreover, we subjectively assign different difficulty levels to all videos ($0 \sim 4$, from the easiest to the hardest) by jointly considering the crowded level, the rotation of the camera, the overlap level, etc..

(3) Simple video removal and dataset partitioning. All videos with difficulty level 0 are discarded. For each difficulty level from 1 to 4, we divide videos of this level to the train, validation, and test sets according to the aforementioned percentages to ensure that these sets share similar difficulties.

5 Experiments

Experiments are divided into four parts[5]. Firstly, we evaluate PointTrack across three datasets: the KITTI MOTS dataset [9], the MOTSChallenge dataset [21], and our proposed Apollo MOTS dataset. Secondly, we show the ablation study on data modalities. Thirdly, to investigate what PointTrack learns from 2D point clouds, we visualize both predicted instance embeddings and critical tracking points. Lastly, we provide our results on the official KITTI MOTS test set.

Metric. Following previous works [12, 18], we focus on sMOTSA, MOTSA, and id switches (IDS). As an extension of MOTA, MOTSA measure segmentation as well as tracking accuracy. sMOTSA [34] is a soft version of MOTSA which weights the contribution of each true positive segment by its mask IoU with the corresponding ground truth segment.

Experimental Setup. Following previous works [34, 12], we pre-train the segmentation network on the KINS dataset [29] due to the limit of training data in KITTI MOTS (only 1704 frames contains Pedestrian where merely 1957 masks are manually annotated). Afterward, SpatialEmbedding is fine-tuned on KITTI MOTS with our proposed seed consistency loss for 50 epochs at a learning rate of $5 \cdot 10^{-6}$. For MOTSChallenge, we fine-tune the model trained on KITTI MOTS for 50 epochs at a learning rate of $5 \cdot 10^{-6}$. For APOLLO MOTS, we train SpatialEmbedding from scratch following [23]. Besides, our PointTrack is trained from scratch in all experiments by margin based hard triplet loss [40]. An instance database is constructed from the train set by extracting all crops $\hat{C}_b$ of all track ids. Unlike previous method [34] which samples T frames as a batch,

[5] More details and ablation studies are provided in the supplementary material.

Type	Method	Det. & Seg.	Speed	Cars			Pedestrians		
				sMOTSA	MOTSA	IDS	sMOTSA	MOTSA	IDS
2D	TRCNN [34]	TRCNN	0.5	76.2	87.8	93	46.8	65.1	78
3D	BePix [31]	RRC[30]+TRCNN	3.96	76.9	89.7	88	-	-	-
2D	MOTSNet [27]	MOTSNet	-	78.1	87.2	-	54.6	69.3	-
3D	MOTSFusion [18]	TRCNN+BS	0.84	82.6	90.2	51	58.9	71.9	36
3D	BePix	RRC+BS	3.96	84.9	93.8	97	-	-	-
3D	MOTSFusion	RRC+BS	4.04	**85.5**	94.6	35	-	-	-
2D	PointTrack	PointTrack	**0.045**	**85.5**	**94.9**	**22**	**62.4**	**77.3**	**19**
2D	PointTrack (without TC)	PointTrack	**0.045**	82.9	92.7	25	61.4	76.8	21
2D	PointTrack (on Bbox)	PointTrack	**0.045**	85.3	94.8	36	61.8	76.8	36

Table 2. Results on the KITTI MOTS validation. Speed is measured in seconds per frame. TC denotes the temporal consistency loss. BS represents BB2SegNet [19].

Fig. 4. Quantitative results on KITTI MOTS. Instances of the same track id are plotted in the same color.

we sample D track ids as a batch, each with three crops. These three crops are selected from three equally spaced frames rather than three consecutive frames to increase the intra-track-id discrepancy. The space between frames is randomly chosen between 1 and 10. Empirically we find a smaller D ($16 \sim 24$) is better for training PointTrack, because a large D (more than 40) leads to a quick overfitting. In addition, to test the generalization ability of instance association, we test PointTrack*, whose instance embeddings extraction is only fine-tuned on KITTI MOTS, on both MOTSChallenge and Apollo MOTS.

We compare recent works on MOTS: TRCNN [34], MOTSNet [27], BePix [31], and MOTSFusion (online) [18]. TRCNN and MOTSNet perform 2D tracking while BePix and MOTSFusion track on 3D.

Results on KITTI MOTS. Following MOTSFusion, we compare different methods on different segmentation results. The main results are summarized in Table 2, where our method outperforms all the state-of-the-art methods, especially for pedestrians. Quantitative results are shown in Fig. 4. On the 'Speed' column, we show the total time of detection, segmentation, and tracking[6]. On KITTI MOTS, our PointTrack takes 0.037s per frame for instance segmentation, 8ms per frame for tracking, and 3ms per instance for embedding extraction.

For cars, the 3D tracking method MOTSFusion adopts a time-consuming detector RRC [30] which takes 3.6s per frame to perform detection. MOTSFusion builds up short tracklets using 2D optical flow and segments. Afterward, 3D world-space motion consistency is used to merge tracklets together into ac-

[6] Our calculated speed is different from MOTSFusion because in [18], the detection time of the RRC detector which takes 3.6s per frame is ignored. The speed of MOTSNet [27] is not mentioned in their work.

	Seg.	sMOTSA	MOTSA
DeepSort [35]	TRCNN	45.71	57.06
TRCNN [34]	TRCNN	49.84	61.19
DeepSort	PointTrack	64.69	73.97
PointTrack*	PointTrack	70.58	79.87
PointTrack	PointTrack	**70.76**	**80.05**

Table 3. Results on APOLLO MOTS validation.

Dataset	Seg.	Method	IDS (car)	IDS (Ped.)
KITTI MOTS Val	TRCNN	TRCNN	93	78
		PointTrack	**46**	**30**
	RRC+BS	BePix	97	-
		MOTSFusion	35	-
		PointTrack	**14**	-
APOLLO MOTS	TRCNN	DeepSort	1263	-
		TRCNN	312	-
		PointTrack	**241**	-
	PointTrack	DeepSort	1692	-
		PointTrack	**292**	-
KITTI MOTS Test	MOTSFusion	MOTSFusion	201	279
		PointTrack	**187**	**150**

Table 4. Comparisons of IDS on KITTI MOTS and APOLLO MOTS.

curate long-term tracks while recovering missed detections. By contrast, though tracking objects purely on 2D images with a light-weight instance segmentation network, PointTrack achieves comparable performance to the 3D tracking method MOTSFusion (0.3% gains on MOTSA) with significant speed improvement (0.045s VS. 4.04s). For pedestrians, PointTrack surpasses current approaches by 3.5% and 5.4% on sMOTSA and MOTSA respectively. It is worth noting that, though only small improvements over MOTSFusion are observed for cars on the KITTI MOTS validation set, PointTrack surpasses MOTSFusion by large margins on the official test set (see Table 8), which demonstrates the good generalization ability. Besides, when the temporal consistency loss is removed (see the last but one row in Table 2), the performance drops are observable (by 2.6% and 0.5% on sMOTSA for cars and pedestrians, respectively). This demonstrates the effectiveness of our temporal consistency loss.

The effectiveness of segment C_s. To investigate the effectiveness of segment C_s, we ignore C_s and instead sample points inside the inmodal bbox C_b. $N_E + N_F$ points are randomly sampled and the network branch for environment embedding is removed. As shown in the last row in Table 2, for cars, IDS increase by 64% (from 22 to 36) after the segments are removed. For pedestrians, more significant performance drops (89.5% IDS increase and 0.6% sMOTSA) are observed. The increase in IDS demonstrates that segment matters for better tracking performances. Moreover, the gap between the performance drop in cars and in pedestrians demonstrates that segments are more effective in improving the tracking performance for non-rigid objects where bbox level feature extraction introduces more ambiguities.

Results on APOLLO MOTS. We show comparisons on APOLLO MOTS in Table 3. All models are trained under the same setting as KITTI MOTS. We

	sMOTSA	MOTSA
MOTDT [5]+ MG	47.8	61.1
MHT-DAM [16]+ MG	48.0	62.7
jCC [15]+ MG	48.3	63.0
FWT [11]+ MG	49.3	64.0
TrackRCNN [34]	52.7	66.9
MOTSNet [27]	56.8	69.4
PointTrack*	57.98	70.47
PointTrack	**58.09**	**70.58**

Table 5. Results on MOTSChallenge. +MG denotes mask generation with a domain fine-tuned Mask R-CNN.

also train DeepSort [35] to tracks instances on inmodal bboxes surrounding segments as a baseline to PointTrack. DeepSort extends SORT [3] by incorporating convolution layers to extract appearance features for instance association. Different from DeepSort, our PointTrack extracts features from 2D point clouds rather than images. When applied to the same segmentation results (see the third row and the fifth row), PointTrack achieves 6% higher sMOTSA than DeepSort and reduces IDS from 1692 to 292. Also, compared with the performance on KITTI MOTS, the sMOTSA of TRCNN and PointTrack decreases by 14.7% and 26.4% when evaluated on APOLLO MOTS. As the training and testing settings are the same, the significant performance drop shows that APOLLO MOTS is more challenging than KITTI MOTS.

The significant IDS reduction by PointTrack. As shown in Table 4, when applied to the same segmentation results on different datasets, PointTrack can effectively reduce IDS. The steady IDS reduction across different datasets and different segmentation results demonstrate the effectiveness of PointTrack.

Results on MOTSChallenge. Compared with KITTI MOTS, MOTSChallenge has more crowded scenarios and more different viewpoints. Following previous work [34, 27], we train PointTrack in a leaving-one-out fashion and show comparisons on MOTSChallenge in Table 5. Our PointTrack outperforms the state-of-the-art methods by more than 1.1% on all three metrics. It's worth noting that, though the instance embeddings extraction is only fine-tuned on KITTI MOTS (see PointTrack* in Table 3,5), PointTrack* also achieves similar high performance on both APOLLO MOTS and MOTSChallenge, demonstrating the good generalization ability on instance embedding extraction.

Ablation Study on the impact of data modalities. We remove four data modalities in turn to examine their impacts on performance. As shown in Table 6, the largest performance drop occurs when the color data are removed. By contrast, the performance drop is minimal when the position data are removed. This difference in performance gap demonstrates that our PointTrack focuses more on the appearance features and the environment features while relies less on the bounding box position to associate instances, leading to higher tracking performances and much lower IDS than previous approaches.

Impact of the point weighting layer, M_F, M_E, and the mask IOU. We remove the point weighting layer (P_W), the foreground embeddings M_F,

				Cars			Pedestrians		
Color	Offset	Category	Position	sMOTSA	MOTSA	IDS	sMOTSA	MOTSA	IDS
√	√	√	√	**85.51**	**94.93**	**22**	**62.37**	**77.35**	**19**
x				83.65	93.08	171	61.15	76.13	60
	x			85.32	94.74	37	62.16	77.14	26
		x		85.33	94.40	38	62.13	77.11	27
			x	85.35	94.77	35	62.31	77.29	21

Table 6. Ablation study on the impact of different data modalities.

				Cars		Pedestrians	
P_W	M_F	M_E	M_I	sMOTSA	MOTSA	sMOTSA	MOTSA
√	√	√	√	**85.51**	**94.93**	**62.37**	**77.35**
x				85.37	94.79	62.04	77.02
	x			83.59	93.01	61.27	76.25
		x		85.30	94.72	61.98	76.96
			x	85.33	94.76	62.31	77.29

Table 7. Experiments on impact of the point weighting layer, M_F, M_E, and the mask IOU.

the environment embeddings M_E, and the mask IOU (M_I) in turn to examine their impacts on performance. When we remove the mask IOU, we set α to zero in Eq. (5). As shown in Table 7, when the foreground embeddings M_F is removed, the performance drops a lot, demonstrating that the foreground point cloud in the segment area matters most in the instance association. By contrast, when the mask IOU is removed, the performance drop is minimal, especially for Pedestrians. Therefore, for instances with rigid shapes, considering the mask IOU in computing similarity is more beneficial than instances with non-rigid shapes like Pedestrians.

Visualizing critical points. We visualize critical foreground points as well as critical environment points in Fig. 5. For each instance, to validate the temporal consistency of critical points, we select crops from three consecutive frames.

For foreground points, points with 10% top weights predicted by the point weighting layer are plotted in red. As shown in Fig. 5, critical foreground points gather around car glasses and around car lights. We believe that the offsets of these points are essential for learning the shape and the pose of the vehicle. Also, their colors are important to outline the instance appearance and light distribution. Moreover, we find that PointTrack keeps the consistency of weighting points in consecutive frames even when different parts are occluded (the second and the fifth in the first row), or the car is moving to the image boundary (the fourth in the first row). The consistency in point weighting across frames shows the effectiveness of our point weighting layer.

For environment points, we visualize the five most critical points in yellow. These points are selected by first fetching the tensor with size of $256 * N_E$ before the max-pooling layer in the environment branch and then gathering the index with the max value for all 256-dimensions. Among these 256 indexes, points belonging to the five most common indexes are selected. As shown in Fig. 5, when instances are adjacent to any other instances, yellow points usually gather on nearby instances. As aforementioned, when combining the category data with the offset data, strong context clues are provided from environment points for

Fig. 5. Visualizations of critical points. Red points and yellow points represent the critical foreground points and the critical environment points respectively.

	Cars		Pedestrians	
	sMOTSA	MOTSA	sMOTSA	MOTSA
TRCNN	67.00	79.60	47.30	66.10
MOTSNet	71.00	81.70	48.70	62.00
MOTSFusion	75.00	84.10	58.70	72.90
PointTrack	**78.50**	**90.90**	**61.50**	**76.50**

Table 8. Results on KITTI MOTS test set.

instance association. The distribution of critical environment points validate that PointTrack learns discriminative context features from environment points.

Results on KITTI MOTS Testset. To further demonstrate the effectiveness of PointTrack, we report the evaluation results on the official KITTI test set in Table 8 where our PointTrack currently ranks first. It is worth noting that, on MOTSA, PointTrack surpasses MOTSFusion by 6.8% for cars and 3.6% for pedestrians. Also, PointTrack is the most efficient framework among current approaches. More detailed comparisons can be found online[7].

6 Conclusions

In this paper, we presented a new tracking-by-points paradigm together with an efficient online MOTS framework named PointTrack, by breaking the compact image representation into 2D un-ordered point clouds for learning discriminative instance embeddings. Different informative data modalities are converted into point-level representations to enrich point cloud features. Evaluations across three datasets demonstrate that PointTrack surpasses all the state-of-the-art methods by large margins. Moreover, we built APOLLO MOTS, a more challenging MOTS dataset over KITTI MOTS with more crowded scenes.

Acknowledgement

This work was supported by the Anhui Initiative in Quantum Information Technologies (No. AHY150300).

[7] The official leader-board: *http://www.cvlibs.net/datasets/kitti/eval_mots.php*

References

1. Baser, E., Balasubramanian, V., Bhattacharyya, P., Czarnecki, K.: Fantrack: 3d multi-object tracking with feature association network. In: 2019 IEEE Intelligent Vehicles Symposium (IV). pp. 1426–1433. IEEE (2019)
2. Bergmann, P., Meinhardt, T., Leal-Taixe, L.: Tracking without bells and whistles. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 941–951 (2019)
3. Bewley, A., Ge, Z., Ott, L., Ramos, F., Upcroft, B.: Simple online and realtime tracking. In: 2016 IEEE International Conference on Image Processing (ICIP). pp. 3464–3468. IEEE (2016)
4. Bhat, G., Danelljan, M., Gool, L.V., Timofte, R.: Learning discriminative model prediction for tracking. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 6182–6191 (2019)
5. Chen, L., Ai, H., Zhuang, Z., Shang, C.: Real-time multiple people tracking with deeply learned candidate selection and person re-identification. In: 2018 IEEE International Conference on Multimedia and Expo (ICME). pp. 1–6. IEEE (2018)
6. Chu, P., Ling, H.: Famnet: Joint learning of feature, affinity and multi-dimensional assignment for online multiple object tracking. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 6172–6181 (2019)
7. Danelljan, M., Bhat, G., Khan, F.S., Felsberg, M.: Atom: Accurate tracking by overlap maximization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4660–4669 (2019)
8. Geiger, A., Lauer, M., Wojek, C., Stiller, C., Urtasun, R.: 3d traffic scene understanding from movable platforms. IEEE transactions on pattern analysis and machine intelligence **36**(5), 1012–1025 (2013)
9. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR) (2012)
10. Held, D., Levinson, J., Thrun, S.: Precision tracking with sparse 3d and dense color 2d data. In: 2013 IEEE International Conference on Robotics and Automation. pp. 1138–1145. IEEE (2013)
11. Henschel, R., Leal-Taixé, L., Cremers, D., Rosenhahn, B.: Fusion of head and full-body detectors for multi-object tracking. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 1428–1437 (2018)
12. Hu, A., Kendall, A., Cipolla, R.: Learning a spatio-temporal embedding for video instance segmentation. arXiv preprint arXiv:1912.08969 (2019)
13. Huang, X., Cheng, X., Geng, Q., Cao, B., Zhou, D., Wang, P., Lin, Y., Yang, R.: The apolloscape dataset for autonomous driving. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops. pp. 954–960 (2018)
14. Karunasekera, H., Wang, H., Zhang, H.: Multiple object tracking with attention to appearance, structure, motion and size. IEEE Access **7**, 104423–104434 (2019)
15. Keuper, M., Tang, S., Andres, B., Brox, T., Schiele, B.: Motion segmentation & multiple object tracking by correlation co-clustering. IEEE transactions on pattern analysis and machine intelligence **42**(1), 140–153 (2018)
16. Kim, C., Li, F., Ciptadi, A., Rehg, J.M.: Multiple hypothesis tracking revisited. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 4696–4704 (2015)

17. Kuhn, H.W.: The hungarian method for the assignment problem. Naval research logistics quarterly **2**(1-2), 83–97 (1955)
18. Luiten, J., Fischer, T., Leibe, B.: Track to reconstruct and reconstruct to track. IEEE Robotics and Automation Letters (2020)
19. Luiten, J., Voigtlaender, P., Leibe, B.: Premvos: Proposal-generation, refinement and merging for video object segmentation. In: Asian Conference on Computer Vision. pp. 565–580. Springer (2018)
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. Milan, A., Leal-Taixé, L., Reid, I., Roth, S., Schindler, K.: MOT16: A benchmark for multi-object tracking. arXiv:1603.00831 [cs] (Mar 2016), http://arxiv.org/abs/1603.00831, arXiv: 1603.00831
22. Mitzel, D., Leibe, B.: Taking mobile multi-object tracking to the next level: People, unknown objects, and carried items. In: European Conference on Computer Vision. pp. 566–579. Springer (2012)
23. Neven, D., Brabandere, B.D., Proesmans, M., Gool, L.V.: Instance segmentation by jointly optimizing spatial embeddings and clustering bandwidth. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (June 2019)
24. Osep, A., Mehner, W., Mathias, M., Leibe, B.: Combined image-and world-space tracking in traffic scenes. In: 2017 IEEE International Conference on Robotics and Automation (ICRA). pp. 1988–1995. IEEE (2017)
25. Ošep, A., Mehner, W., Voigtlaender, P., Leibe, B.: Track, then decide: Category-agnostic vision-based multi-object tracking. In: 2018 IEEE International Conference on Robotics and Automation (ICRA). pp. 1–8. IEEE (2018)
26. Payer, C., Štern, D., Neff, T., Bischof, H., Urschler, M.: Instance segmentation and tracking with cosine embeddings and recurrent hourglass networks. In: International Conference on Medical Image Computing and Computer-Assisted Intervention. pp. 3–11. Springer (2018)
27. Porzi, L., Hofinger, M., Ruiz, I., Serrat, J., Bulò, S.R., Kontschieder, P.: Learning multi-object tracking and segmentation from automatic annotations. arXiv preprint arXiv:1912.02096 (2019)
28. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 652–660 (2017)
29. Qi, L., Jiang, L., Liu, S., Shen, X., Jia, J.: Amodal instance segmentation with kins dataset. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3014–3023 (2019)
30. Ren, J., Chen, X., Liu, J., Sun, W., Pang, J., Yan, Q., Tai, Y.W., Xu, L.: Accurate single stage detector using recurrent rolling convolution. In: CVPR (2017)
31. Sharma, S., Ansari, J.A., Murthy, J.K., Krishna, K.M.: Beyond pixels: Leveraging geometry and shape cues for online multi-object tracking. In: 2018 IEEE International Conference on Robotics and Automation (ICRA). pp. 3508–3515. IEEE (2018)
32. Tian, W., Lauer, M., Chen, L.: Online multi-object tracking using joint domain information in traffic scenarios. IEEE Transactions on Intelligent Transportation Systems (2019)
33. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017)

34. Voigtlaender, P., Krause, M., Osep, A., Luiten, J., Sekar, B.B.G., Geiger, A., Leibe, B.: Mots: Multi-object tracking and segmentation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7942–7951 (2019)
35. Wojke, N., Bewley, A., Paulus, D.: Simple online and realtime tracking with a deep association metric. In: 2017 IEEE international conference on image processing (ICIP). pp. 3645–3649. IEEE (2017)
36. Xu, J., Cao, Y., Zhang, Z., Hu, H.: Spatial-temporal relation networks for multi-object tracking. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3988–3998 (2019)
37. Xu, Z., Yang, W., Meng, A., Lu, N., Huang, H., Ying, C., Huang, L.: Towards end-to-end license plate detection and recognition: A large dataset and baseline. In: Proceedings of the European conference on computer vision (ECCV). pp. 255–271 (2018)
38. Xu, Z., Zhang, W., Ye, X., Tan, X., Yang, W., Wen, S., Ding, E., Meng, A., Huang, L.: Zoomnet: Part-aware adaptive zooming neural network for 3d object detection. In: AAAI. pp. 12557–12564 (2020)
39. Yang, G., Ramanan, D.: Volumetric correspondence networks for optical flow. In: Advances in Neural Information Processing Systems. pp. 793–803 (2019)
40. Yuan, Y., Chen, W., Yang, Y., Wang, Z.: In defense of the triplet loss again: Learning robust person re-identification with fast approximated triplet loss and label distillation. arXiv preprint arXiv:1912.07863 (2019)
41. Zhang, W., Zhou, H., Sun, S., Wang, Z., Shi, J., Loy, C.C.: Robust multi-modality multi-object tracking. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2365–2374 (2019)