Pattern Recognition Letters 178 (2024) 21–27

Contents lists available at ScienceDirect

Pattern Recognition Letters

journal homepage: www.elsevier.com/locate/patrec

LiDAR video object segmentation with dynamic kernel refinement

Jianbiao Mei, Yu Yang, Mengmeng Wang, Zizhang Li, Jongwon Ra, Yong Liu *

Institute of Cyber-Systems and Control, Zhejiang University, Hangzhou 310027, China

ARTICLE INFO

Editor: Wei Zhang

Keywords:
LiDAR segmentation
Video object segmentation
Dynamic kernel

ABSTRACT

In this paper, we formalize memory- and tracking-based methods to perform the LiDAR-based Video Object Segmentation (VOS) task, which segments points of the specific 3D target (given in the first frame) in a LiDAR sequence. LiDAR-based VOS can directly provide target-aware geometric information for practical application scenarios like behavior analysis and anticipating danger. We first construct a LiDAR-based VOS dataset named KITTI-VOS based on SemanticKITTI, which acts as a testbed and facilitates comprehensive evaluations of algorithm performance. Next, we provide two types of baselines, i.e., memory-based and tracking-based baselines, to explore this task. Specifically, the first memory-based pipeline is built on a space–time memory network equipped with the non-local spatiotemporal attention-based memory bank. We further design a more potent variant to introduce the locality into the spatiotemporal attention module by local self-attention and cross-attention modules. For the second tracking-based baseline, we modify two representative 3D object tracking methods to adapt to LiDAR-based VOS tasks. Finally, we propose a refine module that takes mask priors and generates object-aware kernels, which could boost all the baselines' performance. We evaluate the proposed methods on the dataset and demonstrate their effectiveness.

1. Introduction

Recently, the task of RGB-based Video Object Segmentation (VOS) has attracted extensive attention [1–10] attributed to its widespread applications in object tracking and behavior analysis. This task aims to segment objects given in the first frame in successive video snippets. Despite promising results, they face great challenges when dealing with low-light conditions or textureless objects. In contrast, LiDARs are insensitive to texture and robust to light variations, making them a suitable complement to cameras. On the other hand, with the rapid development of LiDAR sensors in the past decade, solving various vision problems with point clouds has become a hot topic due to the huge potential in applications such as autonomous driving and robotics. However, we recognize that the task of LiDAR-based VOS has been unexplored, mainly due to the lack of proper task formalization, datasets, and evaluation benchmarks.

In this paper, we formalize the LiDAR-based VOS task that segments the specific targets (provided in the first frame) in LiDAR sequences, providing their motion and geometric information for practical applications such as behavior analysis and anticipating danger. 3D object tracking is the most related task, which provides objects' boundary and orientation information, much coarser than point-wise segmentation. One can easily obtain the target's shape proposal (segmentation mask) by cropping points inside the bounding box provided by the tracker. However, generating high-quality bounding boxes is challenging since LiDAR points only exist on object surfaces. Moreover, due to the inaccuracy of bounding boxes and the complex scenes, the cropped points usually contain many noisy points or only part of objects, resulting in low-quality segmentation, especially on large objects, which is consistent with our experiments in Table 2.

To begin with, we construct a LiDAR-based VOS dataset named KITTI-VOS based on SemanticKITTI [11] to lay the data foundation for this task, facilitating comprehensive evaluations of algorithm performance. SemanticKITTI is based on the KITTI Vision Benchmark, providing panoptic annotations for all sequences of the Odometry Benchmark. We choose traffic participants (cars, people, trucks, and cyclists) as targets, whose annotations are generated by selecting the corresponding instance mask from the panoptic annotations. We set some constrained rules to form LiDAR sequences with reliable initial frames. For example, we remove very far instances and choose frames that contain instances with more than fifty points as the first frame and the subsequent frames to form the LiDAR sequences.

Next, based on the constructed KITTI-VOS, we provide two types of baselines for comprehensive evaluation. Firstly, inspired by 2D memory-based VOS techniques, we propose a simple and flexible memory-based method, termed 3DSTM, for the LiDAR-based VOS task. As shown in Fig. 1(a), each module of the baseline is very concise

* Corresponding author.
E-mail addresses: jianbiaomei@zju.edu.cn (J. Mei), yu.yang@zju.edu.cn (Y. Yang), mengmengwang@zju.edu.cn (M. Wang), zzli@zju.edu.cn (Z. Li), jongwonra@zju.edu.cn (J. Ra), yongliu@iipc.zju.edu.cn (Y. Liu).

https://doi.org/10.1016/j.patrec.2023.12.013
Received 30 May 2023; Received in revised form 21 September 2023; Accepted 22 December 2023
Available online 27 December 2023
0167-8655/© 2023 Elsevier B.V. All rights reserved.

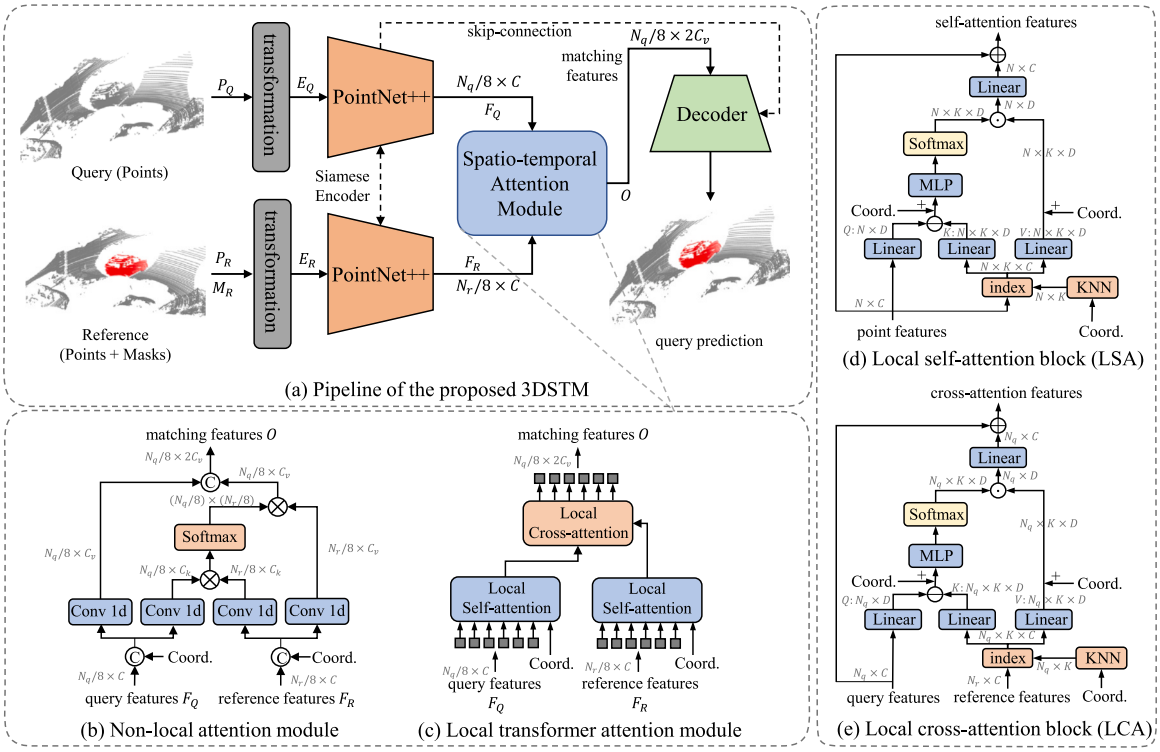

Fig. 1. Overview of memory-based pipeline, which mainly consists of four components, i.e., input transformation modules, Siamese encoder, spatiotemporal attention module, and decoder. The two input transformation modules are used to map the reference points P_R with targets' masks M_R and query points P_Q to the embeddings E_R and E_Q. And the Siamese encoder encodes the input embeddings to features F_R and F_Q. The spatiotemporal models the relationships between F_R and F_Q and outputs the matching features O. Finally, the decoder takes the matching features as input and predicts the segmentation masks of search points. (d) and (e) illustrate the details of the local self-attention block (LSA) and the local cross-attention block (LCA). $\ominus, \oplus$ denote element-wise reduction and element-wise summation. $\otimes, '$ indicate matrix multiplication and concatenate operation. $\odot$ means element-wise production and summation along the dimension of K.

to enable further exploration and improvement. Furthermore, the spatiotemporal attention module in 3DSTM is inherently similar to the non-local matching mechanism, lacking locality. However, the surface-aggregated point cloud shows the property of local density, reflecting the surface structure of objects. Besides, it is challenging to overcome under-segmentation problems with global attention only. Therefore, we replace the spatiotemporal attention block in the baseline with the local-attention transformers as a potent variant dubbed 3DSTM-TR to fully leverage the sequential nature and the local denseness structure of LiDAR videos. Moreover, to provide a more comprehensive evaluation, we added a simple post-processing operation after a 3D object tracker to form the tracking-based baselines.

Finally, we propose a refine module to boost all the baselines' performance. The refine module is plug-and-play, which can be easily extended to these baselines and trained end-to-end. It takes the coarse masks as priors, learning object-aware kernels and mask features to acquire high-quality segmentation masks. For memory-based baselines, the refine module takes the predicted coarse segmentation masks as priors to further improve the masks' qualities. And for tracking-based baselines, the refine module serves as a segmentation head to obtain more accurate predictions with generated coarse masks as priors from the predicted boxes. Thus, by taking advantage of prior knowledge, the refine module can improve the ability to handle distraction problems for memory-based baselines and break the limitation of the trackers.

Our main contributions can be summarized as follows:

- To our knowledge, we first perform the LiDAR-based VOS, which segments specific **3D** targets (given in the first frame) in LiDAR sequences.
- The 3D VOS dataset dubbed KITTI-VOS is constructed based on SemanticKITTI to facilitate comprehensive evaluations of algorithms.
- We provide two types of baselines, i.e., memory-based and tracking-based baselines to perform LiDAR-based video object segmentation, facilitating a more comprehensive study for this task.
- We design an effective refine module to explore the potential solutions to boost performances of these baselines.

2. Related works

2D video object segmentation. Visual segmentation [12–14] is a fundamental problem in computer vision and has widespread real-world applications such as robotics, video editing, and autonomous driving. As one of the popular segmentation tasks, 2D semi-supervised VOS provides the targets' masks in the first frame, and the algorithms should predict the segmentation masks for those targets in the subsequent frames. The existing algorithms can be categorized as tracking-based and matching-based methods. Tracking-based methods [1,3,4] combine an object tracker to indicate the spatial locations of the interesting objects, then segment masks in the detected bounding boxes. The integration of the object tracker helps improve the inference speed, which is more friendly to real-time applications such as edge intelligence [14,15]. However, the segmentation accuracy would be limited by the tracker's performance. The matching-based methods [5–10,16–19] match the features of the query frame and the reference frame to learn the appearances of the target objects. One of the most representative approaches is STM [7], which introduces a memory bank to store the past frames' features and uses a space–time attention module to retrieve the information in the memory for segmentation. Here, we follow the matching-based paradigms to construct the memory-based baseline 3DSTM and a variant 3DSTM-TR that introduces local correlation [20,21] for robust LiDAR-based VOS. To facilitate a more

comprehensive evaluation, we also integrate the tracker to form the tracking-based baselines.

3D singe object tracking. 3D single object tracking (SOT) aims at tracking a single reference object in consecutive LiDAR scans to provide its boundary and orientation information. Most methods [21–27] adopt a match-and-vote paradigm, which extracts features via a Siamese network and locates the target in the search area using appearance matching. P2B [23] proposed the target-specific feature augmentation to integrate the template information and exploited VoteNet [28] to predict the bounding box. MLVSNet [25] utilized multi-scale features for relation modeling and target localization. Recently, M2Track [29] introduced a motion-centric paradigm to handle 3D SOT from a new perspective. Unlike 3DSOT, our LiDAR-based VOS implies target-aware geometric understanding and can directly provide point-wise segmentation masks for practical robotic application scenarios such as behavior analysis and anticipating danger.

3. Methods

3.1. Problem definition

LiDAR-based Video Object Segmentation (VOS) can be formulated as follows: given a LiDAR sequence of length T and the initial target indicator (e.g. segmentation mask M_0) of the first frame P_0, the goal is to predict the foreground masks of the target in all subsequent frames. Specifically, at every timestamp t, we aim to obtain the target's mask M_t of the query frame P_t according to reference frames, i.e., historical frames and the corresponding predicted masks $\{(P_i, M_i)|i \in S, S \subseteq [0, t-1], 1 \leq t \leq T\}$.

3.2. LiDAR-based VOS dataset

We construct the KITTI-VOS dataset based on the large-scale outdoor dataset SemanticKITTI [11] to train models and facilitate comprehensive evaluations of algorithm performance. Since the panoptic annotations in SemanticKITTI provide the point-wise semantic label and instance ID, extracting objects from the raw point cloud scans is feasible. In the 3D VOS task, the initial frame of each LiDAR sequence is crucial for the subsequent tracking and segmentation. To provide a robust initialization, we remove instances containing few points, which are usually far away from the LiDAR sensor and unsuitable as initial targets due to possible noise. Following DSNet [30], we choose the instance with more than fifty points as the valid initial target and combine it with the subsequent frames to form the sequences. Besides, to provide more temporal context, we discard the sequences with a short temporal extent. Specifically, sequences with fewer than fifty frames (about 5 s) are empirically removed. We generate sequences of different categories separately to alleviate the class imbalance issue.

We chose traffic participants such as cars, persons, trucks, and cyclists as the segmentation targets to form LiDAR sequences. KITTI-VOS has 18370 LiDAR frames, containing 200 LiDAR sequences, 125 videos for cars, 39 videos for persons, 18 videos for trucks, and 18 for cyclists. And each video contains more than 50 frames. Table 1 shows the data distribution in detail. Note that the sequences of the validation split and train split are chosen from different scenes in the SemanticKITTI.

The segmentation quality is evaluated by measuring the overlap as the IoU between the predicted segmentation mask and its ground truth. We calculate the average IoU of all video sequences for each category as the final evaluation result.

Table 1
Data distribution of KITTI-VOS dataset. Here lists the number of frames/videos for each category.

	Car	Person	Truck	Cyclist
Train split	7684/75	1366/25	906/10	994/10
Valid split	5109/50	804/14	750/8	757/8
Total	12793/125	2170/39	1656/18	1751/18

3.3. Memory network for LiDAR-based VOS

Recently, matching-based methods [5–10,17] have achieved great success in 2D VOS. Most of them introduce a memory bank to store the past frames' features and use an attention-based matching method to retrieve the information in the memory for segmentation. Inspired by those approaches, we adopt the paradigm of memory networks to design the memory-based baseline, termed 3DSTM, which is flexible to exploit the spatiotemporal information of historical frames and concise to enable further exploration. 3DSTM consists of four components, i.e., input transformation modules, Siamese PointNet++ encoder, spatiotemporal attention module, and PointNet++ decoder, as shown in Fig. 1(a). To unify the inputs, 3DSTM adopts input transformation modules to map two types of inputs, i.e., reference points with targets' masks and query points, into the embedding space for the subsequent hierarchical feature extraction. Based on the unified input representations, the abundant object features are extracted by the weight-share Siamese encoder, which is compact and reduces the model complexity. Afterward, we design the attention-based spatiotemporal module to exploit the spatiotemporal cues in the extracted features better, which helps alleviate the appearance changes and occlusion in the LiDAR sequences. Finally, the decoder is attached after the spatiotemporal module to upsample the feature maps and obtain the final prediction. By this means, 3DSTM achieves a flexible and concise design.

Input transformation modules. We first design the transformation modules to encode points and corresponding masks into the embedding space. As illustrated in Fig. 1(a), the reference/query transformation modules (in gray color) have the same structure, which consists of 3 layers of a fully-connected network followed by ReLU layers [31] and BN layers [32] with filter size [64, 128, D]. The only difference between the two modules is their inputs. The query transformation only takes the query points $P_Q \in \mathbb{R}^{N_q \times 3}$ of the query frame as input and outputs features $E_Q \in \mathbb{R}^{N_q \times D}$. While we concatenate the reference points $P_R \in \mathbb{R}^{N_r \times 3}$ and corresponding object masks $M_R \in \mathbb{R}^{N_r \times 1}$ along the channel dimension to form the input of the reference transformation module, and it outputs the features $E_R \in \mathbb{R}^{N_r \times D}$.

Siamese encoder. The point-wise MLP in the transformation module has a weak ability to integrate local structural information, while the feature matching procedure in the following attention module may cause a heavy computational burden due to large amounts of points. Therefore, we construct our siamese encoder using the PointNet++ [33] encoder, which utilizes multi-scale neighborhoods to achieve both robustness and detail capture. The encoder processes a set of points sampled in a metric space in a hierarchical fashion to extract features from E_Q and E_R. The first three stages are used to aggregate point sets and reduce the number of input points by eight times and output the encoded reference features $F_R \in \mathbb{R}^{(N_r//8) \times C}$ and query features $F_Q \in \mathbb{R}^{(N_q//8) \times C}$.

Spatiotemporal attention module. To extract supportive cues from the reference features to the query features, we design two types of spatiotemporal attention modules, i.e., non-local attention module (in Fig. 1(b)) and local transformer attention module (in Fig. 1(c)).

The non-local attention module firstly uses two parallel convolutions to map features F_R/F_Q concatenated with point coordinates to the pairs of key and value maps $\{(k^R, v^R), (k^Q, v^Q)\}$. Then the similarities between the query key map k^Q and the reference key map k^R are computed by matrix multiplication and $softmax$ normalization. Next, we perform dot-product with the reference value v^R to extract informative features, and the results are concatenated with the query

value v^Q as the final matching results $O \in \mathbb{R}^{(N_q//8)\times(2\cdot C_v)}$. The attention module can be expressed as:

$$F'_* = [F_*, \hat{P}_*], k^* = \Phi_k(F'_*), v^* = \Phi_v(F'_*) \tag{1}$$

$$O(p) = [v^Q(p), \sum_{\forall q} \sigma(k^Q(p), k^R(q)) \cdot v^R(q)] \tag{2}$$

where "*" denotes "R" (reference) or "Q" (query), $\hat{P}_Q$ and $\hat{P}_R$ are Cartesian coordinates of down-sampled query and reference points. Φ_* is 1×1 Convolution, $[\cdot,\cdot]$ is the concatenation operation. "$\cdot$" denotes dot-product and σ is a $softmax$ function.

As discussed above, the non-local mechanism-based spatiotemporal attention module only models global relationships, lacking modeling of the local aggregation nature of point clouds. The variant model 3DSTM-TR adopts the local transformer to construct the spatiotemporal module to alleviate the above challenge. As shown in Fig. 1(c), the transformer-based interaction module consists of two local self-attention blocks (LSA) [21] and one local cross-attention block (LCA) designed in the spirit of LSA. The two LSAs are exploited to enhance the features of the reference and query points, respectively. While the LCA is adopted to perform the features interaction between the reference and adjacent query points. We show the structure of the LSA in Fig. 1(d). Specifically, the LSA can be expressed as:

$$q^* = \Phi_q(F_*), k^* = \Phi_k(F_*), v^* = \Phi_v(F_*) \tag{3}$$

$$y(p) = \sum_{n\in\mathcal{N}(p)} \sigma(\gamma(q^*(p) - k^*(n) + \delta)) \cdot (v^*(n) + \delta) \tag{4}$$

$$O = \alpha(y) + F_* \tag{5}$$

where "*" denotes "R" (reference) or "Q" (query). Φ_* is linear projection. p denotes points in query maps q^*. $\mathcal{N}(p)$ is a set of points in a local neighborhood (specifically, k nearest neighbors) of p. "$\cdot$" denotes dot-product and σ is a $softmax$ normalization function. δ is positional encoding function $\delta = \theta(p_i - p_j)$. p_i and p_j are the 3D point coordinates. The mapping function α, γ and θ are MLPs with two linear layers and one ReLU function.

As shown in Fig. 1(e), in the spirit of the LSA, we designed the LCA, which can be expressed as:

$$q^Q = \Phi_q(F_Q), k^R = \Phi_k(F_R), v^R = \Phi_v(F_R) \tag{6}$$

$$y(p) = \sum_{n\in\mathcal{N}(p,\Omega_R)} \sigma(\gamma(q^Q(p) - k^R(n) + \delta)) \cdot (v^R(n) + \delta) \tag{7}$$

$$O = \alpha(y) + F_Q \tag{8}$$

where $\mathcal{N}(p, \Omega_R)$ is the set of points of reference points Ω_R in the neighborhood of the query point p.

Decoder. We construct a decoder similar to the segmentation decoder of PointNet++ to predict the segmentation masks of the target objects. The decoder takes the matching features from the spatiotemporal attention module and the hierarchical features of the Siamese encoder as input, gradually upscaling the feature maps with three stages. Each stage takes both the previous stage's output and the feature map from the Siamese encoder at the corresponding scale through skip connection and upscales the compressed feature map by two at a time. The output of the last stage is passed through the final MLP followed by a $softmax$ operation to predict the object mask.

3.4. Integrate tracker for LiDAR-based VOS

To provide a more comprehensive evaluation, we added a simple post-processing operation after a 3D object tracker to form the second type of baseline, i.e., tracking-based baselines. Since the KITTI-VOS only provides segmentation annotations, we use the segmentation masks to obtain the objects' points, which are further used to generate bounding boxes. Then we trained two representative tracking approaches, i.e., P2B [23] and MLVSNet [25], on the extended KITTI-VOS to acquire the trackers. After training the trackers, we simply choose those points inside the bounding box predicted by the tracker as the foreground points to obtain the segmentation masks.

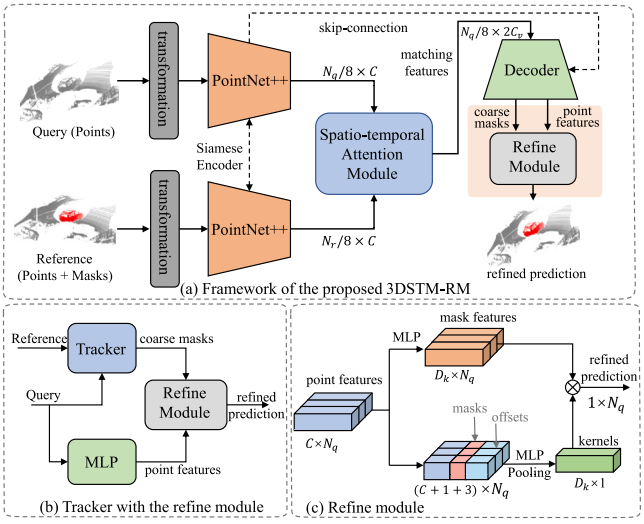

Fig. 2. The frameworks of these baselines equipped with the proposed refine module. The architecture of the refine module is illustrated in (c), which consists of a kernel branch and a mask branch.

3.5. Enhance baselines with refine module

In this section, we propose a refine module to boost these baselines' performance. The refine module is plug-and-play, which can be easily extended to the methods discussed above and trained end-to-end. For memory-based baselines, the refine module takes the predicted coarse segmentation masks as priors to further improve the masks' qualities (Fig. 2(a)). And for tracking-based baselines, the refine module serves as a segmentation head to obtain more accurate predictions with the generated coarse masks from the predicted boxes as priors (Fig. 2(b)). As shown in Fig. 2(c), the refine module consists of two branches, i.e., the kernel branch and mask branch, which are composed of a stack of fully-connected layers. We first calculate the offsets between each point to the object centroids (mass centers of the coarse masks) to inject the object-aware positional information. Then the offsets, mask priors, and the point features are concatenated along the feature dimension and fed into the kernel branch to generate the 1-D object-aware kernels $K \in \mathbb{R}^{1\times D_k}$. Meanwhile, the mask branch takes point features as input and aims to encode mask features $F_M \in \mathbb{R}^{N_q\times D_k}$. Thus, given K and F_M, the refined segmentation masks are produced by $M = K \bigotimes F_M$. Here, $\bigotimes$ denotes the convolution operation. Note that the point features for memory-based baselines are taken from the output of the decoder's last stage (Fig. 2(a)). While for tracking-based baselines, we use another lightweight MLP to encode the point features from the query point clouds (Fig. 2(b)).

4. Experiments

4.1. Implementation details

The output dimension D of the input transformation module is set to 16. The first three stages of the Siamese encoder are used, and output features with channel $[128, 256, C = 256]$. The dimensions C_k, C_v of the pairs of key and value maps in the spatiotemporal attention module used in 3DSTM are 64 and 128. The hidden dimension of the

Table 2
Quantitative results on the validation of the constructed KITTI-VOS dataset. The evaluation metric is the average of $IoU(\%)$. 'RM' indicates equipping with the refine module.

Methods	Car	Person	Truck	Cyclist	Mean
3DSTM	29.5	14.2	36.1	63.8	35.9
3DSTM-TR	33.0	21.4	**38.9**	**64.7**	**39.5**
P2B [23]	**58.2**	24.8	11.0	59.8	38.5
MLVS [25]	53.4	**28.5**	31.4	43.8	39.3
P2B-RM	66.7	31.0	21.5	59.1	44.6(+6.1)
MLVS-RM	**72.7**	**44.6**	40.6	55.2	**53.3**(+14)
3DSTM-TR-RM	39.5	21.7	**43.4**	**69.8**	43.6(+5.1)

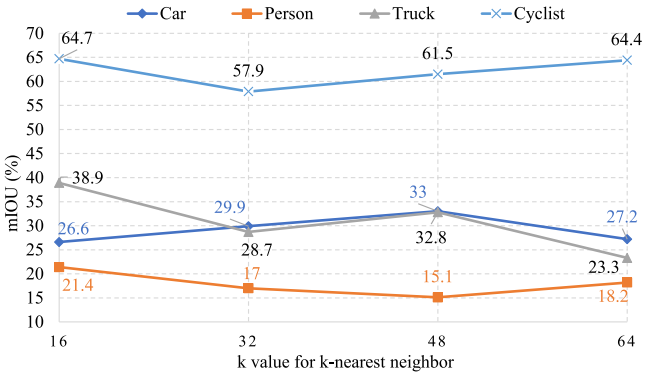

Fig. 3. The exploration of hype-parameter k in local transformers.

transformer-based attention module is set to 256. The hyper-parameter k of the LSA and LCA is set to 16 by default. The dimension of the object kernel is $D_k = 64$. Similar to [23], the input points are cropped and normalized according to the targets. Following many 3D SOT methods [21,23–25], we train and evaluate our models on four target types (Car, Person, Truck, Cyclist), respectively. And all segmentation models are trained using cross-entropy loss on a single TITAN RTX GPU.

4.2. Main results

Table 2 reports the evaluation results on the constructed KITTI-VOS dataset. The reference frame is set to the previous frame for all these baselines for a fair comparison. Compared with 3DSTM, 3DSTM-TR's performances are boosted by 3.5% on cars, 7.2% on the person, 2.8% on trucks, and 0.9% on cyclists, demonstrating the effectiveness of the local attention in transformer-based spatiotemporal attention module.

As shown in Table 2, when comparing tracking-based baselines (P2B, MLVS) and memory-based baselines (3DSTM, 3DSTM-TR), we found the former performs better on cars and persons while the latter achieves better performance on trucks and cyclists. And compared to tracking-based baselines, memory-based baselines achieve a significant improvement on trucks. We explain that tracking-based baselines cannot properly track large objects like trucks, which are always under-segmentation, especially when LiDAR points are sparsely distributed on the part of their surfaces since it is difficult to accurately estimate the center and size of the object from limited observations. On the contrary, the memory-based baselines can effectively solve the under-segmentation problem of large objects and are more suitable for distraction-free scenes since the matching procedure is easily disturbed by noisy points. Thus, the memory-based methods perform worse than the tracking-based methods on cars and persons, which are usually surrounded by more distracting objects that are hard to distinguish. The second part of Table 2 shows that the proposed refine module can significantly boost these baselines' performance. By taking the coarse masks as priors, the refine module can improve the ability to handle distractions and break the tracker's limitations. For example, MLVS-RM gains 19.3% on cars against MLVS. And refine module also improves the 3DSTM-TR's performance on cars from 33.0% to 39.5%.

Table 3
Components analysis on LSA.

Variants	Car	Person	Truck	Cyclist
w/o LSA	29.6	11.3	36.9	62.2
w/ LSA	**33.0**$_{\uparrow 3.4}$	**21.4**$_{\uparrow 10.1}$	**38.9**$_{\uparrow 2.0}$	**64.7**$_{\uparrow 2.5}$

Table 4
The exploration of reference frames.

Reference	Car	Person	Truck	Cyclist
First frame	32.8	21.3	31.0	63.7
Previous frame	**33.0**	21.4	38.9	64.7
First & previous	32.3	**21.7**	**39.7**	65.0
All previous frames	32.9	21.3	38.7	**65.4**

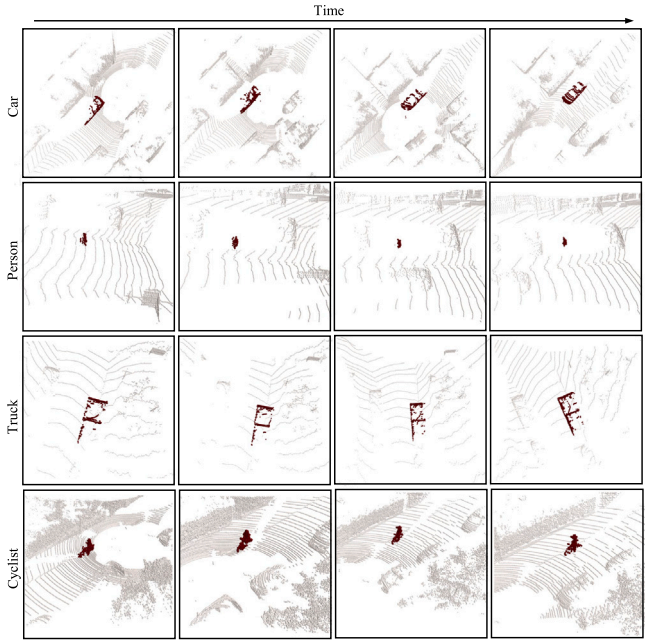

Fig. 4. Visualizations on the validation set of the KITTI-VOS dataset. 3DSTM-TR learned to discriminate the target points from the background in multiple scenes.

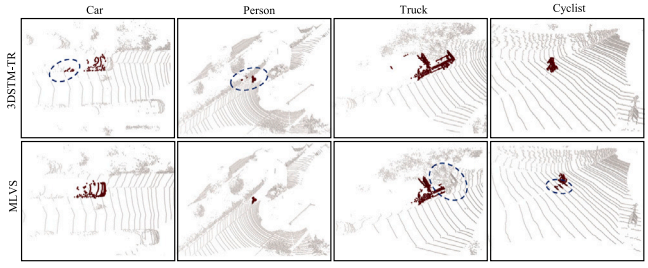

Fig. 5. Visual comparison between 3DSTM-TR and MLVS on the validation set of the KITTI-VOS dataset.

4.3. Visualizations

We show qualitative results to provide more analysis on these baselines. Firstly, we visualize the representative 3DSTM-TR in Fig. 4. We can observe that 3DSTM-TR had learned to robustly discriminate the target points from the background in multiple scenes. Furthermore, we illustrate the visual comparison between two types of baselines (3DSTM-TR vs. MLVS) in Fig. 5. 3DSTM-TR is more easily affected by distractions, which performs worse on cars and persons. In contrast, MLVS performs worse on large objects such as trucks due to the inaccuracy of bounding boxes. Besides, the cropped points contain many noisy points of ground (cyclists).

4.4. Ablation study

We further provide the ablation study on the constructed dataset to analyze the effect of the individual components of memory-based baselines. The reference frame is set to the previous frame by default.

The effectiveness of local transformer. We conduct an ablation study on the components of the spatiotemporal attention module in 3DSTM-TR. We trained models without the LSA to demonstrate its effectiveness. Table 3 shows that with LSA, the performance is higher than without it. LSA brings improvements of IoU by 3.4% on cars, 10.1% on the person, 2.0% on trucks, and 2.5% on cyclists.

The value k for k-nearest neighbor. We do experiments to investigate the number of neighbors k used to determine the local neighborhood around each point in the LSA and LCA of 3DSTM-TR. The results are shown in Fig. 3. The best performance is achieved for the category car when k is set to 48. The model's performance drops dramatically when the neighborhood is smaller ($k = 16$). We explain that the LSA and LCA cannot learn sufficient context information under the lower k setting. For other categories, a larger neighborhood ($k = 32, k = 48, k = 64$) damages the model's performance. The possible reason is that the LSA and LCA would be provided with more irrelevant points when k is larger, introducing excessive noise into the attention-based feature-matching process. And due to the limited training samples, the model cannot learn the accurate local correlation and exclude the interference caused by a large k value.

Effect of different reference frames. Finally, we investigate the effect of choosing different reference frames on 3DSTM-TR. We consider four configurations: (1) using the first LiDAR frame as the reference; (2) initially using the first LiDAR frame and then the previous LiDAR frame. (3) take the first and previous frames as reference. (4) take all the historical frames as reference. As shown in Table 4, for the "truck" category, the model performs poorly with only the first frame as the reference. We explain that the target's object information, especially for large objects, may not be fully reflected in the first frame, which typically contains a limited number of target points since it is always sensed from a large distance. We also notice that not all previous frames are beneficial since the historical frames with low-quality segmentation may mislead the subsequent prediction. And for other categories, such as "car" and "person", the first frame plays a more important role, and the influence of other historical frames is slight.

5. Discussion about the initialization

As mentioned in Section 3.1, the initial indicator of the target is needed in the setting of LiDAR-based VOS. However, obtaining accurate information about the target object for the initialization can be challenging in practical applications. We can use a hand-labeled bounding box to initialize the tracking-based baselines during inference. On the other hand, memory-based baselines require the initial segmentation mask, which can also be generated from an accurate tightened bounding box. Besides, clustering algorithms based on the manually selected centers could also be exploited to generate the initial masks. We will try to implement weak initialization, such as scribbles or points, in future work.

6. Conclusions

This work first performs LiDAR-based 3D Video Object Segmentation (VOS) for practical application scenarios of robotics and constructs a LiDAR-based VOS dataset dubbed KITTI-VOS based on SemanticKITTI. Besides, based on KITTI-VOS, two types of baselines, i.e., memory-based and tracking-based baselines, are provided for comprehensive evaluation. Furthermore, a refine module is developed to boost all baselines' performances, improving the ability to handle distractions, and break trackers' limitations. For example, MLVS-RM gains 19.3% on cars against MLVS. The refine module also improves the 3DSTM-TR's performance on cars from 33.0% to 39.5%. We hope this work will foster a new direction in this line of research. And in future work, we will try to explore label-efficient algorithms such as weakly supervised methods to alleviate the cost of point-wise labels.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgments

This work was supported by a Grant from The National Natural Science Foundation of China (No. U21A20484).

References

[1] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, Philip HS Torr, Fast online object tracking and segmentation: A unifying approach, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 1328–1338.
[2] Vipul Sharma, Roohie Naaz Mir, SSFNET-VOS: Semantic segmentation and fusion network for video object segmentation, Pattern Recognit. Lett. 140 (2020) 49–58.
[3] Paul Voigtlaender, Jonathon Luiten, Philip HS Torr, Bastian Leibe, Siam r-cnn: Visual tracking by re-detection, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 6578–6588.
[4] Xuhua Huang, Jiarui Xu, Yu-Wing Tai, Chi-Keung Tang, Fast video object segmentation with temporal aggregation network and dynamic template matching, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 8879–8889.
[5] Paul Voigtlaender, Yuning Chai, Florian Schroff, Hartwig Adam, Bastian Leibe, Liang-Chieh Chen, Feelvos: Fast end-to-end embedding learning for video object segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 9481–9490.
[6] Haochen Wang, Xiaolong Jiang, Haibing Ren, Yao Hu, Song Bai, Swiftnet: Real-time video object segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 1296–1305.
[7] Seoung Wug Oh, Joon-Young Lee, Ning Xu, Seon Joo Kim, Video object segmentation using space-time memory networks, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 9226–9235.
[8] Yongqing Liang, Xin Li, Navid Jafari, Jim Chen, Video object segmentation with adaptive feature bank and uncertain-region refinement, Adv. Neural Inf. Process. Syst. 33 (2020) 3430–3441.
[9] Haozhe Xie, Hongxun Yao, Shangchen Zhou, Shengping Zhang, Wenxiu Sun, Efficient regional memory network for video object segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 1286–1295.
[10] Meng Lan, Jing Zhang, Fengxiang He, Lefei Zhang, Siamese network with interactive transformer for video object segmentation, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 36, (2) 2022, pp. 1228–1236.
[11] Jens Behley, Martin Garbade, Andres Milioto, Jan Quenzel, Sven Behnke, Cyrill Stachniss, Jurgen Gall, Semantickitti: A dataset for semantic scene understanding of lidar sequences, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 9297–9307.
[12] Shervin Minaee, Yuri Boykov, Fatih Porikli, Antonio Plaza, Nasser Kehtarnavaz, Demetri Terzopoulos, Image segmentation using deep learning: A survey, IEEE Trans. Pattern Anal. Mach. Intell. 44 (7) (2021) 3523–3542.
[13] Guangchen Shi, Yirui Wu, Jun Liu, Shaohua Wan, Wenhai Wang, Tong Lu, Incremental few-shot semantic segmentation via embedding adaptive-update and hyper-class representation, in: Proceedings of the 30th ACM International Conference on Multimedia, 2022, pp. 5547–5556.
[14] Chen Chen, Chenyu Wang, Bin Liu, Ci He, Li Cong, Shaohua Wan, Edge intelligence empowered vehicle detection and image segmentation for autonomous vehicles, IEEE Trans. Intell. Transp. Syst. (2023).
[15] Yirui Wu, Lilai Zhang, Zonghua Gu, Hu Lu, Shaohua Wan, Edge-AI-driven framework with efficient mobile network design for facial expression recognition, ACM Trans. Embedded Comput. Syst. 22 (3) (2023) 1–17.
[16] Mengmeng Wang, Jianbiao Mei, Lina Liu, Guanzhong Tian, Yong Liu, Zaisheng Pan, Delving deeper into mask utilization in video object segmentation, IEEE Trans. Image Process. 31 (2022) 6255–6266.

[17] Xiankai Lu, Wenguan Wang, Martin Danelljan, Tianfei Zhou, Jianbing Shen, Luc Van Gool, Video object segmentation with episodic graph memory networks, in: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16, Springer, 2020, pp. 661–679.

[18] Jianbiao Mei, Mengmeng Wang, Yu Yang, Yanjun Li, Yong Liu, Fast real-time video object segmentation with tangled memory network, ACM Trans. Intell. Syst. Technol. (2023).

[19] Li Hu, Peng Zhang, Bang Zhang, Pan Pan, Yinghui Xu, Rong Jin, Learning position and target consistency for memory-based video object segmentation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 4144–4154.

[20] Yue Zhang, Fanghui Zhang, Yi Jin, Yigang Cen, Viacheslav Voronin, Shaohua Wan, Local correlation ensemble with GCN based on attention features for cross-domain person re-ID, ACM Trans. Multimed. Comput., Commun. Appl. 19 (2) (2023) 1–22.

[21] Changqing Zhou, Zhipeng Luo, Yueru Luo, Tianrui Liu, Liang Pan, Zhongang Cai, Haiyu Zhao, Shijian Lu, Pttr: Relational 3d point cloud object tracking with transformer, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 8531–8540.

[22] Silvio Giancola, Jesus Zarzar, Bernard Ghanem, Leveraging shape completion for 3d siamese tracking, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 1359–1368.

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

[24] Chaoda Zheng, Xu Yan, Jiantao Gao, Weibing Zhao, Wei Zhang, Zhen Li, Shuguang Cui, Box-aware feature enhancement for single object tracking on point clouds, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 13199–13208.

[25] Zhoutao Wang, Qian Xie, Yu-Kun Lai, Jing Wu, Kun Long, Jun Wang, MLVSNet: Multi-level voting siamese network for 3D visual tracking, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 3101–3110.

[26] Jiayao Shan, Sifan Zhou, Zheng Fang, Yubo Cui, PTT: Point-track-transformer module for 3D single object tracking in point clouds, in: 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), IEEE, 2021, pp. 1310–1316.

[27] Le Hui, Lingpeng Wang, Linghua Tang, Kaihao Lan, Jin Xie, Jian Yang, 3D siamese transformer network for single object tracking on point clouds, in: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part II, Springer, 2022, pp. 293–310.

[28] Charles R. Qi, Or Litany, Kaiming He, Leonidas J. Guibas, Deep hough voting for 3d object detection in point clouds, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2019, pp. 9277–9286.

[29] Chaoda Zheng, Xu Yan, Haiming Zhang, Baoyuan Wang, Shenghui Cheng, Shuguang Cui, Zhen Li, Beyond 3D siamese tracking: A motion-centric paradigm for 3D single object tracking in point clouds, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 8111–8120.

[30] Fangzhou Hong, Hui Zhou, Xinge Zhu, Hongsheng Li, Ziwei Liu, Lidar-based panoptic segmentation via dynamic shifting network, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 13090–13099.

[31] Vinod Nair, Geoffrey E. Hinton, Rectified linear units improve restricted boltzmann machines, in: Icml, 2010.

[32] Sergey Ioffe, Christian Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: International Conference on Machine Learning, PMLR, 2015, pp. 448–456.

[33] Charles Ruizhongtai Qi, Li Yi, Hao Su, Leonidas J Guibas, Pointnet++: Deep hierarchical feature learning on point sets in a metric space, in: Advances in Neural Information Processing Systems, vol. 30, 2017.