You Don't Only Look Once: Constructing Spatial-Temporal Memory for Integrated 3D Object Detection and Tracking

Jiaming Sun[1,2*] Yiming Xie[1*] Siyu Zhang[2] Linghao Chen[1] Guofeng Zhang[1]
Hujun Bao[1] Xiaowei Zhou[1†]

[1]Zhejiang University [2]SenseTime Research

Abstract

Humans are able to continuously detect and track surrounding objects by constructing a spatial-temporal memory of the objects when looking around. In contrast, 3D object detectors in existing tracking-by-detection systems often search for objects in every new video frame from scratch, without fully leveraging memory from previous detection results. In this work, we propose a novel system for integrated 3D object detection and tracking, which uses a dynamic object occupancy map and previous object states as spatial-temporal memory to assist object detection in future frames. This memory, together with the ego-motion from back-end odometry, guides the detector to achieve more efficient object proposal generation and more accurate object state estimation. The experiments demonstrate the effectiveness of the proposed system and its performance on the ScanNet and KITTI datasets. Moreover, the proposed system produces stable bounding boxes and pose trajectories over time, while being able to handle occluded and truncated objects. Code is available at the project page: https://zju3dv.github.io/UDOLO.

1. Introduction

Humans start to develop the spatial working memory in an early age [38, 50], resulting in the awareness of the spatial object arrangement of their surroundings as part of the mental "World Model" [20]. With this memory serving as prior knowledge of 3D object locations, together with an estimation of the ego-motion of the eyes, we would anticipate objects to appear in certain regions in the field of view when we look around. This ability enables humans to continuously locate, track and recognize objects in the 3D space, even under severe occlusion or truncation.

However, 3D object detection in most of the existing tracking-by-detection systems still processes each input image from a video stream individually and searches for objects in every new frame from scratch. Then, object tracking is usually performed as a post-processing step to associate the detected bounding boxes with the previously observed tracklets, followed by a recursive filter or optimization to improve the accuracy and temporal stability of the estimated object states. Although this late integration of temporal information would improve the results, the object detection module is still performed on a per-frame basis without using the memory of the objects in the surrounding scene, which is not only counter-intuitive but also very inefficient.

We argue that the key to making full use of temporal information is not only to track objects and smooth object states, but more importantly to feed the temporally-accumulated memory of object states back to the detection module, yielding an integrated detection and tracking system. To this end, we propose a novel system named UDOLO that enables the object detector to take the spatial-temporal memory as a strong prior for more efficient and accurate 3D object detection and tracking, as illustrated in Fig. 1.

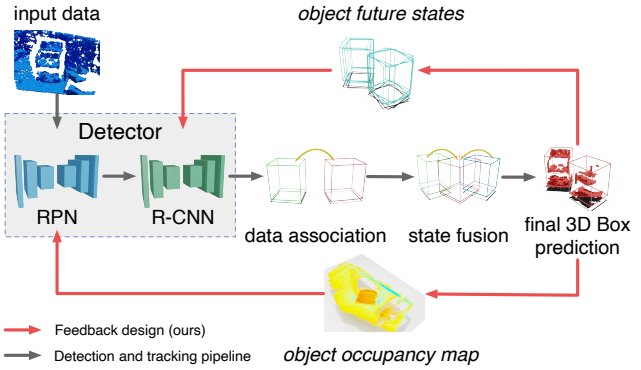

Figure 1. **The spatial-temporal memory in UDOLO** is represented by the object occupancy map and the object future states. The system is able to feedback this memory into the detection pipeline as strong prior to produce better object detection results. **Red arrows** represent the information flow of the feedback.

*The first two authors contributed equally. The authors are affiliated with the State Key Lab of CAD&CG and ZJU-SenseTime Joint Lab of 3D Vision. †Corresponding author: Xiaowei Zhou.

Specifically, UDOLO simultaneously detects and tracks 3D objects in a point cloud sequence, either obtained directly from depth sensors or estimated with multiple views of images. The core idea of UDOLO is to maintain a dynamic object occupancy map (OOM) and object future state predictions as spatial-temporal memory to assist object detection in future frames. The OOM is a 2D Bird's Eye View (BEV) map which shows the likelihood of a location being occupied by an object in the world frame. The OOM is constructed by registering the 3D observations in previous frames to the world frame, given the camera poses from the back-end odometry, and fusing the occupancy states at each location. Next, the UDOLO system integrates the spatial-temporal memory into the modern two-stage object detection pipeline [40, 47] at two different levels: early integration and middle integration. With the **early integration**, the Region Proposal Network (RPN) in the front-end object detector extracts object proposals only in the regions that have high object occupancy scores given by the OOM and unobserved regions where new objects may appear. The early integration design essentially reduces the search space during the region proposal stage and saves the effort on evaluating tens of thousands of densely arranged predefined anchors in the standard 3D object detector [40, 47], thus leading to more efficient detection. With the **middle integration**, current-frame object proposals are fused with back-end object future state predictions by combining the RoI point clouds and passing them through the second stage of the detector to produce the front-end bounding box estimation. The middle integration design enables the detector to leverage the optimized and predicted object future states from back-end, in order to produce more accurate bounding boxes and handle truncation and occlusion.

We evaluate our system on the ScanNet and KITTI datasets and provide ablation analyses on different components of the system. The experiments show that, with the spatial-temporal memory fed back into the object detection pipeline, 3D object detection performance can be largely improved compared to single-frame detection-only baselines in both indoor and outdoor scenes. For the dynamic scenes on KITTI, 3D multiple object tracking (MOT) is also improved by a large margin. The system is also capable of detecting occluded or truncated objects in cluttered indoor environments and produces more stable object bounding boxes.

In summary, our contributions are as follows:

- A novel framework of integrated detection and tracking that feeds the spatial-temporal memory of objects all the way through the detection pipeline to improve both efficiency and accuracy, which has not been explored in the literature to our knowledge.
- An early integration scheme based on a new representation named object occupancy map (OOM) to generate high-quality object proposals and speed up detection and tracking.
- A middle integration design to fuse object state predictions from previous frames with estimations at the current frame to achieve better detection and tracking performance and truncation or occlusion handling.

2. Related Work

3D Object Detection in Point Clouds. 3D object detection plays a central role in our framework. Because of the nature of the proposed method, we only focus on 3D object detectors that are based on point cloud input. To estimate 3D bounding box of objects, the point cloud is usually captured by 3D sensors (RGB-D camera or LiDAR) [7, 48] or estimated from images [6, 57]. In the indoor environment, VoteNet [39, 40] takes the reconstructed point cloud of a scene as input and regresses voting offsets to extract object proposals. In the outdoor driving scenario, PointRCNN [47] and PV-RCNN [46] directly take the point cloud representation as input and design a two-stage pipeline similar to the 2D detection counterpart [43] and achieve impressive performance. Pseudo-LiDAR and its subsequent works [41, 51, 60] use point cloud estimated from images as the input for point-cloud-based 3D detectors and demonstrated state-of-the-art performance. Although these detectors achieve high AP results on benchmarks, they still suffer from occlusion or missing detection in real-world practices since the system has no memory of the surrounding scene. Our work attempts to tackle this problem by incorporating spatial-temporal memory as strong priors into the detection pipeline.

Object Tracking in Video. Other than the object detection module, a typical tracking-by-detection system usually consists of a Multi-Object Tracking (MOT) module to provide object association and an object state estimation module to fuse temporal observations of the same object and estimate its state (position, heading, velocity, etc.) [21]. Recent works on 3D MOT [8, 17, 44, 45, 52, 54, 62] primarily focus on the data association problem based on geometric or appearance cues. The detected bounding boxes are measured with the cues and linked by solving a linear-assignment problem. For object state estimation, [12, 15] combine 2D object detection with visual-inertial odometry achieved by Extended Kalman Filter (EKF) to make persistent 3D state estimation of objects. [31] proposed a novel pipeline to estimate 3D object states in a nonlinear least-square optimization setting. 3D object bounding boxes are initialized by a 2D detection front-end and then optimized by re-projection error of sparse features anchored on the objects. A similar pipeline is proposed in [58]. This pipeline is recently extended with an learning-based object coordinate estimation front-end in [32] and achieves impressive performance. Other than using a 3D object detection front-end to provide direct 3D observations, the major difference between these works and ours

is the feedback design, which fully utilizes the optimized object state from back-end to assist 3D detection front-end at the next time step. It is also worth noting that the overall objective of OOM-Guided RPN in the proposed framework is similar to some related work in 2D and 3D single object tracking [19,30,61], in the sense of reducing the search space for object proposals with temporal accumulated information. However, the major focus of UDOLO is to leverage the accumulated objectness distribution as spatial prior without explicitly distinguishing between individual objects, instead of relocalizing the same object between frames with similarity search based on the appearance.

Integrated Detection and Tracking. Our work is also closely related to the field of video object detection, where the major goal is to leverage temporal information to boost object detection performance. Several works attempt to tackle this problem by temporal feature aggregation in the context of 2D object detection [27, 28, 56, 66]. Notably, [16] proposed to integrate detection and tracking into a unified neural network. [63] proposed a tracklet-conditioned 2D detection pipeline, where object proposals produced from RPN are re-scored before the NMS process according to feature embedding similarities. [56] proposed a spatial-temporal memory module to fuse and align temporal convolutional features of objects. Tracktor++ [3] and CenterTrack [64] directly use the bounding box regression modules from object detection for tracking. As for 3D object detection and tracking, [33,34] proposed a unified neural network to jointly perform 3D object detection, tracking and motion prediction. [24] proposed a joint detection and tracking system with monocular images input. [5] integrates a 3D Kalman filter into a 3D detection system to improve localization accuracy. [26,59] use RNNs to aggregate temporal information for more accurate 3D object detection. There are also some works that exploit BEV map representations that are related to the proposed OOM. Notably, [11, 14, 22, 37] proposed OGM [13] and Dynamic OGM (DOGMa) [36] for object detection or tracking. The major difference between OGM and OOM is that OOM represents the likelihood of a location being occupied by any object, while OGM is a geometric map representation constructed from fusing multiple sensor sources. Recently, [25] proposed a visibility map in BEV to assist distinguishing the ambiguities in 3D object detection.

The proposed method UDOLO falls into this category, but with an emphasis on the feedback of spatial-temporal memory into the detection pipeline. None of the above-mentioned works, especially on the 3D side, attempts to maintain an explicit and persistent object occupancy representation as spatial memory and integrates it into the object proposal stage of the detection pipeline. Without the early integration scheme proposed in our framework, the detector still cannot leverage temporal prior knowledge at the stage that searches for objects, resulting in less efficient object proposal and missing detection due to occlusion or truncation.

3. Methods

Given a point cloud sequence $\{\mathbf{P}_t\}$ and camera pose trajectory $\{\xi_t\} \in \mathbb{SE}(3)$ provided by an odometry system, the goal is to detect and track all the 3D bounding boxes $\{b_t\}$ of interesting objects and estimate true object states $\{\mathbf{x}_t\}$ with temporal accumulated observations[1]. Following conventions in the SLAM community, we consider the modules that make current-frame predictions as front-end and the modules that aggregate temporal observations and predict future object states as back-end. The system diagram and pipeline overview are given in Fig. 2. Different from the standard tracking-by-detection system, UDOLO consists of two feedback designs that deeply integrate temporal information as prior knowledge into the object detection pipeline.

3.1. Object Occupancy Map

Previous works use object tracklets that contain linked 3D bounding boxes to represent and preserve history observations. However, the bounding box representation alone cannot capture the full state of previous detections. Ideally, the representation of human-like spatial-temporal memory should not only tell us where there *are* objects but also where there should be *no* object. We propose a map representation to record and accumulate the object occupancy state in the world frame. The OOM is constructed with the point cloud input and the corresponding object future state predictions produced at the last time step from the back-end motion prediction module. As shown in Sec. 3.2, this representation can serve as a selective attention mechanism to the surrounding world that guides the object detector to only focus on the important part of the scene.

The object occupancy map is represented as a 2D grid in the BEV (top-down view), and each grid cell corresponds to an area in the world frame defined by grid size. The updating process of OOM is visually illustrated in Fig. 3. Conceptually, the OOM construction is similar to the 2D BEV version of the mapping process in dense reconstruction systems like KinectFusion [35]. The major difference is that for OOM construction the goal is to fuse the object occupancy state observations from each frame, instead of reconstructing dense geometry. Formally, object occupancy map M_t is updated by the following equation:

$$M_t = \text{Voxelize}(\phi(\xi_t^{-1}(\mathbf{P}_t))) + M_{t-1}$$

where $\phi(p_i) = +\gamma$ if point p_i is in any bounding box of object future states and -1 otherwise. $\gamma = \lambda s_{\mathbf{x}}^t$ is the fused object classification score rescaled by a factor λ, which we will further illustrate in Sec. 3.3. We use $\xi_t^{-1}(\cdot)$ to represent

[1] Since the system processes different objects independently, only one object instance from one category is considered to simplify the notation.

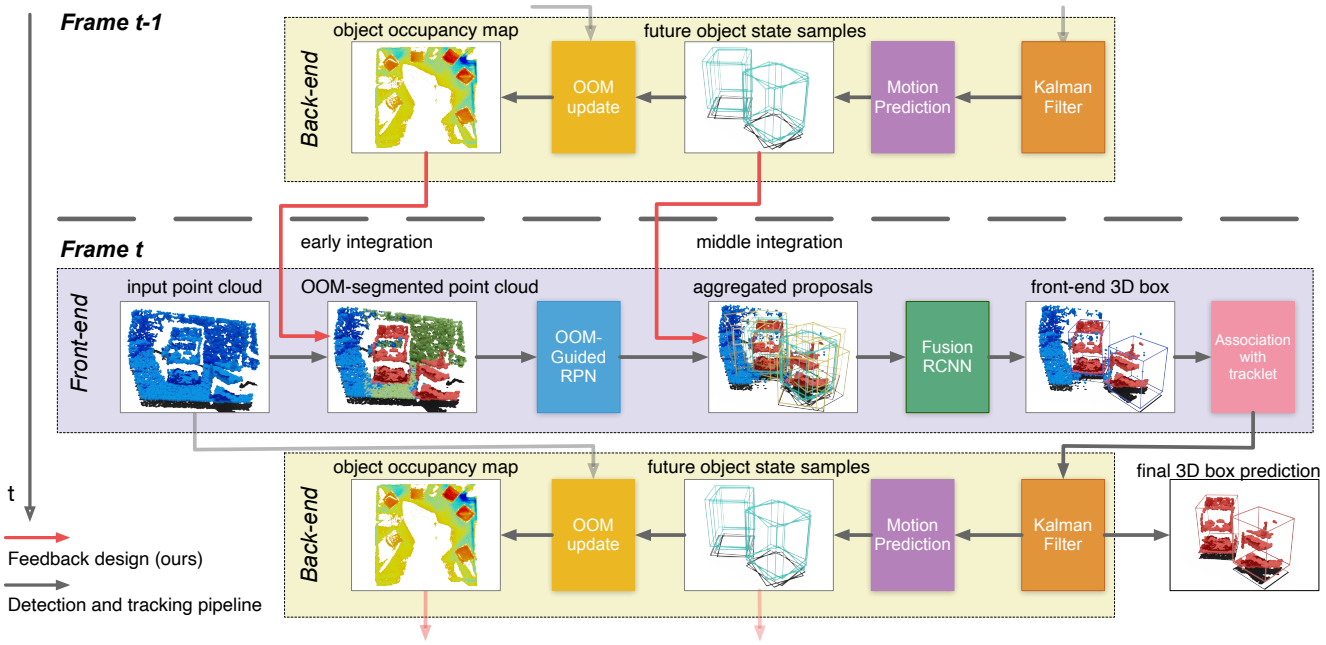

Figure 2. **UDOLO system diagram.** Solid arrows denote the major data flow. Red arrows denote the feedback mechanism design. At each time step, the front-end region proposal network (**OOM-Guided RPN**) takes the point cloud as input, extracts current-frame object proposals only in the regions that have high object occupancy scores (red points) given by the object occupancy map, as well as the unobserved regions (blue points) where new objects may appear. These proposals are later fused with back-end object future state predictions from the last frame and passed through the second stage of the detector (**Fusion R-CNN**). After association with the tracklet, current front-end predictions are fed into the **Kalman Filter** to produce the fused object states as the final bounding box prediction. Then the object occupancy map is updated according to the future object states given by the motion prediction module. Best viewed in color (zoom in for details).

the transformation from the camera frame to the world frame and Voxelize$(\cdot)$ to represent the operation that converts the world-frame point cloud into a 2D grid. M_0 is initialized as a 2D grid with all zero values.

3.2. Early Integration: OOM-Guided RPN

In order to make the detector only detect in the previously unobserved region and continue detecting observed objects, the integration must happen at the very beginning of the detection pipeline.

Given the object occupancy map from the last frame, by selecting a threshold σ, the input point cloud is separated into three parts according to the corresponding occupancy score in grid locations (x, y) in the world frame:

$$\begin{cases} M_t(x,y) > \sigma & \text{high occupancy area} \\ M_t(x,y) < -\sigma & \text{low occupancy area} \\ -\sigma \leq M_t(x,y) \leq \sigma & \text{unobserved area} \end{cases}$$

Since we don't need to detect in the areas that are certain to have no object, only the point cloud in the high occupancy area and the unobserved area is passed through the point cloud backbone and RPN. This process is visually illustrated in Fig. 3. By discarding areas that contain no object based on the temporally fused OOM, the computational cost is saved at the backbone since fewer points are passed into. The number of anchors and generated proposals is also drastically reduced, thus saving the computation time on the RPN and 3D Non-maximum Suppression (NMS) without skipping any input frames. The non-stop detection in each frame potentially improves the reliability of the system compared to some video object detection methods [65] that skip input frames to speed up the detection pipeline.

3.3. Middle Integration: Fusion R-CNN

After object proposals are generated from the RPN, the spatial-temporal memory is further integrated at the R-CNN stage. The fusion step involves two groups of bounding boxes: current-frame object proposals $\{b_{prop}\}$ and object future state prediction samples from the back-end $\{b_{pred}\}$, which makes the fusion process conceptually similar to the correction step in the Kalman Filter. These bounding boxes are first fused in the bounding box regression network by passing through the combination of the RoI point cloud of $\{b_{prop}\}$ and $\{b_{pred}\}$ cropped by the 3D RoI Pooling module [40, 47]. $\{b_{pred}\}$ also participates in the 3D NMS process with the fused object score, where the fusion is achieved by a moving average of the object confidence scores of sequential observations: $s_{\mathbf{x}}^t = \frac{s_b^t + \alpha s_{\mathbf{x}}^{t-1}}{1+\alpha}$. $s_{\mathbf{x}}^t$ denotes the

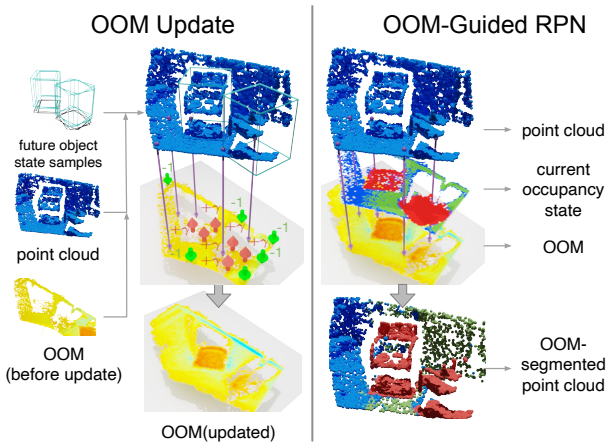

Figure 3. **OOM update process and OOM-Guided RPN illustration.** Red: high occupancy area. Green: low occupancy area. Blue: unobserved area. Best viewed in color (zoom in for details).

score for back-end bounding box from frame t and s_b^t is the score for detected 3D bounding box from frame t. $\alpha \in [0, 1]$ is a decay parameter. Fusion R-CNN produces the current-frame observations at the front-end, denoted as $\{b_{front}\}$. A visual illustration of the Fusion R-CNN is given in Fig. 4.

The Fusion R-CNN incorporates current-frame object proposals and temporally fused and predicted back-end object future states into the bounding box regression network (R-CNN), which makes the front-end bounding box predictions more accurate and stable. The RoI point cloud fusion step before the R-CNN will ensure that the point cloud of the corresponding object of interest will be passed through the network in case the RPN failed to produce accurate object proposals. With the integration of $\{b_{pred}\}$ into the NMS process, the temporally-optimized object state from previous frames further contributes to the overall robustness of the system. Even the R-CNN fails to predict bounding boxes with a high-enough confidence score due to severe truncation or occlusion, the NMS module will still be able to select the correct predictions with the highest score from $\{b_{pred}\}$ thanks to the score fusion process.

3.4. Other Components

Association. The role of the association module is to find correspondences between current-frame detection results $\{b_{front}\}$ with back-end tracklets. Although more complicated methods can be used, since similarity estimation and correspondence searching are not the major focus of this work, we simply choose 3D IoU based association and solve the bipartite matching with the Hungarian algorithm [29].

Kalman Filter and Motion Prediction. The Kalman filter (KF) is used to fuse $\{b_{front}\}$ and future state predictions $\{b_{pred}\}$ from the last frame and estimate the true object states with temporal observations. We follow AB3DMOT [52] for all the parameter settings in the KF. At each time t, object motion predictions on $t + 1$ are predicted by the KF, which are later used in the OOM construction process and the Fusion R-CNN. More details on this process can be found in the supplementary material.

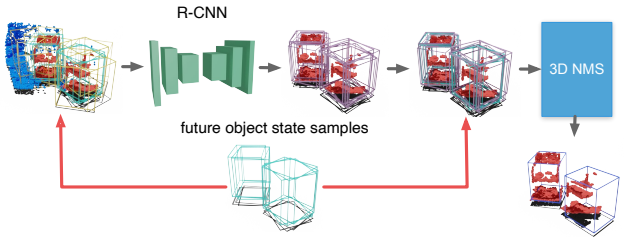

Figure 4. **Fusion R-CNN illustration.** Yellow boxes: current frame proposals ($\{b_{prop}\}$). Green boxes: proposals sampled from back-end object state prediction ($\{b_{pred}\}$). Purple boxes: proposals after R-CNN refinement. Blue boxes: front-end 3D box ($\{b_{front}\}$).

3.5. Discussion

For a standard 3D object detector, each input frame is treated equally as it is from an entirely new scene. It does not make use of the previous detection results of observed objects to achieve more accurate state estimation, while wasting computations on looking for objects in empty regions that are certain to have no object of interest. The proposed UDOLO system attempts to solve these problems by constructing and feeding back the OOM and memorized object future states as spatial-temporal memory, making object detection operate on temporal prior information throughout the entire pipeline. This integration also blurs the boundary between detection and tracking. Thus we consider UDOLO as an integrated detection and tracking system. Notably, we observe a similar trend in the field of 2D object detection and tracking, with examples like Tracktor++ [3] and CenterTrack [64] that directly use modules from object detection for tracking or combine the two tasks in a more coherent way. We believe the idea of early and middle integration proposed in UDOLO is valuable to the community to integrate these two tasks more closely with each other.

4. Experiments

In this section, we conduct a series of experiments to demonstrate the effectiveness of the spatial-temporal memory feedback design in UDOLO and give a detailed ablation study to analyze the importance of different components of our system in Sec. 4.4. We further discuss the motivation behind the experiment design in the supplementary material.

			chair	sofa	bed	stove	table	bathtub	cabinet	trsbin	bkslf	printer	washer	flcab	mAP
Depth	All	VoteNet	41.14	24.59	33.61	22.08	14.87	13.36	9.31	4.90	4.87	3.71	2.86	1.69	14.75
		Ours	**51.63**	**32.65**	**40.32**	**25.84**	**20.04**	**14.38**	**12.19**	**5.66**	**6.52**	**6.02**	**3.46**	**2.40**	**18.43**
		w/o early	51.53	32.17	40.06	25.77	19.94	14.33	12.04	5.50	6.44	6.18	2.96	2.25	18.26
		w/o mid.	41.17	24.89	33.62	22.13	14.94	13.39	9.32	4.91	4.81	3.72	2.85	1.63	14.78
	Occ.	VoteNet	6.67	0.20	0.22	0.50	0.74	1.42	0.11	0.81	0.01	0.01	0.04	0.13	0.91
		Ours	**19.11**	**4.68**	**2.30**	**3.22**	**4.03**	**2.11**	**0.85**	**1.38**	**0.36**	**0.13**	**0.07**	**0.57**	**3.23**
RGB	All	VoteNet	11.30	8.47	18.87	3.14	3.30	2.41	2.32	0.91	0.98	**0.21**	0.31	0.64	4.41
		Ours	**21.92**	**13.75**	**22.89**	**9.69**	**6.09**	**3.30**	**4.23**	**1.12**	**1.98**	0.14	**1.32**	**1.36**	**7.32**
	Occ.	VoteNet	0.47	1.91	10.24	0.58	0.22	1.10	0.26	0.02	0.01	0.01	0.01	0.03	1.24
		Ours	**6.53**	**5.86**	**13.89**	**6.56**	**1.75**	**1.67**	**1.37**	**0.09**	**0.60**	**0.12**	**0.35**	**0.33**	**3.26**

Table 1. **3D object detection results on ScanNet *validation* set.** 12 categories are considered in the experiment, using single frame depth/RGB imagery as input, evaluated with mAP@0.5 IoU (%). Ablation studies of our method without early/middle integration are presented in grey background.

		chair	sofa	bed	stove	table	bathtub	cabinet	trsbin	bkslf	printer	washer	flcab	Overall
Std. Trans.	VoteNet	1.08	1.23	0.33	0.61	1.38	1.11	0.94	0.95	0.18	-	0.43	-	1.07
	Ours	**1.02**	**0.03**	**0.12**	**0.52**	**0.98**	**0.11**	**0.77**	**0.26**	**1.82**	-	**0.05**	-	**0.76**
Std. Rot.	VoteNet	37.88	12.12	18.71	47.22	32.88	38.33	40.28	**2.76**	**2.10**	-	**5.12**	-	37.98
	Ours	**31.82**	**6.54**	**4.44**	**10.45**	**19.97**	**11.83**	**12.66**	6.93	14.09	-	19.78	**0.28**	**17.88**
Std. Dim	VoteNet	0.85	0.21	1.73	0.67	0.73	0.92	0.67	**0.06**	**0.05**	-	**0.08**	-	0.79
	Ours	**0.54**	**0.11**	**0.18**	**0.21**	**0.37**	**0.17**	**0.45**	0.12	0.54	-	0.19	**0.04**	**0.37**

Table 2. **Stability on ScanNet *validation* set** with metrics suggested in [18] using single frame depth input. Lower is better for all the metrics.

4.1. Datasets, Metrics and Baselines

4.1.1 Indoor Environment: ScanNet

The indoor environment has always been challenging for 3D object detectors since indoor objects are usually more diverse in category and also much more cluttered in space. We use the ScanNet [9] dataset to demonstrate the effectiveness of the feedback design of the spatial-temporal memory in UDOLO. Although UDOLO can handle moving objects in indoor scenes, we remove the motion prediction module since all objects in ScanNet are static.

Baseline. Due to the real-time and incremental nature of our approach, the experiment setting is different from most of the baseline methods [23, 40] using this dataset. VoteNet detects axis-aligned bounding boxes defined in the world frame with the complete point cloud of the entire scene as input. We instead use single-view point clouds of each frame as input data and predict oriented bounding boxes in the camera frame, which is more practical in real-time applications. Thanks to the aligned CAD model ground truth annotations from Scan2CAD [1], we train the single-view variant of VoteNet with oriented 3D bounding boxes directly. We further evaluate UDOLO with the point clouds reconstructed from multi-view RGB images by DeepV2D [49] , denoted as RGB in Table 1.

Categories and Metrics. We experiment with 20 classes in Scan2CAD and present 12 classes with AP greater than 1%. In order to demonstrate the ability of occlusion or truncation handling of UDOLO, we create a new difficulty level named "occluded" with strongly occluded or truncated objects only. We use 3D object detection mAP@0.5 IoU as the major metrics. We don't evaluate the MOT metrics considering the objects in ScanNet are static in the world frame. Due to the space constraint, we leave the results on all 20 categories, the results with IoU threshold at 0.25, the detailed ground truth preparation process, and the metrics for bounding box stability to the supplementary material.

4.1.2 Outdoor Environment: KITTI

In order to demonstrate that UDOLO can also be applied to dynamic scenes, we further evaluate it on the KITTI Tracking and KITTI Object datasets. For stereo images input, the point cloud input for the 3D detector is converted from the estimated disparity following the method proposed in *pseudo-lidar* [51]. The Multi-Object Tracking (MOT) performance is measured with the metrics proposed in AB3DMOT [52], which are 3D variants of the CLEAR metrics [4]. We also consider AB3DMOT as the baseline method for LiDAR input since we share the same 3D detector (PointRCNN), association method (IoU-based association) and state fusion method (Kalman Filter). We further compare our method with [32] and [31] that use sequential stereo images as input. The training/validation splits used in all the experiments are made identical with the corresponding baselines to maintain a fair comparison. The 2D MOT results on the test set of the public benchmark are presented in the supplementary material.

Method	Input	3D IoU = 0.25		3D IoU = 0.5		3D IoU = 0.7	
		(sA/A/)MOTA	(A)MOTP	(sA/A/)MOTA	(A)MOTP	(sA/A/)MOTA	(A)MOTP
JT [24]	Mono	- / - / -15.6	- / 47.9	- / - / -55.6	- / 63.8	-	-
ST [31]	Stereo	- / - / 3.3	- / 51.7	- / - / -34.1	- / 65.4	-	-
JST [32]	Stereo	- / - / **56.7**	- / 62.3	- / - / 29.4	- / 69.4	-	-
Ours	Stereo	63.8/22.4/53.7	53.1/**71.7**	54.8/16.9/**45.5**	48.4/**74.4**	26.8/4.3/20.0	32.4/80.6
mmMOT [62]	LiDAR	70.6/33.1/74.1	72.5/78.2	69.1/32.8/73.5	72.2/78.5	63.9/24.9/51.9	67.3/80.7
FANTrack [2]	LiDAR	83.0/40.0/74.3	75.0/75.2	80.1/38.2/72.7	73.6/74.9	62.7/24.7/49.2	66.1/79.0
AB3DMOT	LiDAR	93.3/45.4/86.2	77.4/78.4	90.4/42.8/84.0	75.7/79.0	69.8/27.3/57.1	67.0/82.4
GNN3D [53]	LiDAR	93.9/45.8/86.0	78.1/79.0	- / - / -	- / -	- / - / -	- / -
GNNTrkF [55]	LiDAR	94.4/46.2/86.9	76.8/78.3	- / - / -	- / -	- / - / -	- / -
Ours	LiDAR	**94.9/47.4/87.9**	**80.7/79.8**	**92.7/45.2/86.2**	**78.9/80.0**	**75.7/31.0/65.8**	**69.7/82.6**

Table 3. **3D multi-object tracking results on the KITTI Tracking *validation* set.** We report 3D MOT metrics of the **car** category. *(sA/A)MOTA* stands for *sAMOTA/AMOTA/MOTA* and *(A)MOTP* stands for *AMOTP/MOTP*. The evaluation is conducted in the 3D space using the 3D MOT evaluation tool provided in [52]. The 3D IoU is the threshold value used for True Positive (TP) assignments. Higher is better for all the metrics.

Method	Input	AP_{3d}(IoU=0.5)			AP_{3d}(IoU=0.7)		
		Easy	Mod.	Hard	Easy	Mod.	Hard
PL(AVOD)	Stereo	88.50	**76.40**	61.20	61.90	45.30	39.00
PL(F-PointNet)	Stereo	88.60	72.80	63.90	54.90	36.40	31.10
PL(PointRCNN)	Stereo	89.01	74.36	67.11	62.51	45.78	39.50
Ours	Stereo	**89.32**	75.43	**67.42**	**70.27**	**48.11**	**41.05**

Table 4. **3D object detection results on the KITTI Object *validation* set.** We report AP_{3D}(in%) of the **car** category. PL stands for *pseudo-lidar* [60].

Method	Type	AP_{3d}(IoU=0.25)			AP_{3d}(IoU=0.5)		
		Easy	Moderate	Hard	Easy	Moderate	Hard
3DOP [6]	single-frame	80.62	70.01	65.76	53.73	42.27	35.87
ST [31]	multi-frame	86.57	74.13	**68.96**	48.51	37.13	34.54
Ours	multi-frame	**86.69**	**75.71**	68.09	**86.45**	**67.49**	**59.66**

Table 5. **3D object detection results on the KITTI tracking *validation* set.** We report AP_{3D}(in%) of the car category using stereo images as input, evaluated with IoU thresholds of 0.25 and 0.5. The results for 3DOP come from [31].

4.2. Implementation Details

In principle, the proposed framework works with all kinds of two-stage 3D object detectors. For indoor scenes, we use VoteNet as the base detector and consider the point cloud "voting, sampling and grouping" process in VoteNet as RPN and the bounding box regression module as R-CNN. For outdoor scenes, we use PointRCNN [47] as the base detector, where the "Bin-based 3D Box Generation" module is considered as RPN and the "Bin-based 3D Box Refinement" as R-CNN. During the training process on ScanNet and KITTI, we use the same hyperparameters as the original implementation. Because of the flexibility of the point cloud backbone, it is not necessary to finetune the networks to adapt to the different modalities of the input point cloud during inference within UDOLO. Since it is impractical to construct an OOM that covers the entire outdoor scene on KITTI, we construct a moving OOM in front of the camera which moves with the vehicle. Specifically, the area of the OOM is set to $8m \times 8m$ on ScanNet and $25m \times 15m$ on KITTI, with the voxel size set to $0.04m \times 0.04m$ and $0.5m \times 0.5m$, respectively. The threshold σ in the OOM-Guided RPN is set to 3 on ScanNet and 8 on KITTI. The scale parameter λ is set to 100. We use 50 current frame proposals and 50 back-end future state samples in the Fusion R-CNN. The camera ego-motion is obtained from BundleFusion [10] on ScanNet and VINS-Fusion [42] on KITTI.

4.3. Evaluation Results

3D object detection performance on ScanNet. The result on ScanNet is given in Table 1. Our method also produces much better performance compared to the single-frame detection baseline (the reproduced single-view variant of VoteNet). In particular, on the "occluded" split, the improvement in mAP is up to 250% in the ratio (from 0.91 to 3.23). The experiments with point clouds reconstructed from multi-view RGB images also show consistent improvements. The significant improvements benefit from UDOLO's ability to handle severe truncation and occlusion in the cluttered indoor scene. The feedback of the bounding boxes from the back-end to the front-end enables the system to memorize previously detected objects and reuse this information in the current time step.

Stability evaluation on ScanNet. The results are reported in Table 2. Cells without numbers mean no stable bounding box predictions in any two adjacent frames. With the strong prior provided by the spatial-temporal memory, our method is more stable than the single-frame detection baseline. The improvement in bounding box stability can be also observed in the supplementary video.

3D MOT performance on KITTI Tracking. The results on the validation and test set of KITTI Tracking are summarized in Table 3. In terms of 3D MOT results on the

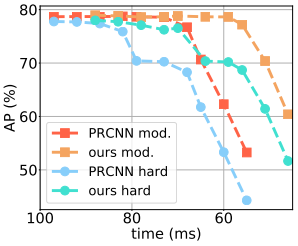

Figure 5. **Trade-off between speed and accuracy.** Speed is measured by runtime per frame (in *ms*) and accuracy is measured by AP_{3D} (in %) at IoU=0.7 for **car**. The curve that is more closer to the top-right corner is better.

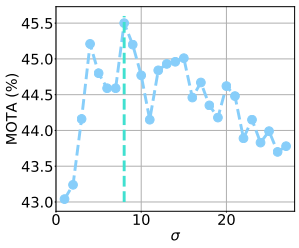

Figure 6. **Ablation study** on the OOM threshold parameter σ on the KITTI tracking set with stereo input. We report *MOTA* (in %) of the **car** category evaluated in the 3D space. $\sigma = 8$ is used in our experiments.

validation set, our method outperforms the corresponding baselines that share the same input by a large margin, especially under the 0.5 IoU threshold. We observe that our method has a substantially lower amount of false positives compared with baseline method JST, thus leading to large improvements on MOTA. For results with LiDAR input, our method surpasses the baseline method AB3DMOT with most of the metrics. Since AB3DMOT shares most of the same modules used in UDOLO (namely the 3D detector, Kalman Filter, and motion prediction), the improvements validate the effectiveness of the feedback of spatial-temporal memory.

3D object detection performance on KITTI Object and KITTI Tracking. The results are summarized in Table 4 and 5. For 3D object detection performance, as shown in the tables, our system surpasses the baseline methods (PL(PointRCNN) and ST) by a substantial margin with the detection metrics. Since PL(PointRCNN) is used as the base detector for UDOLO, the comparison between Ours and PL(PointRCNN) demonstrates that UDOLO also improves the performance with the detection metrics.

4.4. Ablation Study

Feedback design (effectiveness of middle integration). The results are presented in Table 1 with background colored in grey. We validate the spatial-temporal memory feedback design by separately removing the early (titled *w/o early.*) and middle integration (*w/o mid.*). The rest of the system remains unchanged. Comparing to the full system (Ours), the performance drops significantly in experiments without middle integration design (*w/o mid.*). The results *w/o early.* further demonstrate that although input points have been removed by the early integration design, the results do not decrease significantly. As shown later, the early integration design increases runtime speed without losing detection precision. These results demonstrate that the improvements in detection precision are mainly contributed by the middle integration. Particularly on the "chair" category, the AP suffers a severe drop of 10.5% (21% in the ratio) when the middle integration is removed.

Runtime speed (effectiveness of early integration). As explained in Sec. 3.2, the early integration design in the OOM-Guided RPN will save runtime on both the backbone and the RPN since less amount of point cloud is passed through. We analyze the runtime speed up in the backbone and RPN brought by the early integration on KITTI Object with PointRCNN. The runtime is reduced from 42ms to 32ms, with an improvement of 23.8%. We give an analysis of the trade-off between speed and accuracy in UDOLO with the plot in Fig. 5. The trade-off is achieved by choosing a different number of the overall object proposals, starting from 100 to 10 with a step size of 10. For our method, both $\{b_{prop}\}$ and $\{b_{pred}\}$ are used as proposals and the number ratio between them is kept fixed. The experiment is performed on KITTI Object with LiDAR data input. The plot reveals the effectiveness of our spatial-temporal memory feedback design on runtime speed. Our system can retain the precision of prediction even when the number of proposals is reduced to half of the original number, while the precision of single-frame detection baseline rapidly decreases.

Effects on different OOM threshold value σ. As shown in Fig. 6, the AMOTA score [52] first increases and then decreases with the increasing of σ. With a smaller value of σ, a larger part of point cloud will be removed even the number of observations at the corresponding location is not enough, thus decreasing the overall performance. When the threshold gets larger than 20, most of the point cloud is passed through the network since there are few locations that have a number of observations more than the threshold, and the OOM-Guided RPN degrades to the vanilla version of RPN that takes the entire point cloud as input.

5. Conclusion

In this paper, we introduced UDOLO, an integrated detection and tracking system that incorporates object occupancy map and object states as spatial-temporal memory and feeds this memory all the way through the detection pipeline to improve efficiency and accuracy of 3D detection. We demonstrated the effectiveness of the proposed feedback design in the experiments, which surpassed baseline methods in both 3D multi-object tracking and single-frame 3D object detection on outdoor and indoor datasets.

Acknowledgement. The authors would like to acknowledge the support from the National Key Research and Development Program of China (No. 2020AAA0108901), NSFC (No. 61822310), and ZJU-SenseTime Joint Lab of 3D Vision.

References

[1] Armen Avetisyan, Manuel Dahnert, Angela Dai, Manolis Savva, Angel X. Chang, and Matthias Nießner. Scan2CAD: Learning CAD Model Alignment in RGB-D Scans. *CVPR*, 2019.

[2] Erkan Baser, Venkateshwaran Balasubramanian, Prarthana Bhattacharyya, and Krzysztof Czarnecki. FANTrack: 3D Multi-Object Tracking with Feature Association Network. *2019 IEEE Intelligent Vehicles Symposium (IV)*, 2019.

[3] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. In *ICCV*, 2019.

[4] Keni Bernardin and Rainer Stiefelhagen. Evaluating Multiple Object Tracking Performance: The CLEAR MOT Metrics. *EURASIP Journal on Image and Video Processing*, 2008, 2008.

[5] Garrick Brazil, Gerard Pons-Moll, Xiaoming Liu, and Bernt Schiele. Kinematic 3d object detection in monocular video. *ECCV*, 2020.

[6] Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Andrew Berneshawi, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 3D Object Proposals for Accurate Object Class Detection. In *Neural Information Processing Systems (NIPS)*, 2015.

[7] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3D Object Detection Network for Autonomous Driving. In *CVPR*, 2017.

[8] Hsu-kuang Chiu, Antonio Prioletti, Jie Li, and Jeannette Bohg. Probabilistic 3d multi-object tracking for autonomous driving. *arXiv*, 2020.

[9] Angela Dai, Angel X. Chang, Manolis Savva, Maciej Halber, Thomas Funkhouser, and Matthias Nießner. CVPR: Richly-annotated 3D Reconstructions of Indoor Scenes. *CVPR*, 2017.

[10] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. BundleFusion: Real-time Globally Consistent 3D Reconstruction using On-the-fly Surface Re-integration. *TOGS*, 2017.

[11] Julie Dequaire, Peter Ondruska, Dushyant Rao, Dominic Wang, and Ingmar Posner. Deep tracking in the wild: End-to-end tracking using recurrent neural networks. *The International Journal of Robotics Research*, 37, 2017.

[12] Jingming Dong, Xiaohan Fei, and Stefano Soatto. Visual-Inertial-Semantic Scene Representation for 3D Object Detection. In *CVPR*, 2017.

[13] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6), 1989.

[14] Nico Engel, Stefan Hoermann, Philipp Henzler, and Klaus Dietmayer. Deep object tracking on dynamic occupancy grid maps using rnns. 2018.

[15] Xiaohan Fei and Stefano Soatto. Visual-Inertial Object Detection and Mapping. *ECCV*, 2018.

[16] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Detect to Track and Track to Detect. *ICCV*, 2017.

[17] Davi Frossard and Raquel Urtasun. End-to-end Learning of Multi-sensor 3D Tracking by Detection. *ICRA*, 2018.

[18] Mathieu Garon, Denis Laurendeau, and Jean-François Lalonde. A Framework for Evaluating 6-DOF Object Trackers. *ECCV*, 2018.

[19] Silvio Giancola, Jesus Zarzar, and Bernard Ghanem. Leveraging Shape Completion for 3D Siamese Tracking. *CVPR*, 2019.

[20] David Ha and Jürgen Schmidhuber. Recurrent World Models Facilitate Policy Evolution. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems 31*. Curran Associates, Inc., 2018.

[21] David Held, Jesse Levinson, Sebastian Thrun, and Silvio Savarese. Robust real-time tracking combining 3D shape, color, and motion. *The International Journal of Robotics Research*, 35(1-3), 2016.

[22] S. Hoermann, P. Henzler, M. Bach, and K. Dietmayer. Object detection on dynamic occupancy grid maps using deep learning and automatic label generation. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, 2018.

[23] Ji Hou, Angela Dai, and Matthias Nießner. 3D-SIS: 3D Semantic Instance Segmentation of RGB-D Scans. *CVPR*, 2019.

[24] Hou-Ning Hu, Qi-Zhi Cai, Dequan Wang, Ji Lin, Min Sun, Philipp Krähenbühl, Trevor Darrell, and Fisher Yu. Joint Monocular 3D Vehicle Detection and Tracking. *ICCV*, 2019.

[25] Peiyun Hu, Jason Ziglar, David Held, and Deva Ramanan. What you see is what you get: Exploiting visibility for 3d object detection. In *CVPR*, 2020.

[26] Rui Huang, Wanyue Zhang, Abhijit Kundu, Caroline Pantofaru, David A. Ross, Thomas A. Funkhouser, and Alireza Fathi. An LSTM approach to temporal 3d object detection in lidar point clouds. *ECCV*, 2020.

[27] Kai Kang, Hongsheng Li, Tong Xiao, Wanli Ouyang, Junjie Yan, Xihui Liu, and Xiaogang Wang. Object Detection in Videos with Tubelet Proposal Networks. *CVPR*, 2017.

[28] Kai Kang, Wanli Ouyang, Hongsheng Li, and Xiaogang Wang. Object Detection from Video Tubelets with Convolutional Neural Networks. In *CVPR*, 2016.

[29] H. W. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2(1-2), 1955.

[30] Bo Li, Junjie Yan, Wei Wu, Zheng Zhu, and Xiaolin Hu. High performance visual tracking with siamese region proposal network. In *CVPR*, 2018.

[31] Peiliang Li, Tong Qin, and Shaojie Shen. Stereo Vision-based Semantic 3D Object and Ego-motion Tracking for Autonomous Driving. *ECCV*, 2018.

[32] Peiliang Li, Jieqi Shi, and Shaojie Shen. Joint Spatial-Temporal Optimization for Stereo 3D Object Tracking. *CVPR*, 2020.

[33] Ming Liang, Bin Yang, Wenyuan Zeng, Yun Chen, Rui Hu, Sergio Casas, and Raquel Urtasun. Pnpnet: End-to-end perception and prediction with tracking in the loop. In *CVPR*, 2020.

[34] Wenjie Luo, Bin Yang, and Raquel Urtasun. Fast and Furious: Real Time End-to-End 3D Detection, Tracking and Motion Forecasting with a Single Convolutional Net. In *CVPR*, 2018.

[35] Richard A Newcombe, Andrew J Davison, Shahram Izadi, Pushmeet Kohli, Otmar Hilliges, Jamie Shotton, David Molyneaux, Steve Hodges, David Kim, and Andrew Fitzgibbon. KinectFusion: Real-Time Dense Surface Mapping and

Tracking. *Proceedings of the 24th annual ACM symposium on User interface software and technology*, 2011.

[36] Dominik Nuss, Stephan Reuter, Markus Thom, Ting Yuan, Gunther Krehl, Michael Maile, Axel Gern, and Klaus Dietmayer. A random finite set approach for dynamic occupancy grid maps with real-time application. *The International Journal of Robotics Research*, 2016.

[37] Peter Ondruska and Ingmar Posner. Deep tracking: Seeing beyond seeing using recurrent neural networks. *AAAI*, 2016.

[38] Maria Chiara Passolunghi and Irene C. Mammarella. Spatial and visual working memory ability in children with difficulties in arithmetic word problem solving. *European Journal of Cognitive Psychology*, 22(6), 2010.

[39] Charles R Qi, Xinlei Chen, Or Litany, and Leonidas J Guibas. Imvotenet: Boosting 3d object detection in point clouds with image votes. In *CVPR*, 2020.

[40] Charles R. Qi, Or Litany, Kaiming He, and Leonidas J. Guibas. Deep Hough Voting for 3D Object Detection in Point Clouds. *ICCV*, 2019.

[41] Rui Qian, Divyansh Garg, Yan Wang, Yurong You, Serge Belongie, Bharath Hariharan, Mark Campbell, Kilian Q Weinberger, and Wei-Lun Chao. End-to-end pseudo-lidar for image-based 3d object detection. In *CVPR*, 2020.

[42] Tong Qin, Jie Pan, Shaozu Cao, and Shaojie Shen. A General Optimization-based Framework for Local Odometry Estimation with Multiple Sensors. *ArXiv*, 2019.

[43] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2015.

[44] Sarthak Sharma, Junaid Ahmed Ansari, J. Krishna Murthy, and K. Madhava Krishna. Beyond Pixels: Leveraging Geometry and Shape Cues for Online Multi-Object Tracking. *ICRA*, 2018.

[45] Abhijeet Shenoi, Mihir Patel, JunYoung Gwak, Patrick Goebel, Amir Sadeghian, Hamid Rezatofighi, Roberto Martin Martin, and Silvio Savarese. JRMOT: A real-time 3d multi-object tracker and a new large-scale dataset. *arXiv*, 2020.

[46] Shaoshuai Shi, Chaoxu Guo, Li Jiang, Zhe Wang, Jianping Shi, Xiaogang Wang, and Hongsheng Li. Pv-rcnn: Point-voxel feature set abstraction for 3d object detection. In *CVPR*, 2020.

[47] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud. *CVPR*, 2019.

[48] Shuran Song and Jianxiong Xiao. Sliding Shapes for 3D Object Detection in Depth Images. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *ECCV*. 2014.

[49] Zachary Teed and Jia Deng. DeepV2D: Video to depth with differentiable structure from motion. *ICLR*, 2020.

[50] Virve Vuontela, Maija-Riikka Steenari, Synnöve Carlson, Juha Koivisto, Mika Fjällberg, and Eeva T. Aronen. Audiospatial and Visuospatial Working Memory in 6–13 Year Old School Children. *Learning & Memory*, 10(1), 2003.

[51] Yan Wang, Wei-Lun Chao, Divyansh Garg, Bharath Hariharan, Mark Campbell, and Kilian Q. Weinberger. Pseudo-LiDAR from Visual Depth Estimation: Bridging the Gap in 3D Object Detection for Autonomous Driving. *CVPR*, 2019.

[52] Xinshuo Weng, Jianren Wang, David Held, and Kris Kitani. 3D Multi-Object Tracking: A Baseline and New Evaluation Metrics. *IROS*, 2020.

[53] Xinshuo Weng, Yongxin Wang, Yunze Man, and Kris Kitani. Graph Neural Network for 3D Multi-Object Tracking. *ECCVW*, 2020.

[54] Xinshuo Weng, Yongxin Wang, Yunze Man, and Kris M. Kitani. Gnn3dmot: Graph neural network for 3d multi-object tracking with 2d-3d multi-feature learning. In *CVPR*, 2020.

[55] Xinshuo Weng, Ye Yuan, and Kris Kitani. End-to-End 3D Multi-Object Tracking and Trajectory Forecasting. *ECCVW*, 2020.

[56] Fanyi Xiao and Yong Jae Lee. Video Object Detection with an Aligned Spatial-Temporal Memory. *ECCV*, 2018.

[57] Bin Xu and Zhenzhong Chen. Multi-level Fusion Based 3D Object Detection from Monocular Images. In *CVPR*, 2018.

[58] Shichao Yang and Sebastian Scherer. CubeSLAM: Monocular 3D Object SLAM. *IEEE Transactions on Robotics*, 2019.

[59] Junbo Yin, Jianbing Shen, Chenye Guan, Dingfu Zhou, and Ruigang Yang. Lidar-based online 3d video object detection with graph-based message passing and spatiotemporal transformer attention. In *CVPR*, 2020.

[60] Yurong You, Yan Wang, Wei-Lun Chao, Divyansh Garg, Geoff Pleiss, Bharath Hariharan, Mark Campbell, and Kilian Q. Weinberger. Pseudo-LiDAR++: Accurate Depth for 3D Object Detection in Autonomous Driving. *ICLR*, 2020.

[61] Jesus Zarzar, Silvio Giancola, and Bernard Ghanem. Efficient tracking proposals using 2d-3d siamese networks on lidar. *arXiv preprint arXiv:1903.10168*, 2019.

[62] Wenwei Zhang, Hui Zhou, Shuyang Sun, Zhe Wang, Jianping Shi, and Chen Change Loy. Robust multi-modality multi-object tracking. In *ICCV*, 2019.

[63] Zheng Zhang, Dazhi Cheng, Xizhou Zhu, Stephen Lin, and Jifeng Dai. Integrated Object Detection and Tracking with Tracklet-Conditioned Detection. *ArXiv*, 2018.

[64] Xingyi Zhou, Vladlen Koltun, and Philipp Krähenbühl. Tracking objects as points. *arXiv:2004.01177*, 2020.

[65] Xizhou Zhu, Jifeng Dai, Lu Yuan, and Yichen Wei. Towards High Performance Video Object Detection. *CVPR*, 2018.

[66] Xizhou Zhu, Yujie Wang, Jifeng Dai, Lu Yuan, and Yichen Wei. Flow-Guided Feature Aggregation for Video Object Detection. *ICCV*, 2017.