Joint Spatial-Temporal Optimization for Stereo 3D Object Tracking

Peiliang Li, Jieqi Shi, and Shaojie Shen
The Hong Kong University of Science and Technology
pliap@connect.ust.hk, jshias@connect.ust.hk, eeshaojie@ust.hk

Abstract

Directly learning multiple 3D objects motion from sequential images is difficult, while the geometric bundle adjustment lacks the ability to localize the invisible object centroid. To benefit from both the powerful object understanding skill from deep neural network meanwhile tackle precise geometry modeling for consistent trajectory estimation, we propose a joint spatial-temporal optimization-based stereo 3D object tracking method. From the network, we detect corresponding 2D bounding boxes on adjacent images and regress an initial 3D bounding box. Dense object cues (local depth and local coordinates) that associating to the object centroid are then predicted using a region-based network. Considering both the instant localization accuracy and motion consistency, our optimization models the relations between the object centroid and observed cues into a joint spatial-temporal error function. All historic cues will be summarized to contribute to the current estimation by a per-frame marginalization strategy without repeated computation. Quantitative evaluation on the KITTI tracking dataset shows our approach outperforms previous image-based 3D tracking methods by significant margins. We also report extensive results on multiple categories and larger datasets (KITTI raw and Argoverse Tracking) for future benchmarking.

1. Introduction

3D object detection and tracking play a significant role for autonomous driving vehicles where the time-independent detection undertakes the fundamental perception, and continuous object tracking further enables temporal motion prediction and planning. With the rapid evolution of 3D deep learning and feature representation, the detection part has been made great progress in terms of 3D localization ability by many efforts [49, 19, 31, 23, 46, 8, 35, 52, 28, 27, 21, 42]. However, as an equally essential task, the 3D object tracking has rarely been explored. Only a few recent works [25, 30, 16] demonstrate 3D object tracking ability in the context of self-driving scenarios. To bridge this gap and take advantage of sequential visual cues, we aim at a complete 3D object tracking system that joint exploits spatial-temporal information and estimates accurate 3D object trajectories with motion consistency. We focus on the use of stereo cameras as it shows a promising balance between the cost and 3D sensing ability comparing with the expensive LiDAR sensor and the inadequate single camera.

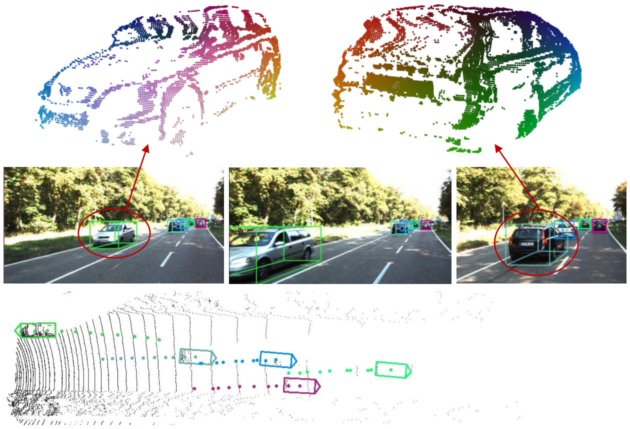

Figure 1. **An example of our 3D tracking system.** From top to bottom: The sampled object local depth which is color mapped by local coordinates; 3D tracking result on sequential images; 3D tracking result on the bird's eye view. Here the trajectory is transformed to global coordinates for visualization using the off-shelf ego-camera pose.

In this paper, we consider the 3D object tracking as not only a *data association* problem but also a *continuous state estimation* problem, where the estimation result should satisfy both instant spatial constraints and the accumulated history evidence. To this end, we propose our joint spatial-temporal optimization-based 3D object tracking system. As illustrated in Fig. 2, our system firstly generates paired region proposals on the concatenated current and previous feature maps. After *RoIAlign* [14], we employ three parallel branches on the concatenated RoI (region of interest) feature to refine the proposal and generate object-level and pixel-level information. As Fig. 2 shows, the paired regression branch refines the paired proposals to accurate 2D bounding box pairs. Benefit from the paired detection, the

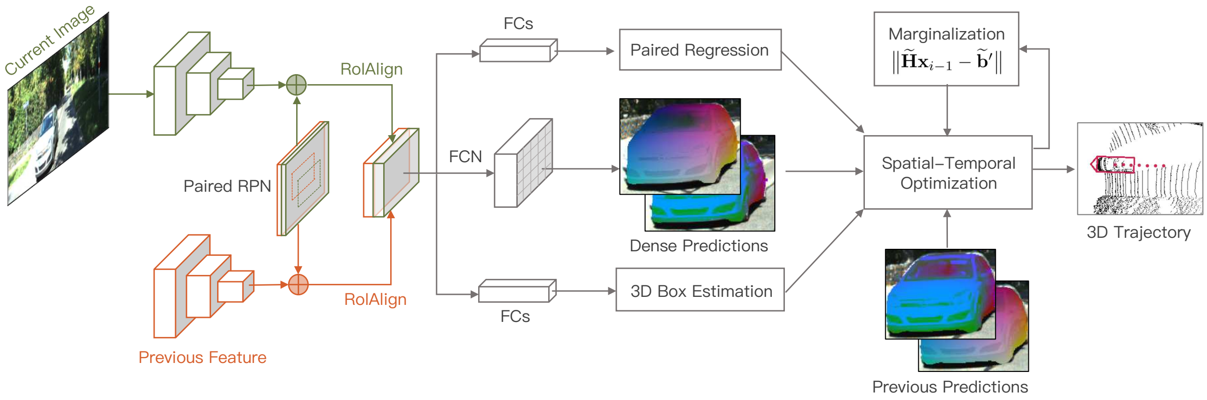

Figure 2. Architecture of the proposed Stereo 3D object tracking system, which generates paired 2D boxes for data association (Sect. 3.1), initial 3D box estimation (Sect. 3.2) and dense local predictions (Sect. 3.3) for the following spatial-temporal optimization (Sect. 4).

sequential objects can be naturally associated without additional similarity computation (Sect. 3.1). The 3D estimation branch predicts object-level information, e.g., centroid projection, observation angle to form an initial 3D box (Sect. 3.2). The dense prediction branch outputs pixel-level segmentation and local geometry cues such as local depth and local coordinates (Sect. 3.3) that are aggregated later to formulate our spatial-temporal optimization.

To estimate a consistent and accurate 3D trajectory, we enforce precise geometry modeling by jointly considering the dense spatial and historic cues. From the spatial view, an optimal object depth should yield minimal stereo photometric error given the local depth relations between foreground pixels and object centroid. From the temporal view, the consistent object motion will yield minimal reprojection error after warping foreground pixels to the adjacent frame. Based on this, we propose a joint spatial-temporal optimization which models all these dense constraints in a tightly-coupled manner (Sect. 4). To trade off the large amount information introduced by dense cues from multiple frames, we further introduce a per-frame marginalization strategy where the previous observations will be iteratively marginalized as a linear prior, s.t., all historical evidence will naturally contribute to the current object estimation without the need of information reuse.

Overall, our main contributions can be summarized as:

- A complete 3D object tracking framework that handles simultaneous detection & association via learned correspondences, and solves continuous estimation by fully exploiting dense spatial-temporal constraints.

- Significantly outperform state-of-the-art image-based 3D tracking methods on the KITTI tracking dataset.

- Report extensive evaluation on more categories and larger-scale datasets (KITTI Raw and Argoverse Tracking) to benefit future benchmarking.

2. Related Work

3D Object Detection. There are plenty of research efforts focus on the detecting 3D object using instant sensor data in autonomous driving scenarios. From the modality of input data, we can roughly outline them into three categories: monocular image-based methods, stereo imagery-based, and LiDAR-based methods. Given a **monocular image**, some earlier works [6, 51, 4, 20] exploit multiple levels of information such as segmentation, shape prior, keypoint, and instance model to help the 3D object understanding, while recent state-of-the-art works [49, 38, 19, 31, 3] pay more attention to the depth information encoding from different aspects to detect and localize the 3D object. Adding additional images with known extrinsic configuration, **stereo based methods** [7, 23, 46, 34] demonstrate much better 3D object localization accuracy, where [23] utilizes object-level prior and geometric constraints to solves the object pose using raw stereo image alignment. [46] converts the stereo-generate depth to a pseudo point cloud representation and directly detect object in 3D space. While [34] takes advantages of both and predict object-level point cloud then regress the 3D bounding box based on the object point cloud. Besides the image-based approaches, rich works [22, 10, 52, 21, 50, 35, 42] utilize the direct 3D information from the **LiDAR point cloud** to detect 3D objects, where [10, 52, 21] samples the unstructured point cloud into structured voxel representation and use 2D or 3D convolution to encode features. [22, 50] project the point cloud to the front or bird's eye views such that the 3D object detection can be achieved by regular 2D convolutional networks. From another aspect, [35, 42] directly localize 3D objects in unstructured point cloud with the aid of PointNet [36] encoder. Furthermore, [8, 18, 28, 27] exploit fuse the image and point cloud in feature level to enable multi-modality detection and scene understanding.

3D Object Tracking. Although extensive object tracking approaches have been studied in recent decades, in this paper, we mainly discuss the most relevant literature: the 3D object tracking. Based on the 3D detection results, [33, 41, 43] employ a filter based modeling (multi-Bernoulli, Kalman filter) to track the 3D object continuously. Alternatively, [16] directly learns the object motion using an LSTM model by taking advantage of data-driving approaches. However, decoupling the detection and tracking might cause a sub-optimal solution due to the information loss. Benefit from the stereo vision, [11] focuses on the object reconstruction with continuous tracked visual cues, and [26] employ an object bundle adjustment approach to solve consistent object motion in a sliding window, while relying on the sparse feature matching and loosely coupling the stereo reconstruction with temporal motion limits its performance on 3D localization for occluded and far-away objects. In another way, [30] encodes sequential 3D point cloud into a concatenated voxel representation, and directly produces associated multi-frame 3D object detections as tracked trajectory together with motion forecasting.

3. Sequential 3D Tracking Network

In this section, we describe our sequential 3D object tracking network, which simultaneously detects and associates objects in consecutive monocular images, and introduce our network predictions to enable the initial 3D box estimation and the subsequent joint optimization (Sect. 4).

3.1. Simultaneous Detection & Association

To avoid additional pair-wise similarity computation for object association, we leverage the network directly detect the corresponding objects in adjacent frames. Before object-wise estimation, we use the region proposal network (RPN) [39] to densely classify the foreground object and predict coarse object region on the feature map. Inspired from [23] for stereo detection, we extend the RPN to recognize the union area of where the object appearing in two sequential images. Specifically, After feature extraction, the feature maps of current image and the previous image are concatenated to involve temporal information (see Fig. 2). We pre-calculate the union of corresponding object boxes in the current and previous images. On the concatenated feature map with per-location defined anchors, an anchor will be classified as the foreground only if its IoU (intersection-over-union) with one of the union box is larger than 0.7. On this definition, the positive anchor will cover the object area on both images, thereby it can be regressed to paired RoIs proposals on the current and previous image respectively. Note that this paired RPN does not beyond the network capability since it can be thought as a special classification task where only a repeated and motion-reasonable pattern with same instances can be recognized as a positive sample.

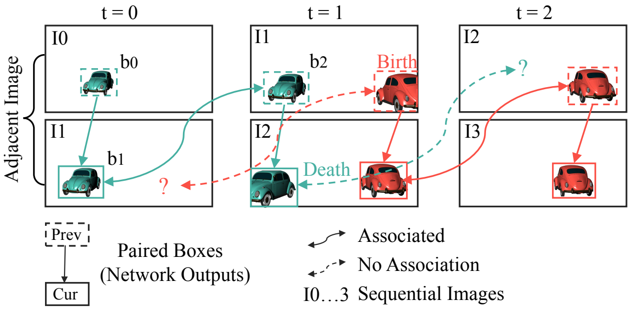

Figure 3. **Association illustration**. At t0 timestamp, the network predicts paired 2D box (b0, b1) for image I0 and I1 respectively. Then at time T1, the green car can be associated by comparing 2D IoU of b1 and b2. The *Birth* and *Death* represent the newborn trackers and died trackers respectively.

The coarse proposal pairs are further refined in the paired regression. As Fig. 2 shows, we use the RoI pairs to perform *RoIAlign* [14] on the current and previous feature maps respectively. The cropped current and previous RoI features are then concatenated again to enable the R-CNN based 2D box refinement. By predicting two sets of box offset $[\Delta x, \Delta y, \Delta w, \Delta h]$ which denote offsets in x, y direction, width and height, we obtain paired 2D boxes for current and previous images at each timestamp. During inference, we associate sequential trackers by comparing the 2D IoU between the previous box and current's previous box. An example is visualized in Fig. 3 for better illustration.

Benefited from this simple design, we achieve simultaneous object detection and association with almost no additional computation, and avoid being affected by large motion as the neural network can find the correspondences around large receptive field.

3.2. 3D Object Estimation

A complete 3D bounding box is parameterized by $[x, y, z, w, h, l, \theta]$, where x, y, z are the 3D centroid position respecting to the camera frame, w, h, l the 3D dimension, and θ the horizontal orientation. Since the global location information is lost after crop and resize operation in *RoIAlign* [14], we predict several local geometric properties (centroid projection, depth, observation angle α) to form a initial 3D bounding box. The centroid projection is defined as the projection coordinates of the 3D object centroid on the image, which can be learnt from the offset between the projection center and the RoI center. For dimension and depth, we predict a residual term based on a dimension prior and an inferred coarse depth $f\frac{h_{3d}}{h_{2d}}$, given by the focal length f, 3D object height h_{3d}, and 2D RoI height h_{2d}. The observation angle represents the object local viewpoint, which can be learnt from the deformed RoI pattern. Note that the observation angle α is not equivalent to the object orientation θ, instead holds the relation: $\alpha = \theta + \arctan\frac{x}{z}$, as proved in [32, 23].

3.3. Dense Prediction

However, the predicted 3D box is far from enough for a consistent and accurate 3D tracker as it does not explicitly utilize spatial nor temporal information, we thus define essential dense representations to enable our following joint spatial-temporal optimization.

Mask: We use a stacked region-based FCN layers [14] to predict dense object mask on the RoI feature maps, which is used for our foreground pixel selection.

Local Depth: For the foreground pixel, we define the local depth δ, given by the depth difference between the pixels and the object centroid, which are integrated later for constraining the object centroid depth using stereo alignment.

Local Coordinates: We predict each pixel's 3D local coordinates respecting to the object frame as also used in [45, 24]. On this representation, the same part of the object holds a unique coordinate which is invariant with object translation and rotation across time, therefore it can be used as the geometric descriptor to obtain dense pixel correspondences between sequential object patches. Comparing to traditional descriptor such as ORB [40], our learned local coordinate is a unique representation in object domain and robust to perspective-changing, textureless, and illumination variance, thereby give us robust and dense pixel correspondences even for high occluded and faraway objects.

4. Joint Spatial-Temporal Optimization

Based on these network predictions, we introduce our joint spatial-temporal optimization model. For simplicity, we consider a single object in the following formulation since we solve all objects analogously in a parallel way. Let I_i^l, I_i^r be the sequential stereo image where i denote the frame index, $\mathbf{c}_i$, α_i be the predicted object centroid projection and observation angle respectively. Let $\mathbf{u}_i$ be the observed foreground pixels given by the object mask. For each observed pixel, we have the local depth δ_i which serves as spatial cues for stereo alignment, local coordinates C_i that serve our temporal cues for pixel association. For each object, we aim to estimate an accurate object position $\mathbf{p}_i = \{x, y, z\}$ and rotation $\mathbf{R}_i(\theta)$ respecting to the instant camera frame, s.t., we have overall minimum spatial alignment errors and meanwhile are best fitted with the previous pixel observation across multiple frames.

Spatial Alignment. The spatial alignment cost is defined as the photometric error between left-right images:

$$\mathbf{E}_{si} := \sum_{\mathbf{u}_i \in \mathcal{N}_s} w_I \left\| I_i^l(\mathbf{u}_i) - I_i^r(\mathbf{u}_i^r) \right\|_h, \tag{1}$$

where $\mathcal{N}_s$ is the set of sampled foreground pixels according to the image gradient, w_I the weight of the photometric error, $\|\cdot\|_h$ the Huber norm. $\mathbf{u}_r^i$ represents the warped pixel location on the right image I_i^r, given by

$$\mathbf{u}_i^r = \pi\big(\pi^{-1}(\mathbf{u}_i, \delta_i + \mathbf{p}_i^z) + \mathbf{p}_s\big) \tag{2}$$

where we use $\pi(\mathbf{p}) : \mathbb{R}^3 \to \mathbb{R}^2$ to denote projecting a 3D point $\mathbf{p}$ on the image and $\pi^{-1}(\mathbf{u}, d) : \mathbb{R}^2 \times \mathbb{R} \to \mathbb{R}^2$ its back-projection according to the pixel localtion $\mathbf{u}$ and depth d. The per-pixel depth is given by the predicted δ_i and the object centroid depth $\mathbf{p}_i^z$, i.e., all pixels are associated with the object centroid. $\mathbf{p}_s$ stands for the extrinsic translation between stereo cameras. Note that we formulate a more accurate stereo alignment model using our predicted local depth (see Fig. 1) instead of the naïve box-shape in [23].

Temporal Constraints. Benefit from the geometric and unique property of the local coordinates representation, we can easily obtain temporal pixel correspondences by calculating the pairwise Euclidean distance between the local coordinates in associated object patches. An example of pixel correspondences can be found in the left column of Fig. 4. Let $\mathbf{u}_{i-1}$ be the dense correspondences for $\mathbf{u}_i$ in the previous frame, given by selecting the closest local coordinates. The temporal constraints encourage all $\mathbf{u}_i$ should also be projected to $\mathbf{u}_{i-1}$ (minimal reprojection error) after rigid-body transformation. Let

$$\mathbf{E}_{ti} := \sum_{\mathbf{u}_i \in \mathcal{N}_t} w_{\mathbf{p}} \left\| \mathbf{u}_i^p - \mathbf{u}_{i-1} \right\|_h, \tag{3}$$

where $\mathcal{N}_t$ is the set of pixels which found correspondence in the previous frame. $\mathbf{u}_i^p$ stands for the projected position of $\mathbf{u}_i$ in the previous frame, given by

$$\mathbf{u}_i^p = \pi\Big(\mathbf{R}_{i-1}\mathbf{R}_i^{-1}\big(\pi^{-1}(\mathbf{u}_i, \delta_i + \mathbf{p}_i^z) - \mathbf{p}_i\big) + \mathbf{p}_{i-1}\Big).$$

Pose Error. In above equations, the object pose in consecutive frames are coupled together, i.e., only relative motion is constrained. Although the object depth $\mathbf{p}_i^z$ is fully observable from Eq. 2, we still need object-level observation angle α_i and centroid projection $\mathbf{c}_i$ to constrain the object orientation and position in x, y direction for each frame separately, which can be simply given by a linear error

$$\mathbf{E}_{pi} := \|\pi(\mathbf{p}_i) - \mathbf{c}_i\| + \|\theta_i - \alpha_i - \arctan(\frac{\mathbf{p}_i^x}{\mathbf{p}_i^z})\| \tag{4}$$

Per-Frame Marginalization. To utilize the history information, a straight forward solution would be minimizing all above error terms over multiple frames in a sliding window,

$$\mathbf{E}_n = \sum_{i=0}^{n} \mathbf{E}_{si} + \mathbf{E}_{ti} + \mathbf{E}_{pi}. \tag{5}$$

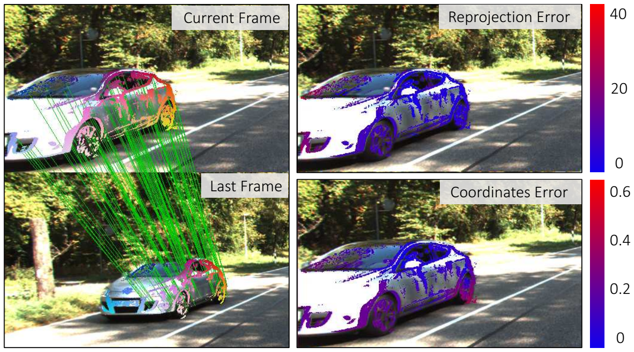

Figure 4. **Left**: Pixel correspondences, where we overlay the color mapped local coordinates on adjacent images. Green lines show sampled pixel matches using pair-wise coordinates distance. **Right**: Reprojection error vs. Coordinates error. The top and bottom show the error pattern for reprojection and coordinates aligning respectively.

However, re-evaluating the spatial alignment cost for all history frames at each timestamp is unnecessary as we already reach the minimum photometric error for historic frames. To fully exploit history information while avoiding the repeated computation, we use a per-frame marginalization strategy to convert the information from the previous optimization to a prior knowledge for the current frame, which is a common technique in SLAM approaches [9, 37].

For each new object, we joint solve the first two frames by minimizing $\mathbf{E}_2$ of Eq. 5 using Gauss-Newton optimization. We use a stacked 8 dimension vector $\mathbf{x} = [\mathbf{x}_1, \mathbf{x}_2]$ to denote the object states at $1^{th}, 2^{th}$ frames, where $\mathbf{x}_i = [\mathbf{p}_i, \theta_i] \in \mathbb{R}^4$ (transpose is omitted for simplicity). During each iteration, we update the object states by

$$\mathbf{x} \leftarrow \mathbf{x} + \Delta\mathbf{x}, \quad \text{with} \quad \Delta\mathbf{x} = -\mathbf{H}^{-1}\mathbf{b} \tag{6}$$

where $\mathbf{H} \in \mathbb{R}^{8\times 8}, \mathbf{b} \in \mathbb{R}^8$ are calculated by summarizing all costs and jacobians in Eq. 5 respecting to the target states via standard Guass-Newton process. $\Delta\mathbf{x} \in \mathbb{R}^8$ is the state increment respecting to the current linearization point. After several iterations, we achieve an optimal estimation $\mathbf{x}$ obtained from a linear system:

$$\mathbf{x} = \widetilde{\mathbf{x}} + \Delta\mathbf{x}, \quad \Leftrightarrow \quad \mathbf{H}\mathbf{x} = \mathbf{H}\widetilde{\mathbf{x}} - \mathbf{b}, \tag{7}$$

given by the last linearization point $\widetilde{\mathbf{x}}$ and the corresponding $\mathbf{H}, \mathbf{b}$. Eq. 7 can be thought as a linear constraint for $\mathbf{x}$ that jointly considers two frames' stereo alignment, dense temporal correspondences, and individual pose constraints. Writing the linear constraints separately for two frames, we have

$$\begin{bmatrix} \mathbf{H}_{11} & \mathbf{H}_{12} \\ \mathbf{H}_{21} & \mathbf{H}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{b}_1' \\ \mathbf{b}_2' \end{bmatrix} \quad \text{with} \ \ \mathbf{b}' = \mathbf{H}\widetilde{\mathbf{x}} - \mathbf{b} \tag{8}$$

where $\mathbf{H}_{11}, \mathbf{H}_{22}$ contain the individual stereo and pose information for $1^{th}, 2^{th}$ frames, while $\mathbf{H}_{12}, \mathbf{H}_{21}$ symmetricly involve the temporal relations from dense pixel correspondences. Marginalizing the $\mathbf{x}_1$ from Eq. 8 using Schur complement will give us $\widetilde{\mathbf{H}}\mathbf{x}_2 = \widetilde{\mathbf{b}}'$, derived by

$$\widetilde{\mathbf{H}} = \mathbf{H}_{22} - \mathbf{H}_{21}\mathbf{H}_{11}^{-1}\mathbf{H}_{12}; \quad \widetilde{\mathbf{b}}' = \mathbf{b}_2' - \mathbf{H}_{21}\mathbf{H}_{11}^{-1}\mathbf{b}_1' \tag{9}$$

As a result, we obtain an isolated linear constraint on the pose $\mathbf{x}_2$ of the 2^{th} frame while still taking both two frames' information into count.

When the object keep tracked in the 3^{th} frame, we can directly borrow the marginalized information as a prior to constrain the 2^{th} pose, meanwhile build the temporal constraints between 2^{th} and 3^{th} frame. Without loss of generality, for the i^{th} frame we minimize

$$\mathbf{E}_i = \mathbf{E}_{si} + \mathbf{E}_{ti} + \mathbf{E}_{pi} + \left\| \widetilde{\mathbf{H}}\mathbf{x}_{i-1} - \widetilde{\mathbf{b}}' \right\|. \tag{10}$$

After $\mathbf{x}_i$ is solved, we analogously marginalize $\mathbf{x}_{i-1}$ as derivative in Eq. 8,9 and extract the linear constraint for $\mathbf{x}_i$ that will be used for the next frame. In this way, we only need to evaluate the dense photometric error and temporal reprojection error for the current frame while still incorporate all history information. All previous stereo constraints will eventually contribute to the current estimation through step-by-step temporal relations. Note that our optimization solves a relative trajectory based on pure geometric relations, we thereby do not rely on the given ego-camera pose. Qualitative examples of our *relative* trajectory estimation can be found in Fig. 5.

Alternative Way to Model Temporal Relations. Besides finding dense pixel matching and minimizing the reprojection error in Eq. 3, we also explore an alternative way to model the temporal relations by directly aligning the object local coordinates patch in adjacent frames, given by:

$$\mathbf{E}_{ti} := \sum_{\mathbf{u}_i \in \mathcal{N}_t} w_{\mathbf{c}} \left\| C_i(\mathbf{u}_i^p) - C_{i-1}(\mathbf{u}_{i-1}) \right\|_h, \tag{11}$$

where the C_i, C_{i-1} are the foreground local coordinates map in the current and previous frames respectively. Benefit from our learned local coordinates representation, we can evaluate a smooth gradient on C_i, C_{i-1} map, and are robust to the exposure imbalance in different frames. We compare and analysis these two ways in the experiment section (Table 1, 2 and Fig. 4).

5. Implementation Details

Network Details. We use ResNet-50 [15] and FPN [29] as our network backbone. Three sets of anchor ratios {0.5, 1, 2} with four scales {4, 8, 16, 32} are used in the paired RPN stage. For each anchor, we regress 8-d outputs that

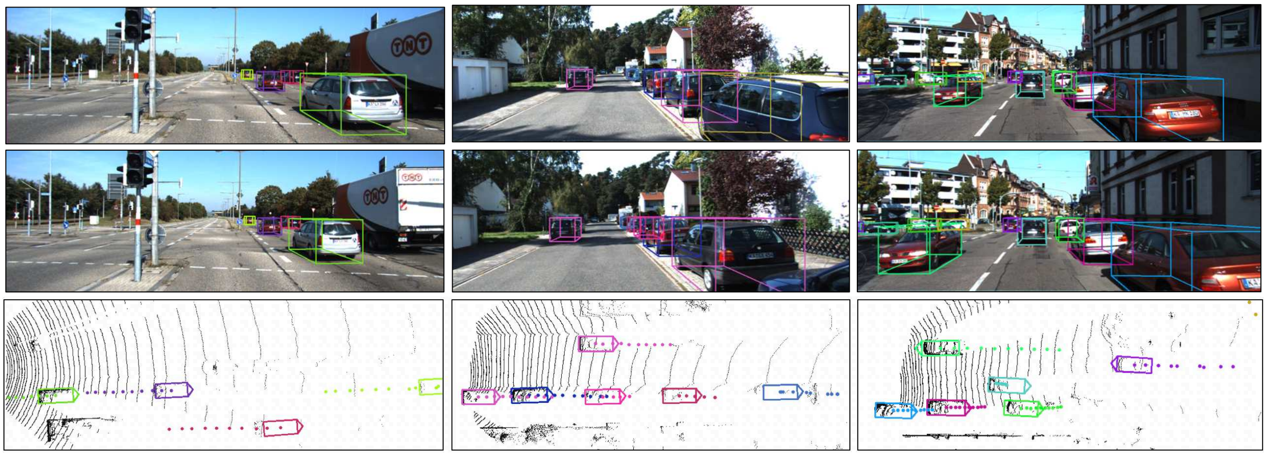

Figure 5. **Qualitative results of our 3D object tracking.** We project the estimated 3D bounding box on two sequential images and bird's view map, where different color represents unique tracking id. Note that the color dots represent the *relative* trajectory with respect to the corresponding ego-camera poses at each timestamp respectively.

correspond to the offsets for the box in the current and previous image respectively. For the 2D box regression and 3D box estimation, we fed the concatenated RoI feature maps into two sequential fully-connected layers to produce 1024-d feature vectors. Similarly, we have 8 channels outputs for the paired 2D box regression and 6 channels output for the 3D box centroid and dimension regression. Following [32], we use two *bin*s for angle classification and residual regression. For the dense prediction, we employ six stacked 256-d FCN (each layer is with 3 × 3 convolution and ReLU) on the dense RoI feature map, and predict 5-d dense output (1-d mask classification, 1-d local depth and 3-d local coordinates regression). The network inference time is ∼80 ms and the joint optimization takes ∼130 ms.

Training. As MOTS [44] provides dense instance segmentation labels for the KITTI tracking [13] sequences, we can directly use it for object mask supervision. The ground-truth for the local depth and local coordinates are calculated from the sparse LiDAR scanning and the labeled 3D bounding box. To leverage the network learn a better 3D object estimation, we firstly pretrain our model on the KITTI object dataset but excluded images appeared in the KITTI tracking sequences (∼ 4000 images are left). Since the KITTI object dataset only provides single image with 3D object label, we apply two random scales with opposite direction (e.g., 0.95, 1.05) on the original image then crop or pad them into the original size, which can be roughly thought as equivalent scales in 3D object position. We thus get simulated adjacent images to initialize our tracking network. After that, we train our network on the tracking dataset to learn more actual association patterns. We expand the training set to 2× by flipping each image respecting the vertical axis, where the object angle and local coordinates are also mirrored respectively. The total loss is defined as:

$$L = \lambda_1 L_{\text{rpn}} + \lambda_2 L_{\text{box}} + \lambda_3 L_{3d} + \lambda_4 L_{\text{angle}} + \lambda_5 L_{\text{dense}}, \tag{12}$$

where $L_{\text{rpn}}, L_{\text{box}}, L_{\text{angle}}, L_{\text{dense}}$ contain both classification loss and regression loss, λ_i denotes the individual uncertainty to balance the loss according to [17]. For each iteration, we feed one adjacent images pair into the network and sample 512 RoIs in RCNN stage. The network is trained using SGD optimizer with a momentum of 0.9 and a weight decay of 0.0005. We train 10 epochs with 0.001 learning rate followed by 2 epochs with 0.0001 learning rate.

6. Experiments

We evaluate our method on the KITTI tracking dataset [13] using the standard CLEAR [1] metric for multiple objects tracking (MOT) evaluation. As this paper focuses on the 3D object tracking, we define the similarity function between the estimated trackers and ground truth objects in the 3D space. Specifically, we use the overlap between two 3D object cuboids with 0.25 and 0.5 IoU thresholds to evaluate the 3D bounding box tracking performance, and use the Euclidean distance between 3D object centroids to evaluate the 3D trajectory tracking performance (3, 2, 1 meters thresholds are used respectively). The overall tracking performance is evaluated by the MOTA (multiple objects tracking accuracy), MOTP (multiple objects tracking precision), F1 score (calculated from the precision and recall), MT (most tracked percent), ML (most lost percent), and FP (false positives), FN (false negatives), etc. Since the official KITTI tracking server does not support 3D tracking evaluation, we follow [44] to split the whole train data into *training* and *val* set, and conduct extensive comparisons and

Method	Sensor	3D IoU = 0.25							3D IoU = 0.5						
		MOTA ↑	MOTP ↑	F1 ↑	MT ↑	ML ↓	# FP ↓	# FN ↓	MOTA ↑	MOTP ↑	F1 ↑	MT ↑	ML ↓	# FP ↓	# FN ↓
Joint-Tracking [16]	Mono	-15.55 [1]	47.91	42.14	9.33	33.33	3855	4868	-55.57	63.76	18.90	0.67	68.00	5378	6366
Semantic-Tracking [25]	Stereo	3.31	51.72	47.32	11.33	40.67	2662	4620	-34.14	65.39	24.72	2.00	62.67	4070	6054
Ours (Coord)	Stereo	56.14	62.20	77.53	42.67	14.00	820	2464	28.56	69.34	61.67	22.67	24.00	1730	3651
Ours (Repro)	Stereo	**56.70**	**62.33**	**77.85**	**44.00**	**12.00**	**794**	**2443**	**29.39**	**69.39**	**62.13**	**24.00**	**23.33**	**1697**	**3618**

Table 1. **3D bounding box tracking results on the KITTI tracking *val* set**, where 3D box IoU are used for True Positive (TP) assignments.

Method	Sensor	Distance = 3m					Distance = 2m					Distance = 1m				
		MOTA ↑	MOTP ↓	F1 ↑	MT ↑	ML ↓	MOTA ↑	MOTP ↓	F1 ↑	MT ↑	ML ↓	MOTA ↑	MOTP ↓	F1 ↑	MT ↑	ML ↓
3D-CNN/PMBM [41]	Mono	47.20	1.11 m	73.86	48.65	11.35	-	-	-	-	-	-	-	-		
Joint-Tracking [16]	Mono	47.22	1.13 m	75.63	40.00	7.33	27.16	0.88 m	65.20	28.00	12.67	-14.58	0.53 m	42.52	10.67	33.33
Semantic-Tracking [25]	Stereo	51.19	1.00 m	74.82	39.33	12.00	34.84	0.76 m	65.54	28.67	20.00	4.03	0.49 m	47.76	14.00	38.67
Ours (Coord)	Stereo	74.75	0.49 m	87.66	64.67	7.33	71.12	0.44 m	85.69	58.67	8.67	56.11	0.32 m	77.57	43.33	13.33
Ours (Repro)	Stereo	**74.92**	**0.49 m**	**87.77**	**65.33**	**7.33**	**71.40**	**0.44 m**	**85.85**	**60.67**	**8.00**	**56.74**	**0.32 m**	**77.94**	**47.33**	**12.00**

Table 2. **3D trajectory tracking results on the the KITTI tracking *val* set.** We assign the True Positive trajectories according to the 3D Euclidean distance between object centroids with different threshholds. Note that the tracking precision (MOTP) is defined based on the distance error, i.e., the lower the better.

Method	MOTA ↑	MOTP ↑	F1 ↑	MT ↑	ML ↓
Stereo R-CNN [23]	23.59	69.98	56.29	18.00	28.00
Pseudo-LiDAR [46]	25.88	**71.10**	58.14	20.00	25.33
Ours (Coord)	28.56	69.34	62.13	24.00	23.33
Ours (Repro)	**29.39**	69.39	**62.13**	**24.00**	**23.33**

Table 3. **Comparing with 3D detectors + KF tracker [48]**. Note that MOTP [2] is defined on TPs (3D IoU > 0.5) only, which is independent of the overall tracking a consistent trajectory ability.

ablation analysis on the val set for the car category. We also report 3D pedestrian tracking results and extend the evaluation to KITTI raw [12] and Argoverse tracking [5] dataset.

3D Object Tracking Evaluation. We compare our 3D bounding box and 3D trajectory tracking performance with recent image-based 3D object tracking approaches in Table.1, 2 respectively, where 3D-CNN/PMBM [41] and Joint-Tracking [16] use monocular image for object detection and use PMBM filter or LSTM to generate continuous 3D tracking. Semantic-Tracking [25] uses stereo images to achieve a better 3D localization accuracy. As the code for PMBM [41] is not available, we directly list the 3D trajectory tracking results in its original paper for reference. We finetune [16] on the same tracking split [44] based on its released pre-trained weight on a large scale GTA dataset. For Semantic-Tracking [25], we replace the 2D IoU-based association and the fixed size prior in its original implementation to our learned association and dimension for a fair comparison. As detailed in Table. 1, 2, our method significantly outperforms all image-based 3D object tracking methods for both 3D bounding box and 3D trajectory tracking evaluation. Note that 3D MOTA can be negative[1] as it assigns TPs using 3D IoU or 3D distance, which poses a high strict requirement for image-based approaches. Although [25] employs the stereo sensor and considers the motion consistency as well, however, it solves the object relative motion in a sliding window and aligns the object box to the sparse point cloud recovered by the discrete stereo feature matching in separate stages, which is in essence differ from our joint spatial-temporal optimization approach. Both the sparse stereo matching and loosely coupled spatial-temporal information limit its 3D tracking performance.

We also note that modeling temporal relations by local coordinates error in Eq. 11 (denoted as Coord) slightly underperforms the reprojection error in Eq. 3 (denoted as Repro). As minimizing the local coordinates error tries to align the whole object patch, however, the visible areas are not identical even for adjacent frames due to slight viewpoint changing and truncation. An error pattern to reveal the phenomenon can be found in Fig. 4, where we can observe a large error in the rear wheel region because the optimizer tries to align the truncated patch to the complete patch in the last frame. Minimizing reprojection error avoids this issue easily by setting a distance threshold for local coordinates matching. If not specified, we report our (repro) results in the following experiments by default.

Comparison with 3D Detection Methods. To further demonstrate our tracking performance, we extend the comparison to state-of-the-art stereo 3D detection methods Stereo RCNN [23] and Pseudo LiDAR [46]. We train these two detectors on our KITTI object split which does not contain images in KITTI tracking sequences, and run the inference on the KITTI tracking val set. We use the recent proposed KF-based tracker [47] to associate the discrete de-

[1] $\text{MOTA} = (1 - \frac{\sum(\text{FN}+\text{FP}+\text{IDS})}{\sum\text{GT}}) \times 100$, i.e. $\in (-\infty, 100)$

Method	MOTA ↑	MOTP ↑	F1 ↑	MT ↑	ML ↓
Mono Regress	-35.26	66.96	22.28	0.00	64.00
+ Spatial	26.86	69.24	60.62	18.00	24.00
+ Temporal	**29.39**	**69.39**	**62.13**	**24.00**	**23.33**

Table 4. **Comparing effects of adding different information**.

tections and produce sequential 3D object trajectories. As Table. 3 shows, although the detection-based method [46] shows a good precision (MOTP) for True Positives, a KF tracker cannot guarantee the optimal trajectory from only detection data as most of the original information is lost. We outperform them in the overall tracking performance (MOTA, MT, etc), which evidences again the advantage of our joint spatial-temporal optimization approach.

Benefits of Spatial & Temporal Information. This experiment shows how the spatial and temporal information helps our 3D object tracking. As listed in Table. 4, we use the Regress to denote the 3D tracking result using the monocular regressed 3D box only, which shows inadequate 3D tracking performance. While modeling spatial constraints (stereo alignment) significantly improves the 3D localization ability due to introducing accurate depth-sensing ability. Further, adding temporal information by considering geometric relations and motion consistency improves 3D tracking robustness again. The tracking accuracy (MOTA), tracking precision (MOTP) and tracking robustness (MT, ML) are all improved by remarkable margins.

More Quantitative Experiments. Since our method predicts object shape and is based on pure geometry, we can seamlessly use it for 3D pedestrian tracking. The quantitative results on the KITTI tracking set and an example can be found in Table. 5 and Fig. 6 respectively. Besides the evaluation on the KITTI tracking dataset, we also report our 3D tracking results on the KITTI raw sequence [12] and Argoverse Tracking [5] dataset for future benchmarking. As reported in Table. 6, we evaluate on totall 24 KITTI raw sequences that are excluded from the tracking dataset. Note that here we train the network on the whole KITTI tracking dataset without pretraining on the object dataset as the KITTI object images are distributed in most of the raw sequences. The Argoverse dataset provides stereo images with 5 fps and labeled 3D object trackers on 10 fps LiDAR scans. Since the official server only evaluates the 10 fps 3D object tracking on the LiDAR timestamps, we thereby report our results on the 24 stereo validation sequences by assigning the ground truths of the LiDAR frame with the nearest timestamp. As detailed in Table. 6, we note that our image-based method works reasonably in short range while unavoidably suffers from performance decent for long-range objects. This is due to a combined reason for low fps stereo images, reciprocal relations between disparity and depth, and non-trivial projection error for extremely faraway objects, etc.

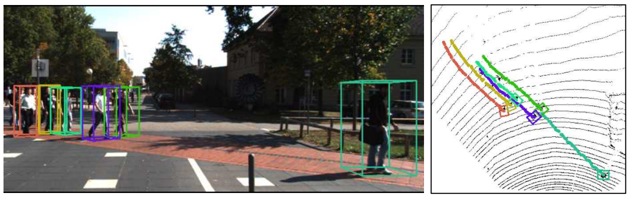

Figure 6. **Qualitative Example for 3D Pedestrian Tracking**.

Threshhold	MOTA	MOTP	F1	MT	ML	# FP	# FN
Distance = 1 m	33.79	0.26 m	67.78	44.12	13.24	1014	1082
3D IoU = 0.25	16.73	48.02	58.60	27.94	22.06	1276	1392

Table 5. **3D pedestrian tracking results on KITTI tracking *val* set**. Note that we require the true positive trajectory has < 1 m distance error since pedestrians are more crowded than vehicles.

Dataset	Threshhold	MOTA	MOTP	F1	MT	ML
KITTI Raw	Distance = 2 m	63.02	0.47 m	84.81	50.32	14.95
	3D IoU = 0.25	46.29	59.88	77.07	37.89	21.05
Argoverse Tracking	Range 100 m	4.10	0.93 m	46.30	16.09	40.66
	Range 50 m	25.71	0.87 m	63.00	30.72	21.02
	Range 30 m	43.81	0.68 m	76.24	72.92	7.22

Table 6. **Evaluation on KITTI raw sequences and Argoverse datasets**, where we seperate the evaluation range and use 2.25 m 3D centroid distance as the threshold for true positive assignment following the Argoverse official setting.

7. Conclusion

In this paper, we propose a joint spatial-temporal optimization approach for stereo 3D object tracking. Our method models the relations between the invisible object centroid and the local object geometric cues into a joint spatial photometric and temporal reprojection error function. By minimizing the joint error with a per-frame marginalized prior, we estimate an optimal object trajectory that satisfies both the instant stereo constraints and accumulated history evidence. Our approach significantly outperforms previous image-based 3D tracking methods on the KITTI tracking dataset. Extensive experiments on multiple categories and larger datasets (KITTI raw and Argoverse Tracking) are also reported for future benchmarking.

Acknowledgment. This work was supported by the HKUST Postgraduate Studentship and the Hong Kong Research Grants Council Early Career Scheme under project 26201616.

References

[1] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: The clear mot metrics. *EURASIP Journal on Image and Video Processing*, 2008(1):246309, May 2008. 6

[2] Keni Bernardin and Rainer Stiefelhagen. Evaluating multiple object tracking performance: the clear mot metrics. *Journal on Image and Video Processing*, 2008:1, 2008. 7

[3] Garrick Brazil and Xiaoming Liu. M3d-rpn: Monocular 3d region proposal network for object detection. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 9287–9296, 2019. 2

[4] Florian Chabot, Mohamed Chaouch, Jaonary Rabarisoa, Céline Teulière, and Thierry Chateau. Deep manta: A coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.(CVPR)*, pages 2040–2049, 2017. 2

[5] Ming-Fang Chang, John Lambert, Patsorn Sangkloy, Jagjeet Singh, Slawomir Bak, Andrew Hartnett, De Wang, Peter Carr, Simon Lucey, Deva Ramanan, and James Hays. Argoverse: 3d tracking and forecasting with rich maps, 2019. 7, 8

[6] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3d object detection for autonomous driving. In *European Conference on Computer Vision*, pages 2147–2156, 2016. 2

[7] Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 3d object proposals using stereo imagery for accurate object class detection. In *TPAMI*, 2017. 2

[8] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *IEEE CVPR*, volume 1, page 3, 2017. 1, 2

[9] Jakob Engel, Vladlen Koltun, and Daniel Cremers. Direct sparse odometry. *IEEE transactions on pattern analysis and machine intelligence*, 40(3):611–625, 2017. 5

[10] Martin Engelcke, Dushyant Rao, Dominic Zeng Wang, Chi Hay Tong, and Ingmar Posner. Vote3deep: Fast object detection in 3d point clouds using efficient convolutional neural networks. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 1355–1361. IEEE, 2017. 2

[11] Francis Engelmann, Jörg Stückler, and Bastian Leibe. Samp: shape and motion priors for 4d vehicle reconstruction. In *2017 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 400–408. IEEE, 2017. 3

[12] Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013. 7, 8

[13] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012. 6

[14] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, pages 2980–2988. IEEE, 2017. 1, 3, 4

[15] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 5

[16] 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. *arXiv preprint arXiv:1811.10742*, 2018. 1, 3, 7

[17] Alex Kendall, Yarin Gal, and Roberto Cipolla. Multi-task learning using uncertainty to weigh losses for scene geometry and semantics. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017. 6

[18] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven Waslander. Joint 3d proposal generation and object detection from view aggregation. *arXiv preprint arXiv:1712.02294*, 2017. 2

[19] Jason Ku, Alex D Pon, and Steven L Waslander. Monocular 3d object detection leveraging accurate proposals and shape reconstruction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11867–11876, 2019. 1, 2

[20] Abhijit Kundu, Yin Li, and James M Rehg. 3d-rcnn: Instance-level 3d object reconstruction via render-and-compare. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3559–3568, 2018. 2

[21] Alex H Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 12697–12705, 2019. 1, 2

[22] Bo Li, Tianlei Zhang, and Tian Xia. Vehicle detection from 3d lidar using fully convolutional network. *In Robotics: Science and Systems*, 2016. 2

[23] Peiliang Li, Xiaozhi Chen, and Shaojie Shen. Stereo r-cnn based 3d object detection for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7644–7652, 2019. 1, 2, 3, 4, 7

[24] Peiliang Li, Siqi Liu, and Shaojie Shen. Multi-sensor 3d object box refinement for autonomous driving, 2019. 4

[25] Peiliang Li, Tong Qin, and Shaojie Shen. Stereo vision-based semantic 3d object and ego-motion tracking for autonomous driving. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 646–661, 2018. 1, 7

[26] Peiliang Li, Tong Qin, and Shaojie Shen. Stereo vision-based semantic 3d object and ego-motion tracking for autonomous driving. In *European Conference on Computer Vision*, pages 664–679. Springer, 2018. 3

[27] Ming Liang, Bin Yang, Yun Chen, Rui Hu, and Raquel Urtasun. Multi-task multi-sensor fusion for 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7345–7353, 2019. 1, 2

[28] Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. Deep continuous fusion for multi-sensor 3d object detection.

In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 663–678, 2018. 1, 2

[29] Tsung-Yi Lin, Piotr Dollár, Ross B Girshick, Kaiming He, Bharath Hariharan, and Serge J Belongie. Feature pyramid networks for object detection. In *CVPR*, volume 1, page 4, 2017. 5

[30] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3569–3577, 2018. 1, 3

[31] Xinzhu Ma, Zhihui Wang, Haojie Li, Pengbo Zhang, Wanli Ouyang, and Xin Fan. Accurate monocular 3d object detection via color-embedded 3d reconstruction for autonomous driving. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6851–6860, 2019. 1, 2

[32] Arsalan Mousavian, Dragomir Anguelov, John Flynn, and Jana Košecká. 3d bounding box estimation using deep learning and geometry. In *Computer Vision and Pattern Recognition (CVPR), 2017 IEEE Conference on*, pages 5632–5640. IEEE, 2017. 4, 6

[33] Aljoša Osep, Wolfgang Mehner, Markus Mathias, and Bastian Leibe. Combined image-and world-space tracking in traffic scenes. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1988–1995. IEEE, 2017. 3

[34] Alex D Pon, Jason Ku, Chengyao Li, and Steven L Waslander. Object-centric stereo matching for 3d object detection. *arXiv preprint arXiv:1909.07566*, 2019. 2

[35] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. *arXiv preprint arXiv:1711.08488*, 2017. 1, 2

[36] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 1(2):4, 2017. 2

[37] Tong Qin, Peiliang Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, 2018. 5

[38] Z Qin, J Wang, and Y Lu. Monogrnet: A geometric reasoning network for 3d object localization. In *AAAI Conference on Artificial Intelligence (AAAI)*, volume 2, page 8, 2019. 2

[39] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99, 2015. 3

[40] Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary R Bradski. Orb: An efficient alternative to sift or surf. In *ICCV*, volume 11, page 2. Citeseer, 2011. 4

[41] Samuel Scheidegger, Joachim Benjaminsson, Emil Rosenberg, Amrit Krishnan, and Karl Granström. Mono-camera 3d multi-object tracking using deep learning detections and pmbm filtering. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 433–440. IEEE, 2018. 3, 7

[42] Shaoshuai Shi, Xiaogang Wang, and Hongsheng Li. Pointrcnn: 3d object proposal generation and detection from point cloud. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–779, 2019. 1, 2

[43] Martin Simon, Karl Amende, Andrea Kraus, Jens Honer, Timo Samann, Hauke Kaulbersch, Stefan Milz, and Horst Michael Gross. Complexer-yolo: Real-time 3d object detection and tracking on semantic point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 3

[44] Paul Voigtlaender, Michael Krause, Aljosa Osep, Jonathon Luiten, Berin Balachandar Gnana Sekar, Andreas Geiger, and Bastian Leibe. Mots: Multi-object tracking and segmentation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019. 6, 7

[45] He Wang, Srinath Sridhar, Jingwei Huang, Julien Valentin, Shuran Song, and Leonidas J Guibas. Normalized object coordinate space for category-level 6d object pose and size estimation. *arXiv preprint arXiv:1901.02970*, 2019. 4

[46] 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. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8445–8453, 2019. 1, 2, 7, 8

[47] Xinshuo Weng and Kris Kitani. A Baseline for 3D Multi-Object Tracking. *arXiv:1907.03961*, 2019. 7

[48] Xinshuo Weng and Kris Kitani. A baseline for 3d multi-object tracking. *arXiv preprint arXiv:1907.03961*, 2019. 7

[49] Bin Xu and Zhenzhong Chen. Multi-level fusion based 3d object detection from monocular images. In *IEEE CVPR*, 2018. 1, 2

[50] Bin Yang, Wenjie Luo, and Raquel Urtasun. Pixor: Real-time 3d object detection from point clouds. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7652–7660, 2018. 2

[51] Muhammad Zeeshan Zia, Michael Stark, and Konrad Schindler. Are cars just 3d boxes?-jointly estimating the 3d shape of multiple objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3678–3685, 2014. 2

[52] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. *arXiv preprint arXiv:1711.06396*, 2017. 1, 2