Know Your Surroundings: Panoramic Multi-Object Tracking by Multimodality Collaboration

Yuhang He[1], Wentao Yu[2], Jie Han[2], Xing Wei[2], Xiaopeng Hong[3,4]*, Yihong Gong[2]
[1]College of Artificial Intelligence , Xi'an Jiaotong University
[2]School of Software Engineering, Xian Jiaotong University
[3]School of Cyber Science and Engineering, Xi'an Jiaotong University
[4]Research Center for Artificial Intelligence, Peng Cheng Laboratory
{hyh1379478,yu1034397129,hanjie1997}@stu.xjtu.edu.cn, {weixing,hongxiaopeng,ygong}@mail.xjtu.edu.cn

Abstract

In this paper, we focus on the multi-object tracking (MOT) problem of automatic driving and robot navigation. Most existing MOT methods track multiple objects using a singular RGB camera, which are prone to camera field-of-view and suffer tracking failures in complex scenarios due to background clutters and poor light conditions. To meet these challenges, we propose a MultiModality PAnoramic multi-object Tracking framework (MMPAT), which takes both 2D panorama images and 3D point clouds as input and then infers target trajectories using the multimodality data. The proposed method contains four major modules, a panorama image detection module, a multimodality data fusion module, a data association module and a trajectory inference model. We evaluate the proposed method on the JRDB dataset, where the MMPAT achieves the top performance in both the detection and tracking tasks and significantly outperforms state-of-the-art methods by a large margin (15.7 and 8.5 improvement in terms of AP and MOTA, respectively).

1. Introduction

Multiple Object Tracking (MOT) aims to locate the positions of interested targets, maintains their identities across frames and infers a complete trajectory for each target. It has a wide range of applications in video surveillance [82, 27], custom behavior analysis [21, 27], traffic flow monitoring [56] and etc. Benefited from the rapid development of object detection techniques [64, 24, 8, 48, 74, 87, 40], most state-of-the-art MOT trackers follow a *tracking-by-detection* paradigm. They first detect targets in each image using modern object detectors and then associate these detection responses into trajectories by *data association*. These methods have achieved significant improvement in recent years and became the main stream of MOT.

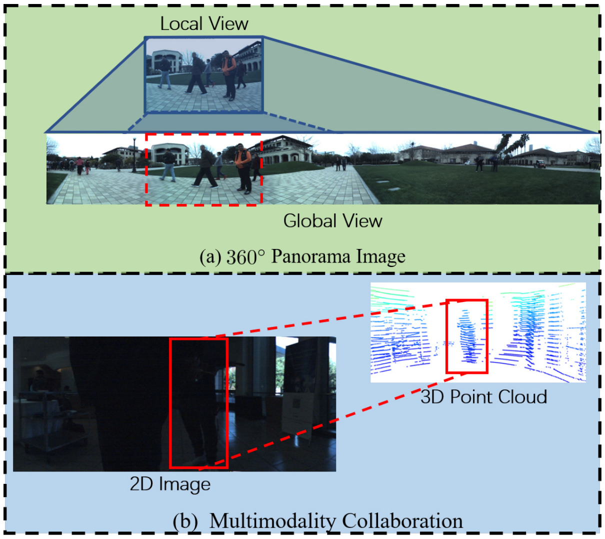


Figure 1. Illustrations of multimodality panoramic multi-object tracking. (a) 360° panorama image. (b). Multimodality collaboration.

Accurate and efficient as they are, these methods are prone to camera field-of-view (FOV) and cannot handle the blind areas of camera views. Besides, limited to the properties of RGB cameras, these methods also have difficulties tracking targets in complex scenarios such as poor light conditions and background clutters. Figure 2 illustrates a couple of tracking examples of the singular camera multi-object tracking. In (a), the MOT trackers track targets in a crowded scene. We can see that, the MOT trackers only generate sporadic trajectories while unconscious of the other targets in the surrounding. In (b), MOT trackers are failed to track targets due to background clutters and poor light condition-

*Corresponding author.

s. These drawbacks of the single-camera MOT trackers prevent them from many important applications such as robot navigation [54] and automatic driving [22, 6].

To meet this challenge, we propose a MultiModality PAnoramic multi-object Tracking framework (MMPAT), which takes 2D 360° panorama images and 3D LiDAR point clouds as input and generates trajectories for targets by multimodality collaboration. The key insights of our MMPAT are twofold. First, *a wider vision brings more information*. As shown in Figure 1 (a), compared with the singular-camera MOT that tracks targets in a local view, taking the 360° panorama images as input enables us to have a global view of the surroundings and opens up opportunities for optimal tracking. Second, *singular modality is biased while multimodality complement each other*. As shown in Figure 1 (b), when the target in red bounding box is invisible due to poor light condition, the 3D point cloud supplements the target information for tracking. This provides a foundation for robust object tracking in complex scenarios. On this basis, we design the MMPAT algorithm with taking both the 2D 360° images and 3D point clouds as input. The proposed method is an online MOT method containing four major modules, 1) a panoramic object detection module, 2) a multimodality data fusion module, 3) a data association module and 4) a trajectory extension module. The Module 1 takes 2D images as input and outputs detection results for the panorama images. As panorama images are often longwidth, the target responses in the feature maps of panorama images are narrow. This makes it difficult to locate the targets and generate accurate bounding boxes. To handle this problem, we design a split-detect-merge detection mechanism to detect targets in panorama images, which first splits panorama image into image slices, then detects targets in each slice, and finally merges detection responses from different slices. In Module 2, we fuse 2D images with 3D point clouds and append each detection with a 3D location characteristic. In Module 3, we match existing trajectories with newly obtained detections, where target appearance, motion and 3D location are exploited for data association. In Module 4, we generate accurate and complete trajectories for targets according to the data association results. The proposed MMPAT achieves the best performance in the detection and tracking tracks of the 2nd JRDB workshop[1] and significantly outperforms state-of-the-art methods by a large margin (15.7 and 8.5 improvements on AP and MOTA, respectively).

In summary, the contributions of this paper include:

- We propose a MultiModality PAnoramic multi-object Tracking framework (MMPAT) for robot navigation and automatic driving.
- We provide an efficient object detection mechanism to detect targets in panorama images.
- We design a 3D points collection algorithm to associate the point clouds with 2D images.
- The proposed method significantly improves the detection and tracking performance by a large margin.

2. Related Work

2.1. 2D Multi-object Tracking

Motivated by the rapid development of using Deep Convolution Neural Network in computer vision tasks [53, 40, 24, 26, 46], most state-of-the-art MOT methods follow a tracking-by-detection paradigm due to the rapid development of the DCNN-based detection techniques. According to whether frames following the future frame are available in the tracking process, these methods can be further divided into two subcategories: offline and online trackers

Offline MOT methods allow using (a batch of) the entire sequence to obtain the global optimal solution of the data association problem. A series of works [51, 69, 73, 75, 83, 84] use graph models to link detections or tracklets (short trajectories) in the graph into trajectories. Ma *et al.* [51] introduce a hierarchical correlation clustering (HCC) framework which builds different graph construction schemes at different levels to generate local, reliable tracklets as well as globally associated tracks. Wang *et al.* [75] utilize a graph model to generate tracklets by associating detections based on the appearance similarity and the spatial consistency measured by the multi-scale TrackletNet and cluster these tracklets to get global trajectories. A few methods [32, 23, 33] tackle MOT problems by finding the most likely tracking proposals. Kim *et al.* [32] propose a novel multiple hypotheses tracking (MHT) method which enumerates multiple tracking hypothesis and selects the most likely proposals based on the features from long-term appearance models. There are also methods formulating the result optimization problem of MOT as a minimum cost lifted multicut problem [73], a multidimensional assignment problem for multiple tracking hypothesis [32], a Maximum Weighted Independent Set (MWIS) problem [68] or a lifted disjoint paths problem [28]. Besides, a series of deep network based trackers are developed, such as Deep Tracklet Association (DTA) [88], bilinear LSTM (bLSTM) [33], Message Passing Netowrk (MPN) [4], and TrackletNet Tracker (TNT) [75]. There are other methods improve the performance of MOT by Tracklet-Plane Matching (TPM) [58] and Correlation Co-Clustering (CCC) [31].

Online MOT methods require that only the information in the current frame and the previous frame can be used to predict the tracking result of current frame, and the tracking result of the previous frame cannot be modified based

[1] https://jrdb.stanford.edu/workshops/jrdb-cvpr21

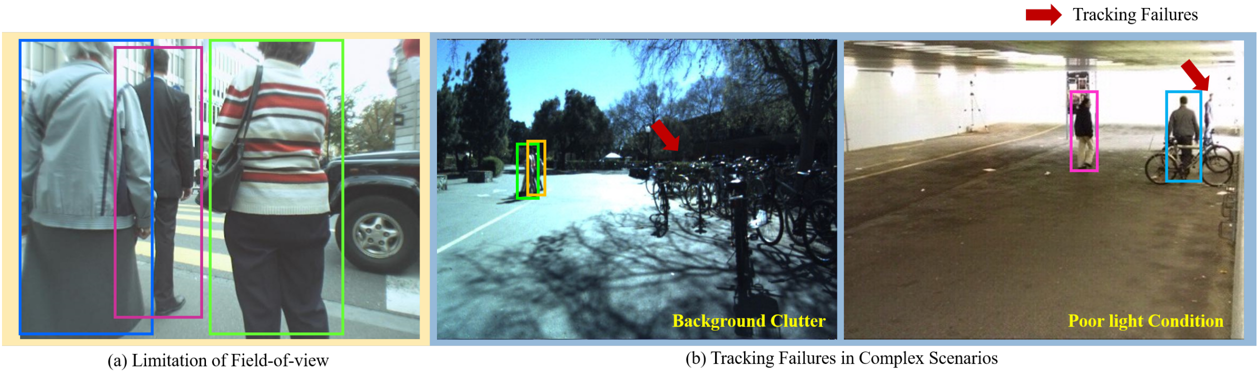

Tracking Failures
Background Clutter
Poor light Condition
(a) Limitation of Field-of-view
(b) Tracking Failures in Complex Scenarios

Figure 2. Limitations of the single-modality single-camera tracking. (a) Limitation of camera field-of-view. (b) Tracking failures in background clutters and poor light conditions. The red arrows point to tracking failures.

on the information of the current frame. A large number of research studies [77, 14, 78, 80, 66] utilize bipartite matching to tackle online MOT problems. Wojke *et al.* [77] divide the existing trajectories and new detections into two disjoint sets, and tackle the trajectory-detection matching problems by the Hungarian algorithm [55]. The method [66] uses a Recurrent Neural Networks (RNN) to integrate Appearance, Motion and Interaction information (AMIR) to jointly learn robust target representations. A series of deep learning approaches are proposed to measure the similarity between a target and a tracklet, like Spatial Temporal Attention Mechanism (STAM) [16], Recurrent Autoregressive Network (RAN) [20], Dual Matching Attention Network (DMAN) [90], and Spatial-Temporal Relation Network (STRN) [80]. In FAMNet [15], Feature extraction, Affinity estimation and Multi-dimensional assignment are integrated into a single Network. Besides, there are several works that incorporate the technologies from other fields, such as Tracktor++ [1] leverages bounding box regression from object detection, Instance Aware Tracking (IAT) [14] leverages the idea of model updatng from single object tracking, and GSM [47] leverages the graph matching module from target relations.

2.2. 3D Multi-object Tracking

3D Object Detection. There is a large literature on the use of instant sensor data to detect 3D object in the domain of autonomous driving. Depending on the modality of input data, 3D object detectors can be roughly divided into three categories: monocular image-based methods, stereo imagery-based methods, and LiDAR-based methods. Given a monocular image, early 3D object detection works [85, 11, 10, 38] usually exploit the rich detail information of the 3D scene representation to strengthen the understanding of 3D targets, like semantic and object instance segmentation, shape features and location priors, key-point, and instance model, while later state-of-the-art studies [79, 36, 62, 52, 5] pay more attention to 3D contexts and the depth information encoding from multiple levels for accurate 3D localization. Compared with the monocular-image based methods, stereo-imagery based methods [12, 42, 76, 59] add additional images with known extrinsic configuration and achieve much better 3D object detection accuracy.The method [12] first generates high-quality 3D object proposals with stereo imagery by encoding depth informed features that reason about free space, point cloud densities and distance to the ground, and employs a CNN on these proposals to perform 3D object detection. Stereo R-CNN [42] exploits object-level disparity information and geometric-constraints to get object detection by stereo imagery alignment. Wang *et al.* [76] convert image-based depth maps generated from stereo imagery to pseudo-LiDAR representations and apply existing LiDAR-based detection approaches to detect object in 3D space. In addition to image-based methods, there is abundant literature [41, 18, 89, 39, 60, 71] directly using the 3D information from the LiDAR point cloud to detect 3D objects. Several works [18, 89, 39] sample the unstructured point cloud as a structured voxel representation and encode the features using 2D or 3D convolution networks to detect object. Methods [41, 81] utilize conventional 2D convolutional networks to achieve 3D object detection by projecting point clouds to the front or bird's-eye views. Besides, there are also methods [60, 71] directly employ raw unstructured point clouds to localize 3D objects with the help of PointNet [61] encoder. Moreover, there are other methods [13, 45, 35, 44] fusing LIDAR point clouds and RGB images at the feature level for multi-modality detection.

3D Object Tracking. Due to the success of the tracking-by-detection paradigm in 2D object tracking, many 3D object tracking methods also follow this paradigm. Based on the 3D detection results, methods [57, 67, 72] utilize filter-based (Kalman filter; Poisson multi-Bernoulli mixture filter) algorithm to continuously track 3D objects,

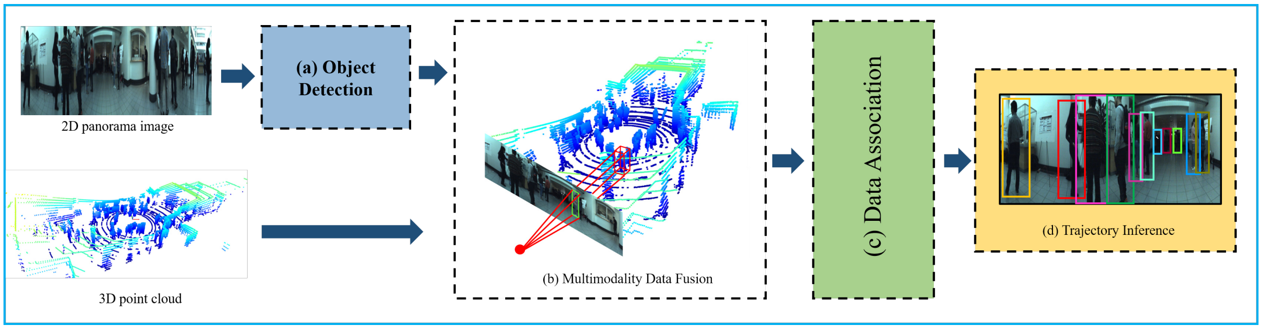


Figure 3. Overview of the MMPAT framework. The inputs are 2D image frames and 3D point clouds. The proposed method contains four major modules. a) The object detection module. b) The Multimodality data fusion module. c) The data association module. d) The Trajectory inference module.

while Hu *et al.* [29] design an LSTM-based module using data-driving approaches to directly learn the object motion for more accurate long-term tracking. However, the loss of information caused by decoupling detection and tracking may lead to sub-optimal solutions. Benefit from stereo images, the method [19] focuses on reconstructing the object using 3D shape and motion priors, and the method [43] exploits a dynamic object bundle adjustment (BA) approach which fuses temporal sparse feature correspondences and the semantic 3D measurement model to continuously track the object, while the performance on 3D localization for occluded objects is limited. From another aspect, Luo *et al.* [50] encode 3D point clouds into 3D voxel representations and jointly reason about 3D detection, tracking and motion forecasting so that it is more robust to occlusion as well as sparse data at range.

3. Methodology

In this section, we first overview the framework of our proposed method and then provide detailed descriptions of the key techniques.

3.1. Framework Overview

As illustrated in Figure 3, the proposed MMPAT is an online MOT method containing four major modules: 1) an object detection module to locate targets in the panorama images, 2) a multimodality data fusion module to associate 3D point clouds with 2D images, 3) a data association module to match existing trajectories with newly obtained detections and 4) a trajectory inference module to generate trajectories for targets. In the following, we provide detailed descriptions of each module.

3.2. Object Detection in Panorama Image

Compared with object detection on ordinary-size images (such as 720P and 1080P images), there are two additional challenges that need to be solved with the panorama images. First, most two-stage object detectors resize the input images into a fixed size and then generate region proposals on the feature maps. However, as the 2D panorama images have a long width, the target responses are narrow and feeble in the feature maps of resized images. This makes it difficult to locate the target in the feature maps and generate accurate proposals for the targets. Second, in panorama images, the size of targets often varies in a large range due to perspective changes. This leaves a challenging problem to handle the size variations of targets for accurate object detection. To tackle these problems, we design an object detection algorithm for panorama image. As shown in Figure 4, we first split the panorama image into several image slices along the image width. Then, we detect objects in each image slice following a cascade detection paradigm [7]. In the end, detections responses from different image slices are merged together using NMS [3].

3.2.1 Detection Pipeline

Panorama image split. Given the panorama image $\mathbf{I}_t$ at time t, we first split the image $\mathbf{I}_t$ into N image slices $\mathcal{I}_t = \{\mathbf{I}_t^n\}_{n=1}^N$, where the image slices $\mathcal{I}_t$ are obtained by splitting image $\mathbf{I}_t$ along the width dimension with an overlap of 0.2.

Cascade object detector. We then detect objects in each image slice $\mathbf{I}_t^n$ using a cascade object detector. As shown in Figure 4, the object detector is composed of three components, *i.e.*, a deformable convolution network, a region proposal network and a cascade detection header. In the deformable convolution network, we take the ResNet50 architecture [25] as our backbone, with the fully-connected layers and last pooling layer removed. To handle the target size variations, a deformable convolution [17] is employed, which adds 2D offsets to the sampling location of standard convolutions and enables free form deformations of the sampling grid. Then, taking the feature maps, a region proposal network [64] is adopted to generate proposals of targets. Taking the region proposals and feature maps as input, the cascade detection header iteratively regresses the bounding boxes to produce more accurate bounding box-

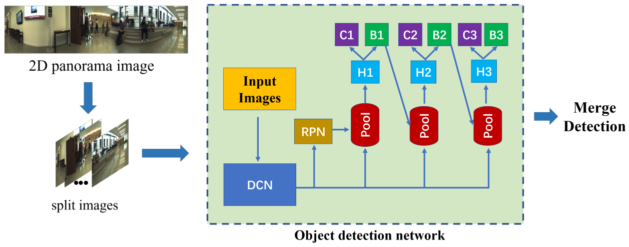


Figure 4. Pipeline of object detection. The "DCN" denotes deformable convolution network, "RPN" denotes region proposal network, "pool" denotes pooling layers, "H" denotes regression head layer, "B" and "C" denote bounding box and classification, respectively.

es. At each regression layer l, there is a classifier h_l and a bounding box regressor f_l. For a bounding box $\mathbf{b}$, the cascade object detector iteratively regresses the bounding box, which can be written as:

$$f(x, \mathbf{b}) = f_L \circ f_{L-1} \circ \dots f_1(x, \mathbf{b}), \tag{1}$$

where x is the input feature map and L is the total number of regression layers. At the inference, the regression takes the region proposals as input and iteratively regresses bounding boxes. We use $\mathbf{b}_0$ to denote the input region proposals and $\mathbf{b}_l = f_l(x_l, \mathbf{b}_{l-1})$ to denote the output bounding box of the l-th regressor.

Detection merge. We use $\mathcal{B}_t(i)$ to denote the bounding box collection of the i-th image slice $\mathbf{I}_t^i$. We merge detection responses from all the image slices by Non-Maximum Suppression (NMS) [3]:

$$\mathcal{B}_t = \text{NMS}(\mathcal{B}_t(1), ..., \mathcal{B}_t(N)), \tag{2}$$

where $\mathcal{B}_t$ denotes the detection set of panorama image $\mathbf{I}_t$. We use $\mathcal{B}_t^i$ to denote the i-th detection in $\mathcal{B}_t^i$.

3.2.2 Loss Function

For each regression layer l of the cascade object detector, the loss function is composed of two parts: bounding box regression and classification.

Bounding box regression. The objective of bounding box regression is to refine a candidate bounding box $\mathbf{b}_l = (\mathbf{b}_l^x, \mathbf{b}_l^y, \mathbf{b}_l^w, \mathbf{b}_l^h)$ into a ground-truth bounding box $\mathbf{g}_l = (\mathbf{g}_l^x, \mathbf{g}_l^y, \mathbf{g}_l^w, \mathbf{g}_l^h)$, where $(*^x, *^y)$ are the coordinate of bounding box center and $*^w$ and $*^h$ are the width and height, respectively. Transforming this objective into loss function, we have:

$$\text{L}_{loc}(\mathbf{b}_l, \mathbf{g}_l) = \sum_{j \in \{x,y,w,h\}} \text{smooth}_{L1}(\mathbf{b}_l^j - \mathbf{g}_l^j), \tag{3}$$

in which

$$\text{smooth}_{L1}(a) = \begin{cases} 0.5\,a^2, & \text{if } |a| < 1, \\ |a| - 0.5, & \text{else.} \end{cases} \tag{4}$$

Classification. We adopt the cross-entropy loss to optimize the classification header. We use $\mathbf{y}_l$ to denote the one-hot ground-truth label of $\mathbf{b}_l$ and use $\mathbf{p}_l = h_l(\mathbf{b}_l)$ to denote the output classification vector of $\mathbf{b}_l$. Then the classification loss function can be written as:

$$L_{cls}(\mathbf{b}_l, \mathbf{y}_l) = \text{CrossEntropy}(\mathbf{p}_l, \mathbf{y}_l), \tag{5}$$

in which

$$\text{CrossEntropy}(p, q) = -\sum_{i}^{C+1} p(i) \cdot log(q(i)), \tag{6}$$

where C is the number of classes and $p(i)$ (or $q(i)$) is the i-th element of the vector p (or q). On this basis, the loss function of the cascade object detector can be formulated as:

$$L_{total}(\mathbf{b}_l, \mathbf{g}_l, \mathbf{y}_l) = \sum_{l=1}^{L} L_{cls}(\mathbf{b}_l, \mathbf{y}_l) + \delta(\mathbf{y}_l) \cdot L_{loc}(\mathbf{b}_l, \mathbf{g}_l), \tag{7}$$

where $\delta(\mathbf{y}_l) = 0$ if $\mathbf{y}_l$ belongs to background class and $\delta(\mathbf{y}_l) = 1$ for otherwise.

3.3. Multimodality Data Fusion

As shown in Figure 5, this module aims to associate detections with 3D points and append each detection $\mathcal{B}_t^i$ with a 3D location characteristic l_t^i. The collection of 3D points for a detection contains two steps. First, we perform instance segment in the 2D bounding box to filter out the background clutters. Then, we collect 3D points of the target based on 3D-to-2D projection.

Specifically, let $\mathbf{M}$ be the projection matrix from 3D point cloud to 2D image plane, Ω_{box} be the collection of foreground pixels of the 2D bounding box and Ω_{ptc} be the collection of 3D points in the point cloud. We collect 3D points of the target by:

$$\mathcal{P} = \{\mathbf{h} \mid \forall\, \mathbf{h} \in \Omega_{ptc}, \text{if } \rho(\mathbf{h}; \mathbf{M}) \in \Omega_{box}\}, \tag{8}$$

where $\rho(\mathbf{h}; \mathbf{M})$ projects the input 3D point $\mathbf{h}$ to 2D pixel using the input projection matrix $\mathbf{M}$. For computation efficiency, similar to [60], we define a 3D frustum search space according to the 2D bounding box and then project 3D points to image plane within the search space. We use $\mathcal{P}_t^v$ to denote the 3D points of detection $\mathcal{B}_t^v$, the 3D location l_t^v of $\mathcal{B}_t^v$ is obtained by averaging the points in $\mathcal{P}_t^v$, *i.e.*, $l_t^v = \text{average}(\mathcal{P}_t^v)$.

3.4. Data Association

We use $\mathcal{T}_{t-1} = \{\mathcal{T}_{t-1}^1, ..., \mathcal{T}_{t-1}^{K_{t-1}}\}$ to denote the collection of trajectories at time $t-1$, where K_{t-1} is the number of trajectories. Each trajectory $\mathcal{T}_{t-1}^i$ is made of a serious tuples:

$$\mathcal{T}_{t-1}^i = \{(a_k^i, b_k^i, l_k^i), k \in \varphi_{t-1}^i\}, \tag{9}$$

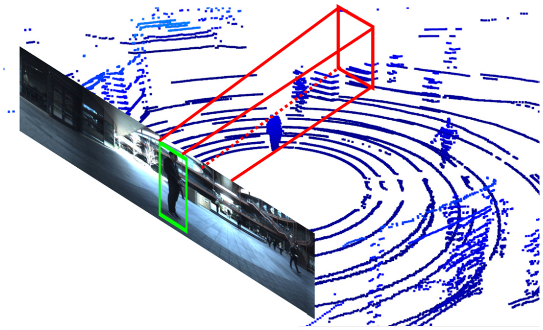


Figure 5. Illustration of multimodality data fusion.

where φ^i_{t-1} is the time index set of the trajectory $\mathcal{T}^i_{t-1}$, a^i_k, b^i_k, l^i_k are the appearance feature, bounding box and location of the target at time k. Taking the bounding box collection $\mathcal{B}_t = \{\mathcal{B}^1_t, ..., \mathcal{B}^{Q_t}_t\}$, where Q_t is the number of detections at time. The objective of data association is to match the newly obtained bounding boxes $\mathcal{B}_{t-1}$ with the existing trajectories $\mathcal{T}_{t-1}$, and manage target trajectories according to the matching result.

We formulate the data association as a bipartite graph matching problem, where we first compute a pairwise trajectory-detection affinity matrix between trajectoreis and detections and then solve the matching problem using the Hungarian algorithm [37].

3.4.1 Affinity Measurement

We use $\mathbf{A} \in \mathbb{R}^{K_{t-1}\times Q_t}_{\geq 0}$ to denote the pairwise affinity matrix of $\mathcal{T}_{t-1}$ and $\mathcal{B}_t$, where each element $\mathbf{A}(u, v)$ in $\mathbf{A}$ denotes the affinity between $\mathcal{T}^u_{t-1}$ and $\mathcal{B}^v_t$. The larger $\mathbf{A}(u, v)$ is, the higher affinity of $\mathcal{T}^u_{t-1}$ and $\mathcal{B}^v_t$ is. We compute the affinity $\mathbf{A}(u, v)$ score of each trajectory-detection pair using the appearance, motion and 3D location information:

$$\mathbf{A}(u, v) = \psi_{app}(\mathcal{T}^u_{t-1}, \mathcal{B}^v_t)+\psi_{mot}(\mathcal{T}^u_{t-1}, \mathcal{B}^v_t)+\psi_{loc}(\mathcal{T}^u_{t-1}, \mathcal{B}^v_t), \tag{10}$$

where $\psi_{app}(\cdot,\cdot)$, $\psi_{mot}(\cdot,\cdot)$ and $\psi_{loc}(\cdot,\cdot)$ compute the appearance, motion and location affinity of the input trajectory and detection, respectively.

Appearance similarity. The appearance similarity is computed by an averaged cross-correlation between the trajectory and detection appearance features, which can be written as:

$$\varphi_{app}(\mathcal{T}^u_{t-1}, \mathcal{B}^v_t) = \frac{\sum_{\forall k \in \tau^u_{t-1}} \left[e^{k-t} \cdot \gamma(a^u_k, \phi(\mathcal{B}^v_t))\right]}{\sum_{\forall k \in \tau^u_{t-1}} e^{k-t}}, \tag{11}$$

where τ^u_{t-1} is the collection of time index of trajectory $\mathcal{T}^u_{t-1}$, $\phi(\cdot)$ is a feature extractor and $\gamma(\cdot,\cdot)$ outputs the cross-correlation score of the input features.

Motion affinity. The motion affinity is calculated by computing the Intersection-over-Union (IoU) between a predicted bounding box $\mathcal{O}^i_t$ and detection $\mathcal{B}^v_t$:

$$\varphi_{mot}(\mathcal{T}^u_{t-1}, \mathcal{B}^v_t) = \text{area}(\mathcal{O}^u_t \cap \mathcal{B}^v_t)/\text{area}(\mathcal{O}^u_t \cup \mathcal{B}^v_t), \tag{12}$$

where $\mathcal{O}^u_t = \Phi(\mathcal{T}^u_{t-1})$ is a predicted bounding box according to the input trajectory $\mathcal{T}^u_{t-1}$ using the Kalman filter [30].

Location proximity. We calculate the 3D location proximity between a trajectory $\mathcal{T}^u_{t-1}$ and a detection $\mathcal{B}^v_t$ by:

$$\varphi_{loc}(\mathcal{T}^u_{t-1}, \mathcal{B}^v_t) = \sum_{k \in \tau^u_k} \frac{\sigma_t(k, t) \cdot \sigma_l(\mathcal{T}^u_{t-1}(k)_{loc}, l^v_t)}{|\tau^u_k|}, \tag{13}$$

where τ^u_{t-1} is the time index set of trajectory $\mathcal{T}^u_{t-1}$, $\mathcal{T}^u_{t-1}(k)_{loc}$ is the 3D location of trajectory $\mathcal{T}^u_{t-1}$ at time k. The $\sigma_t(\cdot,\cdot)$ and $\sigma_l(\cdot,\cdot)$ output normalized time distance and location distance using two RBF kernels, respectively.

3.4.2 Bipartite Graph Matching

Given the trajectory-detection affinity matrix $\mathbf{A}$, we aim to calculate a matching matrix $\mathbf{X}_t \in \{0, 1\}^{K_{t-1}\times Q_t}$ according to $\mathbf{A}$. Each element $\mathbf{X}(u, v)$ in $\mathbf{X}$ corresponds to the matching (*i.e.*, $\mathbf{X}$(u,v)=1) and non-matching (*i.e.*, $\mathbf{X}$(u,v)=0) between trajectory $\mathcal{T}^u_{t-1}$ and detection $\mathcal{B}^v_t$. The bipartite graph matching can be solved by the following optimization problem:

$$\begin{aligned} \mathbf{X}^* = \underset{\mathbf{X}}{\text{argmax}}\ & ||\mathbf{A} \odot \mathbf{X}||_2, \\ s.t. \quad \forall\, u, & \sum \mathbf{X}(u, :) \leq 1, \\ \forall\, v, & \sum \mathbf{X}(:, v) \leq 1, \end{aligned} \tag{14}$$

where $\odot$ denotes element-wise matrix multiplication and $||\cdot||_2$ outputs the L2-norm of input matrix. The constraints ensure the mutual exclusion of trajectories, where each detection will be occupied with at most one trajectory. On this basis, the optimization problem can be efficiently solved by the Hungarian algorithm [37].

3.5. Trajectory Inference

In this module, we manage the target trajectories according to the matching matrix $\mathbf{X}$. As shown in Figure 6, there are three conditions of the matching results: a) detection $\mathcal{B}^v_t$ does not match with any trajectories, *i.e.*, $\sum \mathbf{X}(:, v) = 0$. b) Trajectory $\mathcal{T}^u_{t-1}$ is matched with detection $\mathcal{B}^v_t$, *i.e.*, $\mathbf{X}(u, v) = 1$. c) Trajectory $\mathcal{T}^u_{t-1}$ does not match with any detections, *i.e.*, $\sum \mathbf{X}(u, :) = 0$. In the following, we provide detailed descriptions of the trajectory management of different matching results.

If $\sum \mathbf{X}(:, v) = 0$, *i.e.*, the detection $\mathcal{B}^v_t$ does not match with any existing trajectories. This indicates the detection is either a new occurred target or a false positive (FP) detection. Similar to [77], we initialize a "tentative" trajectory $\mathcal{T}^i_t$ using $\mathcal{B}^v_t$:

$$\mathcal{T}^i_t = \{(\phi_{app}(\mathcal{B}^v_t), \phi_{box}(\mathcal{B}^v_t), \phi_{loc}(\mathcal{B}^v_t))\}, \tag{15}$$

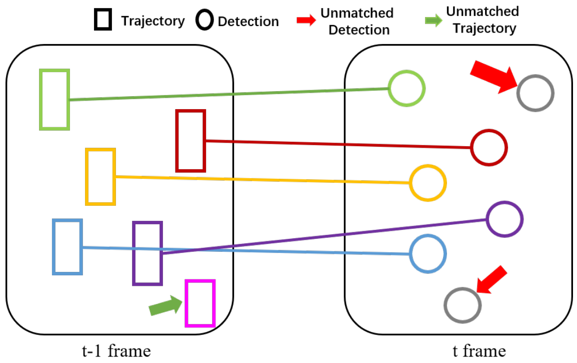


Figure 6. Illustration of data association. The rectangles and circles denote trajectories and detections, respectively. The lines indicate the linkage between trajectories and detections.

where $\phi_{app}(\cdot)$, $\phi_{box}(\cdot)$ and $\phi_{loc}(\cdot)$ output the appearance feature, bounding box and 3D location of the input detection, respectively. If the trajectory $\mathcal{T}_t^i$ is matched with detections in coming image frames, the $\mathcal{T}_t^i$ is then converted into a "confirmed" trajectory. Otherwise, we remove $\mathcal{T}_t^i$ from trajectory list.

If $\mathbf{X}(u,v)=1$, the detection $\mathcal{B}_t^v$ is assigned to trajectory $\mathcal{T}_{t-1}^u$. We extend the trajectory $\mathcal{T}_{t-1}^u$ using $\mathcal{B}_t^v$:

$$\mathcal{T}_t^u=\mathcal{T}_{t-1}^u\cup\{(\phi_{app}(\mathcal{B}_t^v),\phi_{box}(\mathcal{B}_t^v),\phi_{loc}(\mathcal{B}_t^v))\}. \quad (16)$$

If $\sum\mathbf{X}(u,:)=0$, *i.e.*, the trajectory $\mathcal{T}_{t-1}^u$ does not match with any detections, the target is temporally occluded or leaves the scene. If the trajectory is matched again in a following temporal window (such as within a coming 30 frames), we consider the target reappear after occlusion. Otherwise, we remove $\mathcal{T}_{t-1}^u$ from trajectory list.

4. Experiment

4.1. Dataset

We evaluate our proposed method on the JRDB dataset [54]. This dataset contains over 60K data from 5 stereo cylindrical panorama RGB cameras and two Velodyne 16 LiDAR sensors. There are 54 sequences of 64 minutes captured from both indoor and outdoor environments, where 27 sequences are used for training and the others are for testing. The frame rate is at 15 FPS and the resolution is 752×480. The dataset has over 2.3 million annotated 2D bounding boxes on 5 camera images and 1.8 million annotated 3D cuboids of over 3,500 targets.

4.2. Implementation Details

We follow the cascade detection paradigm [7] and adopt a ResNet50 [25] as the backbone of our detector. We split the panorama image into 7 image slices with an overlap 0.2 along the image width, and the ground-truth annotations to different image slices accordingly. We augment the training data by mixup [86] and multiscale augment. During training, the parameters of the detector are updated using an Adam optimizer [34] with a total number of 20 epochs, and the initial learning is set to 10^{-5}. In the tracking, we adopt a ReID model [49] as our feature extractor, which is pre-trained on the DukeMTMC dataset [65] using a triplet loss.

Table 1. Detection results on the JRDB Dataset

Method	AP ↑	Runtime ↓
YOLOV3 [63]	41.73	0.051
DETR [9]	48.51	0.350
RetinaNet [46]	50.38	0.056
Faster R-CNN [64]	52.17	**0.038**
Ours	**67.88**	0.070

Table 2. Tracking Results On the JRDB Dataset

Method	MOTA ↑	IDS ↓	FP ↓	FN ↓
Tracktor [1]	19.7	7026	79573	681672
DeepSORT [77]	23.2	**5296**	78947	650478
JRMOT [70]	22.5	7719	**65550**	667783
Ours	**31.7**	5742	67171	**580565**

4.3. Evaluation Result

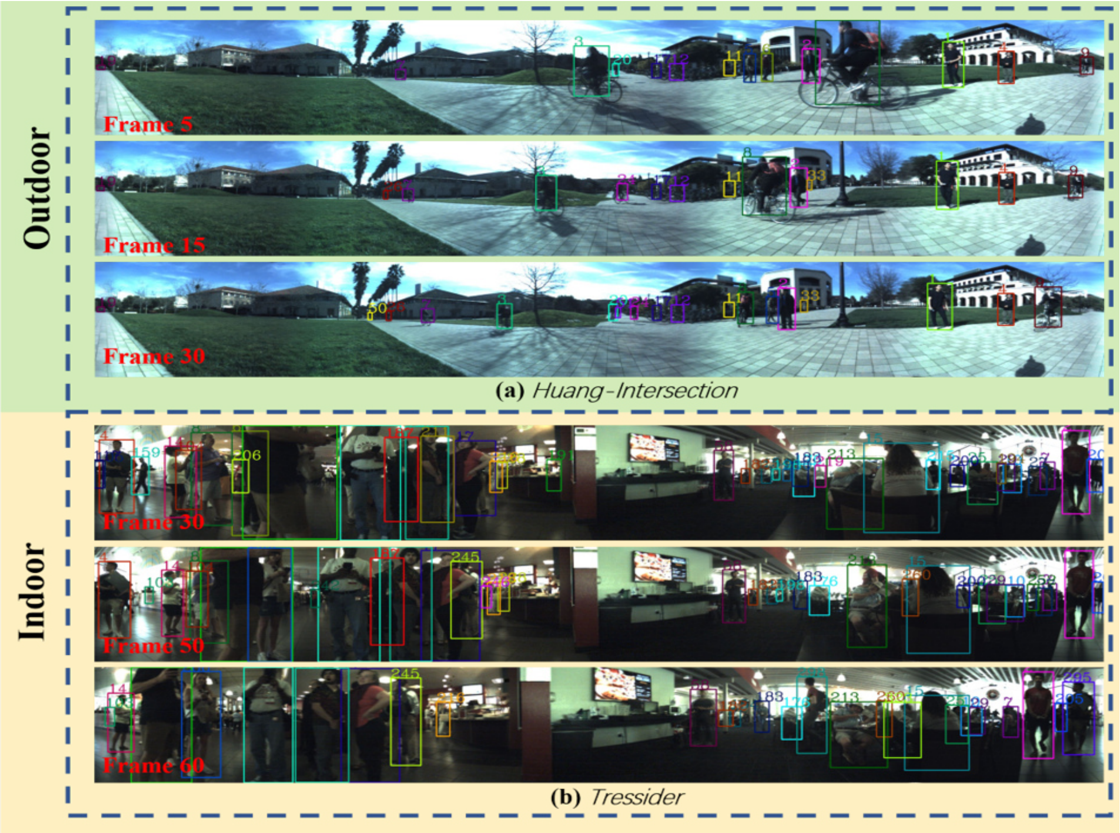
We compare the detection and tracking performance of our MMPAT on the JRDB dataset with state-of-the-art methods. For detection, we evaluate the proposed method in terms of Average Precision (AP ↑) and processing time (Runtime ↓). For tracking, we evaluate the MMPAT in terms of Multi-Object Tracking Accuracy (MOTA ↑) [2], IDentity Switch (IDS ↓), False Positive (FP ↓), and False Negative (FN ↓). The ↑ indicates the higher is better, and ↓ is on the contrary.

Table 1 shows the detection results. We can see that, the proposed method significantly outperforms the other state-of-the-art method by a large margin (at least 15.7 improvement on AP) with a competitive processing speed (about 14 frames per second). This is a strong evidence that demonstrates the proposed detection algorithm is efficient for object detection in panorama image. In Table 2, compared with state-of-the-art method JRMOT, the proposed method significantly improves the tracking performance by a large margin (9.2 improvement on MOTA) by reducing IDS and FN number (reduce 25% and 13% IDS and FN, respectively), while slightly worse on FP.

Figure 7 illustrates some qualitative tracking results of the MMPAT on JRDB dataset. It can bee seen that, no matter in outdoor scenario with poor light conditions or in indoor scene with complex background clutters, the proposed method can robustly track targets and generate accurate trajectories for targets.

4.4. Ablation Study

In Table 3, we provide ablation studies on the validation set of the JRDB dataset to analyze the influence of different components. The cascade r-cnn [7] is adopted as



(a) *Huang-Intersection*

(b) *Tressider*

Figure 7. Qualitative tracking results of (a) Huang-intersection (b) Tressider subsets of the JRDB dataset, where (a) is an outdoor scene and (b) is an indoor scene. The numbers upon bounding boxes denote ID labels.

Table 3. Ablation Study on Object Detection

Method	AP ↑
Baseline	52.8
Baseline+DCN	53.1
Baseline+DCN+split	64.6
Baseline+DCN+split+mixup	68.2
Baseline+DCN+split+mixup+multiscale	69.7
Baseline+DCN+split+mixup+multiscale+softnms	70.7

the `Baseline`. The `DCN` stands for deformable convolutional network [17], `split` denotes splitting the panorama image into image slices, `mixup` denotes data mixing up, `multiscale` denotes multiscale testing, `softnms` denotes using the softnms. It can be seen that: 1) compared with the baseline method, our detector dramatically improves the detection performance by a large margin (17.9 improvement on AP). 2) The split of image can efficiently improves detection in panorama image (11.5 improvement on AP). This demonstrates that the response regions of targets in the feature maps can largely affect the detection performance. 3) Adding different data augmentation methods can steadily improves the performance.

5. Conclusion

This paper focuses on the multi-object tracking (MOT) problem of automatic driving and robot navigation. We propose a MultiModality PAnoramic multi-object Tracking framework (MMPAT), which takes both 2D 360° panorama images and 3D point clouds as input. An object detection mechanism is designed to detect targets in panorama images. Besides, we also provide a 3D points collection algorithm to associate the point clouds with 2D images. We evaluate the proposed method on the JRDB dataset, which achieves the top performance in detection and tracking tasks and significantly outperforms state-of-the-art methods by a large margin (15.7 improvement on AP and 8.5 improvement on MOTA).

6. Acknowledge

This work is funded by National Key Research and Development Project of China under Grant No. 2019YFB1312000 and 2020AAA0105600, National Natural Science Foundation of China under Grant No. 62076195, 62006183, and 62006182, and by China Postdoctoral Science Foundation under Grant No. 2020M683489.

References

[1] Philipp Bergmann, Tim Meinhardt, and Laura Leal-Taixe. Tracking without bells and whistles. *arXiv preprint arXiv:1903.05625*, 2019. 3, 7

[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] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms–improving object detection with one line of code. In *Proceedings of the IEEE international conference on computer vision*, pages 5561–5569, 2017. 4, 5

[4] Guillem Brasó and Laura Leal-Taixé. Learning a neural solver for multiple object tracking. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6247–6257, 2020. 2

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

[6] Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020. 2

[7] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6154–6162, 2018. 4, 7

[8] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: High quality object detection and instance segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, page 11, 2019. 1

[9] Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. End-to-end object detection with transformers. In *European Conference on Computer Vision*, pages 213–229. Springer, 2020. 7

[10] Florian Chabot, Mohamed Chaouch, Jaonary Rabarisoa, Céline Teuliere, and Thierry Chateau. Deep manta: A coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2040–2049, 2017. 3

[11] Xiaozhi Chen, Kaustav Kundu, Ziyu Zhang, Huimin Ma, Sanja Fidler, and Raquel Urtasun. Monocular 3d object detection for autonomous driving. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2147–2156, 2016. 3

[12] Xiaozhi Chen, Kaustav Kundu, Yukun Zhu, Huimin Ma, Sanja Fidler, and Raquel Urtasun. 3d object proposals using stereo imagery for accurate object class detection. *IEEE transactions on pattern analysis and machine intelligence*, 40(5):1259–1272, 2017. 3

[13] Xiaozhi Chen, Huimin Ma, Ji Wan, Bo Li, and Tian Xia. Multi-view 3d object detection network for autonomous driving. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1907–1915, 2017. 3

[14] Peng Chu, Heng Fan, Chiu C Tan, and Haibin Ling. Online multi-object tracking with instance-aware tracker and dynamic model refreshment. In *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 161–170. IEEE, 2019. 2, 3

[15] Peng Chu and Haibin Ling. Famnet: Joint learning of feature, affinity and multi-dimensional assignment for online multiple object tracking. *arXiv preprint arXiv:1904.04989*, 2019. 3

[16] Qi Chu, Wanli Ouyang, Hongsheng Li, Xiaogang Wang, Bin Liu, and Nenghai Yu. Online multi-object tracking using cnn-based single object tracker with spatial-temporal attention mechanism. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4836–4845, 2017. 3

[17] Jifeng Dai, Haozhi Qi, Yuwen Xiong, Yi Li, Guodong Zhang, Han Hu, and Yichen Wei. Deformable convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 764–773, 2017. 4, 8

[18] 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 *2017 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1355–1361. IEEE, 2017. 3

[19] 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. 4

[20] Kuan Fang, Yu Xiang, Xiaocheng Li, and Silvio Savarese. Recurrent autoregressive networks for online multi-object tracking. In *2018 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 466–475. IEEE, 2018. 3

[21] James Ferryman and Ali Shahrokni. Pets2009: Dataset and challenge. In *2009 Twelfth IEEE international workshop on performance evaluation of tracking and surveillance*, pages 1–6. IEEE, 2009. 1

[22] Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361. IEEE, 2012. 2

[23] Mei Han, Wei Xu, Hai Tao, and Yihong Gong. An algorithm for multiple object trajectory tracking. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pages I–I. IEEE, 2004. 2

[24] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 2961–2969, 2017. 1, 2

[25] 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. 4, 7

[26] Yuhang He, Zhiheng Ma, Xing Wei, Xiaopeng Hong, Wei Ke, and Yihong Gong. Error-aware density isomorphism reconstruction for unsupervised cross-domain crowd counting. AAAI, 2021. 2

[27] Yuhang He, Xing Wei, Xiaopeng Hong, Weiwei Shi, and Yihong Gong. Multi-target multi-camera tracking by tracklet-to-target assignment. *IEEE Transactions on Image Processing*, 29:5191–5205, 2020. 1

[28] Andrea Hornakova, Roberto Henschel, Bodo Rosenhahn, and Paul Swoboda. Lifted disjoint paths with application in multiple object tracking. *arXiv preprint arXiv:2006.14550*, 2020. 2

[29] Hou-Ning Hu, Qi-Zhi Cai, Dequan Wang, Ji Lin, Min Sun, Philipp Krahenbuhl, Trevor Darrell, and Fisher Yu. Joint monocular 3d vehicle detection and tracking. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 5390–5399, 2019. 3

[30] Rudolph Emil Kalman. A new approach to linear filtering and prediction problems. 1960. 6

[31] Margret Keuper, Siyu Tang, Bjorn Andres, Thomas Brox, and Bernt Schiele. Motion segmentation & multiple object tracking by correlation co-clustering. *IEEE transactions on pattern analysis and machine intelligence*, 2018. 2

[32] Chanho Kim, Fuxin Li, Arridhana Ciptadi, and James M Rehg. Multiple hypothesis tracking revisited. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4696–4704, 2015. 2

[33] Chanho Kim, Fuxin Li, and James M Rehg. Multi-object tracking with neural gating using bilinear lstm. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 200–215, 2018. 2

[34] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 7

[35] Jason Ku, Melissa Mozifian, Jungwook Lee, Ali Harakeh, and Steven L Waslander. Joint 3d proposal generation and object detection from view aggregation. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1–8. IEEE, 2018. 3

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

[37] Harold W Kuhn. The hungarian method for the assignment problem. *Naval research logistics quarterly*, 2(1-2):83–97, 1955. 6

[38] 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. 3

[39] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 12697–12705, 2019. 3

[40] Hei Law and Jia Deng. Cornernet: Detecting objects as paired keypoints. In *15th European Conference on Computer Vision, ECCV 2018*, pages 765–781. Springer Verlag, 2018. 1, 2

[41] Bo Li, Tianlei Zhang, and Tian Xia. Vehicle detection from 3d lidar using fully convolutional network. *arXiv preprint arXiv:1608.07916*, 2016. 3

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

[43] Peiliang Li, Tong Qin, et al. 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. 4

[44] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 7345–7353, 2019. 3

[45] Ming Liang, Bin Yang, Shenlong Wang, and Raquel Urtasun. Deep continuous fusion for multi-sensor 3d object detection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 641–656, 2018. 3

[46] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision*, pages 2980–2988, 2017. 2, 7

[47] Qiankun Liu, Qi Chu, Bin Liu, and Nenghai Yu. Gsm: Graph similarity model for multi-object tracking. 3

[48] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C. Berg. Ssd: Single shot multibox detector. *ECCV*, 2016. 1

[49] Hao Luo, Youzhi Gu, Xingyu Liao, Shenqi Lai, and Wei Jiang. Bag of tricks and a strong baseline for deep person re-identification. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 7

[50] 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. 4

[51] Liqian Ma, Siyu Tang, Michael J Black, and Luc Van Gool. Customized multi-person tracker. In *Asian Conference on Computer Vision*, pages 612–628. Springer, 2018. 2

[52] 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/CVF International Conference on Computer Vision*, pages 6851–6860, 2019. 3

[53] Zhiheng Ma, Xing Wei, Xiaopeng Hong, and Yihong Gong. Bayesian loss for crowd count estimation with point supervision. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 6142–6151, 2019. 2

[54] Roberto Martín-Martín, Hamid Rezatofighi, Abhijeet Shenoi, Mihir Patel, JunYoung Gwak, Nathan Dass, Alan Federman, Patrick Goebel, and Silvio Savarese. Jrdb: A dataset and benchmark for visual perception for navigation in human environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2019. 2, 7

[55] James Munkres. Algorithms for the assignment and transportation problems. *Journal of the society for industrial and applied mathematics*, 5(1):32–38, 1957. 3

[56] Milind Naphade, Shuo Wang, David C Anastasiu, Zheng Tang, Ming-Ching Chang, Xiaodong Yang, Liang Zheng,

Anuj Sharma, Rama Chellappa, and Pranamesh Chakraborty. The 4th ai city challenge. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 626–627, 2020. 1

[57] 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

[58] Jinlong Peng, Tao Wang, Weiyao Lin, Jian Wang, John See, Shilei Wen, and Erui Ding. Tpm: Multiple object tracking with tracklet-plane matching. *Pattern Recognition*, page 107480, 2020. 2

[59] Alex D Pon, Jason Ku, Chengyao Li, and Steven L Waslander. Object-centric stereo matching for 3d object detection. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 8383–8389. IEEE, 2020. 3

[60] Charles R Qi, Wei Liu, Chenxia Wu, Hao Su, and Leonidas J Guibas. Frustum pointnets for 3d object detection from rgb-d data. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 918–927, 2018. 3, 5

[61] Charles R Qi, Hao Su, Kaichun Mo, and Leonidas J Guibas. Pointnet: Deep learning on point sets for 3d classification and segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 652–660, 2017. 3

[62] Zengyi Qin, Jinglu Wang, and Yan Lu. Monogrnet: A geometric reasoning network for monocular 3d object localization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 8851–8858, 2019. 3

[63] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. *arXiv preprint arXiv:1804.02767*, 2018. 7

[64] 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. 1, 4, 7

[65] Ergys Ristani, Francesco Solera, Roger Zou, Rita Cucchiara, and Carlo Tomasi. Performance measures and a data set for multi-target, multi-camera tracking. In *European Conference on Computer Vision*, pages 17–35. Springer, 2016. 7

[66] Amir Sadeghian, Alexandre Alahi, and Silvio Savarese. Tracking the untrackable: Learning to track multiple cues with long-term dependencies. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 300–311, 2017. 2, 3

[67] 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

[68] Hao Sheng, Jiahui Chen, Yang Zhang, Wei Ke, Zhang Xiong, and Jingyi Yu. Iterative multiple hypothesis tracking with tracklet-level association. *IEEE Transactions on Circuits and Systems for Video Technology*, 2018. 2

[69] Hao Sheng, Yang Zhang, Jiahui Chen, Zhang Xiong, and Jun Zhang. Heterogeneous association graph fusion for target association in multiple object tracking. *IEEE Transactions on Circuits and Systems for Video Technology*, 2018. 2

[70] 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 preprint arXiv:2002.08397*, 2020. 7

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

[72] 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/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 0–0, 2019. 3

[73] Siyu Tang, Mykhaylo Andriluka, Bjoern Andres, and Bernt Schiele. Multiple people tracking by lifted multicut and person re-identification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3539–3548, 2017. 2

[74] Zhi Tian, Chunhua Shen, Hao Chen, and Tong He. Fcos: Fully convolutional one-stage object detection. *arXiv preprint arXiv:1904.01355*, 2019. 1

[75] Gaoang Wang, Yizhou Wang, Haotian Zhang, Renshu Gu, and Jenq-Neng Hwang. Exploit the connectivity: Multi-object tracking with trackletnet. In *Proceedings of the 27th ACM International Conference on Multimedia*, pages 482–490, 2019. 2

[76] 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/CVF Conference on Computer Vision and Pattern Recognition*, pages 8445–8453, 2019. 3

[77] Nicolai Wojke, Alex Bewley, and Dietrich Paulus. Simple online and realtime tracking with a deep association metric. In *2017 IEEE international conference on image processing (ICIP)*, pages 3645–3649. IEEE, 2017. 2, 3, 6, 7

[78] Yu Xiang, Alexandre Alahi, and Silvio Savarese. Learning to track: Online multi-object tracking by decision making. In *Proceedings of the IEEE international conference on computer vision*, pages 4705–4713, 2015. 2

[79] Bin Xu and Zhenzhong Chen. Multi-level fusion based 3d object detection from monocular images. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2345–2353, 2018. 3

[80] Jiarui Xu, Yue Cao, Zheng Zhang, and Han Hu. Spatial-temporal relation networks for multi-object tracking. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3988–3998, 2019. 2, 3

[81] 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. 3

[82] Fan Yang, Wongun Choi, and Yuanqing Lin. Exploit all the layers: Fast and accurate cnn object detector with scale de-

pendent pooling and cascaded rejection classifiers. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2129–2137, 2016. 1

[83] Min Yang, Yuwei Wu, and Yunde Jia. A hybrid data association framework for robust online multi-object tracking. *IEEE Transactions on Image Processing*, 26(12):5667–5679, 2017. 2

[84] Amir Roshan Zamir, Afshin Dehghan, and Mubarak Shah. Gmcp-tracker: Global multi-object tracking using generalized minimum clique graphs. In *European Conference on Computer Vision*, pages 343–356. Springer, 2012. 2

[85] 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. 3

[86] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 7

[87] Xiaosong Zhang, Fang Wan, Chang Liu, Rongrong Ji, and Qixiang Ye. FreeAnchor: Learning to match anchors for visual object detection. In *Neural Information Processing Systems*, 2019. 1

[88] Yang Zhang, Hao Sheng, Yubin Wu, Shuai Wang, Weifeng Lyu, Wei Ke, and Zhang Xiong. Long-term tracking with deep tracklet association. *IEEE Transactions on Image Processing*, 29:6694–6706, 2020. 2

[89] Yin Zhou and Oncel Tuzel. Voxelnet: End-to-end learning for point cloud based 3d object detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4490–4499, 2018. 3

[90] Ji Zhu, Hua Yang, Nian Liu, Minyoung Kim, Wenjun Zhang, and Ming-Hsuan Yang. Online multi-object tracking with d-ual matching attention networks. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 366–382, 2018. 3