3D Object Tracking Methods with the Focus on Infrastructure sensors

Cavit Cakir, cavit.cakir@tum.de
Jianzhe Liu, jianzhe.liu@tum.de
Rui Xiao, rui.xiao@tum.de

***Abstract*—In recent years, a lot of techniques for autonomous driving have already been developed and tested. Current computer vision techniques with onboard sensors can provide satisfactory object detection, re-identification, and tracking results in various scenarios. However, it is inevitably limited by perspective and occlusions. Infrastructure sensors can solve these challenges because they are more flexible in a matter of perspective, location, and pose. Therefore, infrastructure-based computer vision techniques can improve the performance of detection and tracking in challenging situations. In this paper, recent progress and state-of-the-art methods for infrastructure sensors-based 3D object tracking methods are comprehensively reviewed. To further investigate trends in 3D Object Tracking, we have covered different single object tracking (SOT) and multi-object tracking (MOT) methods. In addition to methods, we presented AI City Challenge as a significant competition based on multi object tracking. We underlined current challenges and anticipated future trends such as cooperative perception and sensor fusion methods.**

***Index Terms*—Infrastructure-sensor based Tracking, MCMT**

I. Introduction

The rapid advancement of the transportation system has increased the efficiency of daily commuting and the movement of goods. The increase in the number of vehicles in the traffic comes with several critical issues in terms of efficiency, safety, and reliability [1] [2]. Taking advantage of improvements in technology, several sensors on roads can be used in traffic to automate and observe vehicles. Advanced sensors, wireless information transfer, and artificial intelligence are enabling the automation of vehicles. With the infrastructure sensors, vehicles in the traffic are not only observing the other vehicles but also other road users such as pedestrians and cyclists. Different types of sensors installed on infrastructure systems are capable of detecting traffic conditions in a complicated and dense traffic environment, which makes Vehicle-to-everything (V2X) possible. V2X is a way for vehicles to communicate with vehicles in traffic and every other entity. Some application areas of sensors are: cameras that can provide high-quality video data to detect and track different traffic objects such as vehicles, pedestrians, and cyclists [3]. LiDAR can provide highly accurate 3D point cloud data to find the precise 3D location of traffic objects [4]. Infrastructure-based detection systems have the potential to achieve better object detection and tracking performance with a large number of sensors and greater flexibility in terms of mounting height and pose.

This survey report reviews recently published infrastructure-based object detection and tracking articles & workshops. It aims to show recent research and innovations. Currently, in AI City Challenge, the best tracking papers use the tracking-by-detection paradigm and use traffic rules and zones to match tracklets also in other benchmarks this trend follows [5], [6], [7]. Tracking-by-detection is a two-step approach, that starts with the detection of the object and is followed by tracking, various methods for detection and tracking are explained in section V. In March 2022, Bai et al. compared infrastructure-based detection and tracking methods but it does not include the newest methods [8]. Our main contribution is to collect and compare the newest novelty methods in 3D object tracking with infrastructure sensors. The structure of our survey is as follows: Mentioning state-of-the-art methods and current challenges. Continued with explaining and comparing articles that have proposed different novelties and proven themselves in certain competitions and conferences, and finally, the conclusion.

II. Fundamentals

To better understand the topic of infrastructure-based 3D Object Tracking, we first need to start by introducing some fundamental concepts related to 3D object tracking.

A. 3D Object Detection:

3D object detection serves as the foundation for many tracking algorithms. In many 3D object tracking algorithms, object detection is the crucial step before tracking in order to identify targets. There exist various 3D object detection methods while nowadays researchers mainly focus on the popular deep learning-based methods. Categorized by the inputs, 3D Object Detection can be divided into point cloud-based and image-based methods. As for point cloud input, the detector either perform Voxelization [9] or BEV (Bird's Eye View) [10] Projection for the upcoming process. While for image input, the detector can take information from monocular or multiple cameras. The final step of detection is usually predicting a 3D Bounding Box, which can be further modified in the tracking process.

B. Vehicle ReID:

When performing multi-object multi-camera tracking, it would be a challenge to re-identify the same vehicle appearing in different cameras at a different time step. As for ReID, various CNNs [11] can be applied to perform feature extraction and matching between images from multiple cameras. Another

frontier that ReID is focusing on is the construction of a proper dataset.

C. 3D Object Tracking:

3D Object Tracking can be performed on LiDARs [12] or Cameras [11] or both [10]. In a point cloud sequence, 3D object tracking aims to predict the location and orientation of an object in the current search point cloud given a template point cloud. In terms of the number of tracked objects, 3D object tracking can be also defined as SOT (Single Object Tracking) and MOT (Multi-Object Tracking). While Multi-object tracking (MOT) enables mobile robots or autonomous vehicles to perform well-informed motion planning and navigation by localizing surrounding objects in 3D space and time.

III. Related Works

Infrastructure-based 3D object tracking methods have also been studied by other survey papers. For example, Bai [8] reviews the infrastructure-based object detection and tracking approaches with an analysis of details in literature, with a focus on LiDAR-based perception methods. Bai also introduces a classical tracking pipeline, which mainly consists of; Background Filtering, Clustering, Classification, and Tracking. After the introduction of LiDAR-based tracking methods, Bai also summarizes the current datasets, which include general datasets, roadside datasets, and simulators. At the end of his paper, he points out that cooperative perception and multi-sensor fusion should be considered future trends in the field.

Datondji et al. [13], comprehensively discussed camera-based 3D tracking methods. In this paper, the authors first give an introductory overview of general vision-based vehicle monitoring approaches. Subsequently, they present a review of studies regarding vehicle detection and tracking in intersection-like scenarios. Then, they focus on camera-based roadside monitoring systems, with special attention to omnidirectional setups. Finally, they present some research directions which are likely to improve the performance of vehicle detection and tracking at intersections.

IV. Datasets

In this part, we will make some brief introductions to the datasets that will be mentioned in the methods of tracking, with further information in Table I.

A. KITTI

The KITTI dataset [14] is currently the largest computer vision algorithm evaluation dataset in the world for autonomous driving scenarios and the pioneering multi-modal dataset providing dense point clouds from a LiDAR sensor. This dataset is used to evaluate the performance of computer vision techniques such as visual odometry, 3D object detection, and 3D tracking in the vehicle environment.

B. Waymo

The Waymo dataset [15] contains a large number of high-quality, manually annotated 3D ground truth bounding boxes for the LiDAR data, and 2D tightly fitting bounding boxes for the camera images. All its annotations were created and subsequently reviewed by trained labelers using production-level labeling tools. Using, and selecting the test set scenes from a geographical holdout area will allow the user to evaluate how well models that were trained on the dataset generalize to previously unseen areas.

C. NuScenes

NuScenes [16] represents a large leap forward in terms of data volumes and complexities and is the first dataset to provide 360◦ sensor coverage from the entire sensor suite. It is also the first multi-modal dataset that contains data from nighttime and rainy conditions, with object attributes and scene descriptions in addition to object class and location. It enables research on multiple tasks such as object detection, tracking, and behavior modeling in a range of conditions.

D. DAIR-V2X

DAIR-V2X dataset [17] is the first large-scale, multi-modality, multi-view dataset for VICAD (Vehicle-Infrastructure Cooperative Autonomous Driving). The dataset covers 10 km of city roads, 10 km of highway, 28 intersections, and 38 km^2 of driving regions with diverse weather and lighting variations with annotations.

E. MS coco

MS coco (Microsoft Common Objects in Context) [18] is a large-scale image dataset with annotations you can use to train machine learning models to recognize, label, and describe objects. MC COCO provides the following types of annotations: Object detection, Captioning, Keypoints, “Stuff image” segmentation, Panoptic and Dense pose.

F. CityFlow

CityFlow [19] is the largest-scale dataset in terms of spatial coverage and the number of cameras/videos in an urban environment. Camera geometry and calibration information are provided to aid Spatial-temporal analysis. In addition, a subset of the benchmark is made available for the task of image-based vehicle re-identification (ReID).

G. CityFlowV2

CityFlowV2 [19] has the same validation as the test set of the original CityFlow dataset. This dataset contains 3.58 hours (215.03 minutes) of videos collected from 46 cameras spanning 16 intersections in a mid-sized U.S. city. The distance between the two furthest simultaneous cameras is 4 km. The dataset covers a diverse set of location types, including intersections, stretches of roadways, and highways. The dataset is divided into 6 scenarios. 3 of the scenarios are used for training, 2 are for validation, and the remaining 1 is for testing. In total, the dataset contains 313931 bounding boxes for 880

Name	Year	View	Image	Pointcloud	Boxes	Classes
KITTI [14]	2012	single vehicle	15k	15k	200k	8
Waymo Open [15]	2019	single vehicle	1M	200k	12M	4
NuScenes [16]	2019	single vehicle	1.4M	400k	1.4M	23
DAIR-V2X [17]	2021	vehicle-infrastructure cooperative	71k	71k	1.2M	10
MScoco [18]	2014	-	328k	-	-	80(objects)+91(stuff)
CityFlow [19]	2019	Multiple-camera	3.25 hours videos	-	314k	-
CityFlowV2 [19]	2021	Multiple-camera	3.25 hours videos	-	314k	-

TABLE I: commonly used datasets categorized by features

distinct annotated vehicle identities. Only vehicles passing through at least 2 cameras have been annotated. The resolution of each video is at least 960p and the majority of the videos have a frame rate of 10 FPS. Additionally, in each scenario, the offset from the start time is available for each video, which can be used for synchronization.

V. Methods of single and multiple object tracking

In this part, we will introduce several recently proposed methods for object tracking and their novelties. Though in the AI City Challenge 2021&2022 most of the top-ranked papers are focusing Multi-Target tasks, to make sure that the coverage of the articles is more comprehensive, we will introduce both single and multiple target tracking methods.

A. LiDAR Single Object Tracking

LiDAR is used more often for target detection, for its data mainly includes point clouds, depth maps, etc. Yet the point cloud model is of great significance for target tracking. From this point, although the practical value and application of single-target tracking technology are not as good as other tracking technologies nowadays, the technology of single-target tracking using LiDAR is still developing in recent years.

For example, PTTR (Point Tracking Transformer) [12] is a novel 3D point cloud tracking model that efficiently predicts high-quality 3D tracking results in a coarse-to-fine manner with the help of transformer operations. The methodology of PTTR contains three parts: feature extraction, feature matching, and coarse-to-fine tracking prediction. Firstly, both search point clouds and template point clouds are fed into the feature extraction part, which is based on a Point-Net++ [20] backbone. However, the D-FPS (Distance Farthest-Point Sampling) [20] strategy used in PointNet++ tends to generate uniformly-distributed sample points, which often leads to important information loss during the sampling process. To minimize information loss, a novel Relation Aware Sampling strategy is put forward and has proven to show significant advantages over the traditional sampling strategy. Secondly, the feature matching part would process the data by exploiting a novel Point-Relation Transformer Module. The Point-Relation Transformer Module is based on a Relation Attention Module and it performs a self-attention operation and cross-attention operation, which generate a coarse prediction. Thirdly, the prediction refinement part focuses on generating the final prediction by processing coarse predictions. This part involves an offset operation, local pooling, and catenating matched features with pooled features. The PTTR model is tested on the KITTI [14] dataset. However, since KITTI [14] has only a limited size, the author constructs a novel Waymo SOT Dataset [12] based on the Waymo [15] Open Dataset, which dataset is of a significantly larger scale with a more balanced class distribution than KITTI. In terms of success and precision (two metrics that are adopted), the PTTR model has shown significantly lower computational complexity with an incredibly lightweight design and has outperformed previous state-of-the-art methods in both KITTI [14] and Waymo SOT Dataset [15].

Another method for Single Object Tracking that is worth mentioning is called Motion-Centric Paradigm, which also concentrates on performing Single Object Tracking with LiDAR sensors. A novel motion-centric paradigm is proposed in this method to provide a brand-new insight into solving 3D single object tracking problems. The proposed method distinguishes itself from the well-known traditional Siamese paradigm, which emphasizes appearance matching techniques and faces down-grade performance when encountering textureless and incomplete 3D LiDAR point clouds. These shortcomings could to some extent be avoided by the motion-centric paradigm. The main novelty of this method is the whole motion-centric paradigm including an M2–Track [21], which is a two-stage tracker. The detailed description of the paradigm (and also the tracker) is demonstrated as follows: First of all, similar to a data-preprocessing step, target segmentation with spatial-temporal learning takes previous 3D bounding boxes and point clouds as input, and produces target point clouds as output. Secondly, the M2-Tracker reaches stage 1, which is called "Motion-Centric BBox Prediction" using a multi-layer perceptron algorithm. At this stage, the tracker can get the current target BBox (Bounding Box). Thirdly, the M2-Tracker reaches its second and final stage, which is named" BBox Refinement with Shape Completion", this process is based on an original novel motion-assisted shape completion strategy. As the output of the second stage, we're getting a regressed RTM (Relative Target Motion) and a refined current bounding box (β_t). As for Datasets, the model is tested based on KITTI [14], NuScenes [16], and WOD [15]. The results show that compared with previous paradigms such as SC3D, P2B, and BAT, the M2-Track shows better performance in terms of two evaluation metrics (both success and precision). It's interesting that this method further demonstrates the possibility of combining the motion-centric paradigm with the traditional appearance matching paradigm, which shows incredible po-

tential for the improvement of tracking performance. The authors believe that the motion-centric paradigm can serve as a primary principle to guide future architecture designs. The general pipeline of LiDAR-based object tracking is shown in Fig.1. As we can see from the figure, a typical LiDAR-based object tracking method takes a 3D point cloud as the input, and go through data pre-processing, feature extraction, preliminary results generation, and post-processing step. In the end, a refined prediction is made.

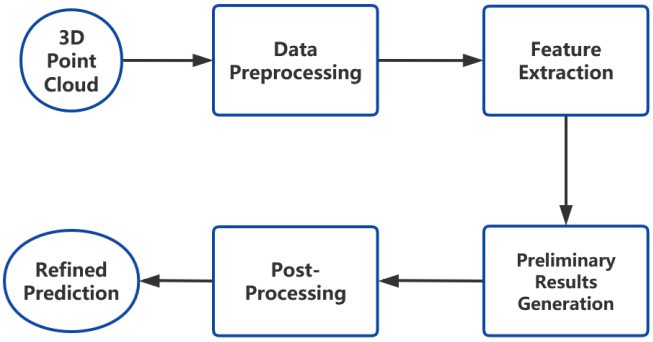

Fig. 1: Pipeline for the LiDAR-based object tracking.

B. LiDAR Multi-Object Tracking

As mentioned in the above section, LiDAR achieves relatively good performance on SOT (Single Object Detection). Despite that, researchers manage to exploit LiDARs to conduct tracking in MOT (Multi-Object Tracking). For instance, after receiving input from the 3D point cloud, Weng [22] uses a straightforward combination of a 3D Kalman-Filter [23] and the Hungarian Algorithm for state estimation and data association. His method became the state-of-the-art 3D MOT performance on KITTI [14] and nuScenes [16] in 2020. However, the 3D Kalman-Filter method [22] was quick overtaken by some deep-learning-based methods. One of the most outstanding deep-learning-based methods is the CentrePoint [24], a two-stage tracking model put forward by Yin in 2021. As we know, typical backbones like VoxelNet [25] and PointPillars [26] convert 3D point cloud input into a map-view feature map. Yin's model takes the map-view feature map as input and predicts a 3D bounding box based on object centers. The first stage of CentrePoint is to answer "where are the center point and 3D bounding box?" It exploits a 2D CNN architecture detection head to find object centers and predict 3D bounding boxes with center features. The second stage is to answer this question: "how confident is my prediction and how can I refine it?" Following [27], Yin passes them into a Multi-Layer Perceptron to predict a confidence score w.r.t each 3D bounding box. At last, he trains the model with L1 loss to gain box refinement. CentrePoint became the state-of-the-art 3D MOT performance on the nuScenes dataset [16] in 2021, which was then overtaken by fusion-based methods, which fuse the input from both LiDAR and camera, to achieve better tracking performance.

C. Fusion-Based Multi-Object Tracking

With their competitiveness in combing advantages from LiDARs and Cameras, fusion-based methods have recently taken up top positions on tracking leaderboards. Especially on the nuScenes [16] tracking dataset, the fusion-based method has become the state-of-the-art method. As an early fusion-based method, EagerMOT [28] adopts a two-stage data association module. First, they associate data from different sensor modalities. Second, they manage to update track states even when only partial information is provided. With the two-stage data association method, their model has been proven to be more robust to false negatives resulting from different sensor modalities. EagerMOT achieved great performance on both KITTI [14] and nuScenes [16] tracking datasets and outperformed some LiDAR-based MOT methods such as the aforementioned CentrePoint [24]. Outperforming EagerMOT, AlphaTrack [7], which was published on IROS (International Conference on Intelligent Robots and Systems) 2021. The main novelty of AlphaTrack consists of two parts: 1. A cross-modal fusion scheme is proposed to fuse the camera appearance feature with the LiDAR feature to facilitate 3D detection and tracking. 2. An additional branch is attached to the 3D detector, which leads to significant improvement in tracking performance. Different from previous methods, AlphaTrack fully exploits appearance and location information to perform joint 3D object detection and tracking, outperforming previous models such as PointNet [29]. The general pipeline of Fusion-Based Tracking is shown in Fig.2. Depending on when to fuse the data, we can categorize it to be erarly fusion, deep fusion and late fusion.

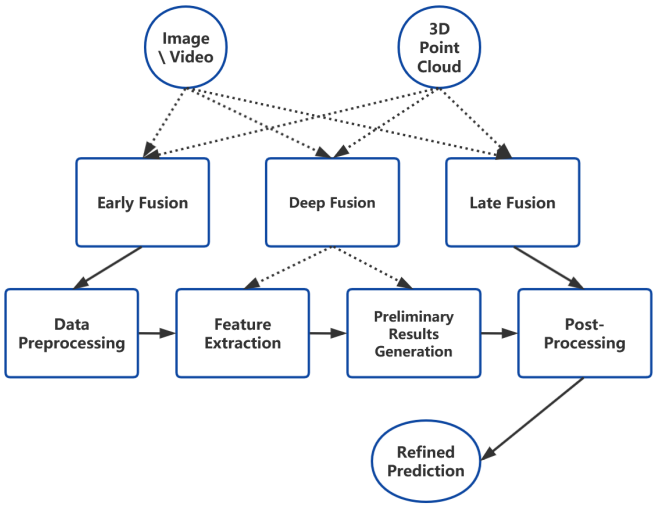

Fig. 2: Pipeline for the Fusion-Based Multi-Object Tracking.

D. Multi Camera Multi Object Tracking

Multi-Camera Tracking Multi-Target (MCMT) is a very popular but also challenging task due to unreliable object detection, heavy occlusion, low resolution, and varying lighting and viewing-perspective conditions. For vehicles, this task

becomes more challenging because;
I. Vehicles may stop for a long time at the traffic signs and continually be occluded among each other, which makes occlusion even more severe;
II. Inter-class similarity for vehicles is higher because there may exist many different identities with a similar appearance.
In the following parts, we will introduce several methods that are proposed for AI City Challenge and their novelty to the MCMT problem. The general pipeline of Multi-Camera Multi-Object tracking is also shown in Fig.3. As shown in the figure, a general pipeline of Multi-Camera Multi-Object Tracking should consist of four parts: Object Detection, ReID, Single-Camera Multi-Target Tracking, and Cross-Camera Trajectory Association. The following methods all follow similar procedures as in the general pipeline.

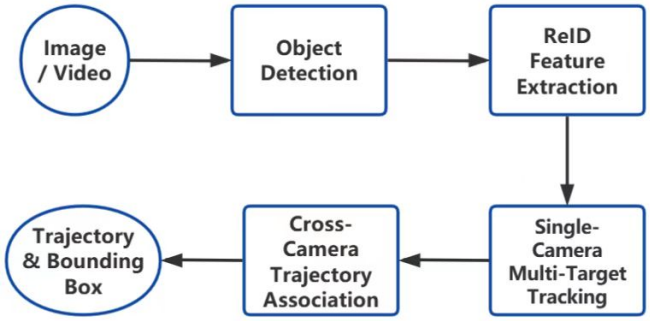

Fig. 3: Pipeline for the Multi-Camera Multi-Object Tracking.

1) Spatial-Temporal Filtering: Regarding the topic, the first method that we introduce [11] puts forwards a system that concentrates on the task of 3D object tracking using multiple sensors. The sensors used in this method are cameras instead of LiDARs. The methodology of this multi-camera vehicle tracking system consists of three steps: object detection on the video frame, multi-object tracking on a single camera, and feature matching under multiple cameras according to spatial and temporal information. The detailed process is as follows: After the preprocessing stage of object detection on a video frame, the system performs multi-object tracking on a single camera based on DeepSORT [30], which is a well-known detection-based tracking paradigm. Then the process of vehicle re-identification is executed, including a CNN model (based on ResNet [31]) to obtain features of appearance and a training model using aggregation loss. After vehicle re-identification, the system reaches it is last stage, feature matching with multiple cameras, by proposing a novel low-cost MCMT strategy, with the assistance of a spatial filter and a temporal filter.
The novelty of this method lies in three parts; 1: It proposes a powerful vehicle ReID model which is robust against specific conditions. 2: It puts forwards a novel time-GPS-fusing strategy that significantly improves accuracy by cross-camera vehicle matching. 3: It proposes a novel technique to filter the optimal frame to avoid mismatching between cross-cameras. As for the datasets, the backbone networks are pretrained by COCO [18] and ImageNet [32]. Besides, the system is tested on training and evaluation data from AI City Challenge 2021 Track 3. Experiments are conducted w.r.t different sub-tasks: MTSC, Query Image Selection, Vehicle ReID, and MCMT, all showing reliably satisfying results. This system was ranked 6th on the leaderboard of AI City Challenge 2021.

2) Box-Grained Reranking Matching: The next method for MCMT is called Box-Grained Reranking Matching [5]. As already mentioned, the main challenges of the MCMT are heavy occlusion and appearance variance caused by various camera perspectives and congested vehicles, they proposed a practical framework for dealing with city-scale MCMT tasks. This framework consists of four modules.

The first two modules are vehicle detection and ReID feature extraction. The location of all vehicles and extraction of the appearance features for all cameras is done in these two modules. They used the state-of-art object detection framework Cascase-RCNN [33]. Feature Pyramid Network (FPN) [34] is followed by backbone to increase semantic features information at each level in the extracted features. They trained this model with COCO [18] pretrained weights and data of track1 2022 AI City Challenge. They used ensemble of HRNet [35], ResNeXt101 [36], ResNet [31], Res2Net [37] and ConvNeXt [38] as backbone for ReID training.

The third module is Single-Camera Multi-Target tracking (SCMT). In this module, tracking multiple vehicles to generate candidate trajectories within each camera according to the detected boxes and extracted appearance features. They use the tracking-by-detection paradigm. They adopt the classic tracker DeepSORT [30] as their baseline method and improved its various techniques. DeepSORT uses the Kalman filter [39] & Hungarian algorithm [40] combination. In order not to miss potential targets, they set thresholds of filter detection results as BYTETrack [41] does. One of the problems is the velocity of the cars changes sharply, the Kalman filter is unable to predict correct states. To solve frequent track ID switches, they refine the tracking results with offline re-link. Furthermore, They run the tracker on the video frames one time in the forward direction, one time in the backward direction, and merge the tracked targets to generate complete trajectories, the recall can be further improved. The fourth module, Inter-Camera Association (ICA) is to associate all candidate trajectories between two successive cameras using the K-reciprocal nearest neighbors algorithm, and combine all successively matched trajectories for final results. There are several challenges in this module, vehicles with similar appearances in the matching pool of tracklet candidates, the different locations of cameras, and some objective factors like illumination, and perspective. They propose a novel box-grained matching module to find the same identities at the box level successively and sequentially. This module contains several methods, Firstly, 'Zone-based Tracklet Candidates Filter' is to roughly filter out tracklet candidates with traffic rules, road structures, and traveling time. Secondly, 'Box-grained Distance matrix Construction and Optimization' is to calculate

the box-grained distance matrix. Third, 'Tracklet Association with k-reciprocal Nearest Neighbors' is used for associating tracklets between two connected zones with the distance matrix D. They propose a novel and effective matching strategy to find all the convincing pairs. All tracklets are associated with the principle of k-reciprocal nearest neighbors. Lastly, 'Post-processing after matching' is to check the validity of all matched pairs and assign a global id if the two pairs share the same tracklet. The proposed method is tested on the public test set of the 2022 AI City Challenge Track1. It achieves IDF1 of 84.86%, ranking 1st on the leaderboard.

3) Multi-Camera Vehicle Tracking System: The next method for the MCMT task is a Vehicle Tracking System. [42]. This is an accurate system that is composed of 4 parts.

The first part is State-of-the-art detection and re-identification models for vehicle detection and feature extraction. They followed the tracking-by-detection paradigm and used the state-of-the-art network YOLOv5 [43], which pretrained on the COCO dataset [18] and they tune the detection classes to only cars, trucks, and buses. For the ReID part, they used the ensemble model of ResNet50-IBN-a [31], ResNet101-IBN-a [31], and ResNeXt101-IBN-a [31] which was pretrained on the CityFlow dataset [19].

The second part is Single camera tracking, where they introduce augmented tracks prediction and multi-level association method on top of the tracking-by-detection paradigm. They followed Simple Online and Realtime Tracking (SORT) [44] and improved it with various methods. Firstly, Kalman Filter often produces ID switches when the direction of movement changes. To improve it, they utilized two more Single Object Tracking (SOT), Efficient Convolution Operators (ECO) [45] and MedianFlow [46], and propose an augmented tracks prediction method. Then they include vehicle appearance features, which then go through a feature dropout filter and a multi-level matching process. Finally, to make sure the completeness of tracklets, they added another post-process for tracklet merging within a single camera.

Third part is Zone-based single-camera tracklet merging strategy [Fig 4]. To select tracklets, they divide crossroad images into 9 effective zones and 1 traffic zone, which can be determined by specific cases. Before merging, They pick some tracklets under the criteria:

- Tracklet that starts normally and ends in either the same zone or middle zone.
- Tracklet that starts in either the middle zone or traffic zone. These tracklets are thought to be abnormal and will become candidates for merging. From there, these candidates will go to tracklet merging in the next step. They cope with abnormal tracklet fragments using hierarchical clustering. The clustering to get tracklets under the same cluster:

1. Sort tracklets by their starting frame in ascending order.
2. Check if two tracklets agree with space and time.

Using the above techniques, tracklets fragments can be selected and merged under the same cluster, yielding more accurate tracklet results for single-camera tracking.

The last part is the Multi-camera spatial-temporal matching

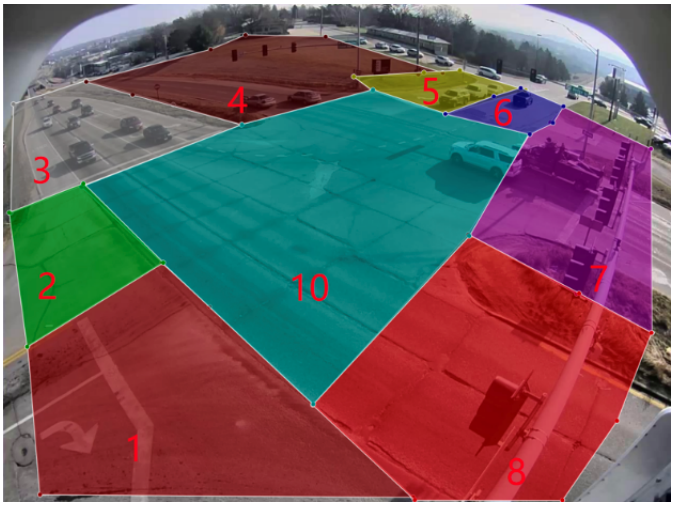

Fig. 4: Crossroad images are divided into 9 effective zones to enable the singe-camera tracklet merging strategy. [42]

and clustering strategy. Their approach consists of selection, aggregation, and clustering steps. The cosine similarity matrix is used in multi-camera matching. They used the GPS location of each camera, and they simplified their proposed zones. They also take speed limits and traffic light signals into account. For the tracklet clustering, they proposed two rounds, the first round is directional based and the second round is to aggregate together the same vehicles from adjacent cameras. They also proposed an iterative searching strategy that effectively solves the edge cases like U-turns. The proposed method is tested on the public test set of the 2022 AI City Challenge Track1. It achieves IDF1 of 84.37%, ranking 2nd on the leaderboard.

4) Space-Time-Appearance Features: The following [47] MTMCT method consists of object detection and re-identification (ReID), single-camera tracking, cross-camera trajectory association.

For the detection task, they used the YOLOv5x1 [43] model which is pre-trained on the COCO dataset. As for the ReID task, they retrain the models as their ReID feature extractor following the work proposed by Luo et al. Two challenges are considered and addressed in this method: (1) low-confidence objects could be missed without extra data annotation, and (2) precise association of trajectories from different cameras is affected by multiple factors. For the first challenge, a cascaded tracking method based on detection, appearance features, and trajectory interpolation is proposed, exploiting potential real targets in low-confidence objects to improve detection and identification recall. ByteTrack [41] is a state-of-art method that mines the real target from the low confidence box sufficiently to improve the tracking performance. They use a cascaded matching strategy. First, associate the high confidence box with ReID features, then the unmatched trackers are associated with boxes by IoU. Lastly, they match the low confidence boxes with IoU to enhance the stability of tracking. The Kalman filter [39] is used for track updating.

For the second challenge, space, time, and appearance

features are proposed to be the most crucial factors for trajectory association, so a zone-gate and time-decay-based matching mechanism is proposed to adjust the original appearance matrix to link tracklets more precisely from different cameras. Due to the similar appearance, the ambiguity of the cropped image, and the numerous candidates in the gallery, directly using appearance features for ID clustering faces many challenges. They cluster each pair of adjacent cameras separately and then extend the clustering results to the entire scene chain. The zone-gate mechanism is proposed as, From Fig 5, it can be seen that for intersection (camera) N, there are a total of 12 driving routes for vehicles. For all of these driving routes, if the tracklet under this camera needs to associate with the next intersection N + 1, it must pass through zone 3 and 4; similarly, if it needs to associate with the previous intersection N-1, it must go through zone 7 and 8.

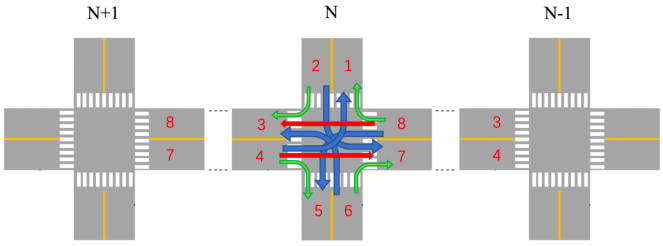

Fig. 5: Total of 12 driving routes for vehicles that associate the crossroad N and its neighbors. [47]

Time-Decay Strategy is inspired by humans, as they take into account elapsed time as an important factor when trying to identify a target across different cameras. They also did trajectory post-processing, they design an interpolated post-processing module for interrupted trajectories. The proposed method is tested on the public test set of the 2022 AI City Challenge Track1. It achieves IDF1 of 83.71%, ranking 3rd on the leaderboard.

5) Vehicle Counting Based on CenterTrack: This method [48] is based on footage analysis, which is captured with traffic cameras by counting the number of vehicles performing various predefined motions of interest. They proposed this method based on the CenterTrack object detection and tracking neural network used in conjunction with a simple IoU-based tracking algorithm. Since there are no annotations of vehicle bounding boxes or tracks provided for this challenge, a model pre-trained on the MS COCO dataset [18] is used. Usually, CenterTrack is based on a single-stage key-point-based object detection network CenterNet, which is well-suited for applications with limited computational resources. Compared to the pure object detection network, CenterTrack has an additional output for each detected object which denotes the displacement of the object from its previous position. So, with this information, they can then track the objects across multiple frames using a greedy algorithm based on the IoU metric.

Research in recent years has shown that a simple tracker based on the IoU metric of object bounding boxes in consecutive frames can outperform more complicated trackers when the objects are detected reliably. For example, a similar IoU-based strategy was already employed in 2020's AI City Challenge. In this research, CenterTrack will first use a simple greedy algorithm that associates objects in consecutive frames based on distances of bounding box centers. The distance is calculated between the center of the bounding box in the previous frame and the center of the bounding box in the current frame shifted by the displacement vector. This approach adds only a very small computational overhead over the base CenterNet architecture and is surprisingly effective. However, it is found that this approach is not suitable especially in very crowded scenes with vehicles of various sizes present. To remedy this, they then use the IoU metric of the object bounding boxes instead of the distance of the centers. Similar to the approach based on the centers, they use the displacement vector to shift the bounding box in the current frame. If detection does not have IoU greater than 0.1 with bounding boxes of any of the active tracks which have not had a new bounding box assigned to them for that frame so far, then they either discard it if has a confidence score lower than 0.4 or they create a new track if this threshold is met.

The proposed method is published on Track 1 of the 2021 AI City Challenge, and in the public evaluation server, it achieved the IDF1 score of 0.8449 and placed 8th place on the public leaderboard.

6) ReID and Camera Link Model: This method's [49] focus is an MCMT framework, which mainly consists of two innovations, i.e., traffic-aware single-camera tracking (TSCT) and the trajectory-based camera link model (CLM).

First, TSCT is proposed to handle the long-term occlusions created in the traffic scenarios. Usually, there will be a large number of isolated and fragmented vehicle trajectories, created from a single camera multi-target tracker, in the center of the frames where vehicles do not enter or exit the camera's field of view (FoV). For example, when a vehicle stops in front of a red traffic light, it can be partially or even fully occluded in the camera's FoV for a long time. This kind of zone can be called a traffic-aware zone. Here, they use the TrackletNet tracker (TNT) [50] , which is a superior SCT method in intelligent transportation system applications, as their single-camera tracker. According to this condition, TSCT is proposed to find out the traffic-aware zones, where this kind of occlusion happens, by clustering the start and end nodes of all the resulting trajectories from the TNT. Vehicle Re-ID in the single-camera is then implemented for these traffic-aware zones to connect these disconnected trajectories created in the traffic scenarios. Second, facing higher inter-class appearance similarity of distinct vehicles, trajectory-based CLM is further proposed to impose spatial-temporal constraints and reduce solution search space for the cross-camera ReID. For two different vehicles with very similar appearances, it is nearly impossible to re-identify them using a typical ReID method. However, taking advantage of the spatial and temporal

constraints between a pair of adjacently connected cameras, they can easily filter out the vehicles that are not likely to appear in a certain camera at a certain timestamp. They define these constraints, including the vehicle entry/exit zones and the transition times, as the CLM. Using the CLM automatically generated from the training data, cross-camera vehicle ReID becomes much more accurate and efficient.

Finally, a hierarchical clustering algorithm, based on the Euclidean distance between the feature space of different trajectories, is used to merge the trajectories among all the cameras to obtain the final MCMT results.

The proposed method is shown to be effective and robust. It also achieves a new state-of-the-art performance with IDF1 of 74.93% on the CityFlow dataset [19].

7) Semantic Attribute Parsing and Cross-Camera Tracklet Matching: This proposed method [50] focuses on the city-scale cross-camera vehicle tracking problem. As illustrated in Fig 6, to obtain a wide range of field of view (FOV) and reduce costs, the cameras are often placed far apart and their FOV is always non-overlapping. The target attributes such as appearance features and motion patterns of the same target could be significantly different in different cameras. Moreover, as the occurrences of each target under different

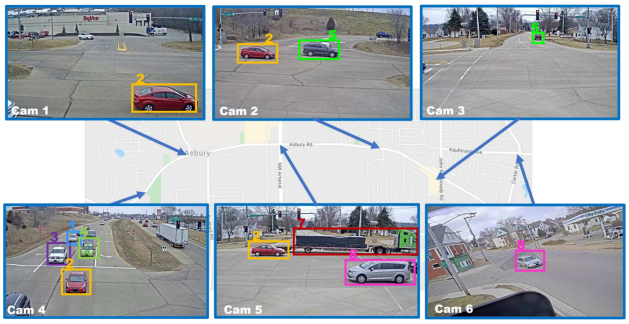

Fig. 6: The far apart placing of cameras has non-overlapping FOV thus difficult generating a complete global trajectory for each target.

cameras are different and unknown, it is difficult to solve the tracklet matching problem and generate a complete global trajectory for each target across all the cameras. To tackle these problems, they provide an efficient two-step approach to tracking multiple vehicles in a city-scale multi-camera network, which first generates local tracklets for all the targets under each camera, respectively, and then connects these local tracklets across different cameras to generate a complete global trajectory for each target.

More specifically, they first follow the tracking-by-detection paradigm to generate local tracklets for all the targets under each camera, respectively. Then they compute the affinity of local tracklets in different cameras by semantic attribute parsing, which produces a robust tracklet representation using a spatial-temporal attention mechanism and prunes false matching candidates by traffic topology reasoning. Taking the local tracklet affinity as input, the Tracklet-to-Target Assignment (TRACTA) [50] algorithm is exploited to solve the cross-camera tracklet matching problem, and the complete trajectory of each target across all the cameras is obtained by reconnecting the split local tracklets.

The proposed method is evaluated on the City-Scale Multi-Camera Vehicle Tracking task in the 2020 AI City Challenge and achieves the second-best result.

VI. Comparison

State-of-the-art tracking problems can be categorized variously. Besides, as mentioned in the above sections, there also exist various 3D object tracking models and paradigms that can achieve great performance in different metrics such as precision, success, efficiency, and so on. The summary of the aforementioned MTMC methods can be concluded in Table IV.

Following the evaluation metrics of P2B [52] and measure the “Success” and “Precision”. Specifically, “Success” is defined as the IoU between predicted boxes and the ground truth, and “Precision” measures the AUC (Area Under Curve) of the distance between prediction and ground truth box centers from 0 to 2 meters. The summary of the aforementioned LiDAR-based methods can be concluded in Table II.

Following the metrics of [53], we can show the evaluation results of the aforementioned LiDAR and fusion-based MOT methods tested on NuScenes test set III. For the evaluation metric, AMOTA stands for ”Average Multi-Object Tracking Accuracy” and AMOTP stands for ”Average Multi-Object Tracking Precision”, both representing tracking quality. ”False Positive” and ”False Negative” can be used to calculate Recall, AMOTA, and AMOTP. Along with ”ID Switch”, a smaller value of them, tends to mean better tracking results. Depending on the specific scenario, there could also be different proper solutions for specific tracking performance. Therefore, It’s quite difficult to compare and evaluate different 3D object tracking models and methods since most of them are trained and tested on different datasets. As a result, as mentioned in the above section, the occurrence of well-known datasets, such as the KITTI [14] tracking dataset, Waymo Open Dataset [15], NuScences [16], and Dair-V2X [17] dataset has provided us an opportunity to compare different tracking models under relatively similar conditions in terms of same metrics, so that we can tell which one achieves better performance. However, is there any much “stricter” comparison between different tracking models? A very good comparison is achieved by the “AI City Challenge”, which is a CVPR workshop competition that we’re going to introduce later. In this section, we’re going to discuss the similarities and differences of the methodologies that the aforementioned papers have used. If several papers are tested on the same dataset or simply developed in the same track of the competition, then their performance will be compared in terms of the same metrics.

Name	Year	Car	Pedestrian	Van	Cyclist	Avarage	Inference Time(ms)
P2b [52]	2020	56.2 / 72.8	28.7 / 49.6	40.8 / 48.4	32.1 / 44.7	39.5 / 53.9	23.6
PTTR [12]	2022	65.2 / 77.4	50.9 / 81.6	52.5 / 61.8	65.1 / 90.5	58.4 / 77.8	19.9
M^2-Track [21]	2022	**65.5 / 80.8**	**61.5 / 88.2**	**53.8 / 70.7**	**73.2 / 93.5**	**62.9 / 83.4**	-

TABLE II: Summary of aforementioned LiDAR-Based SOT methods on KITTI [14] benchmark . Success / Precision are used for evaluation.

Name	Year	Type	AMOTA(%)	AMOTP(%)	FP	FN	IDS
CentrePoint [24]	2020	LiDAR-based	63.8	55.5	18612	22928	760
EagerMOT [28]	2020	Fusion-based	67.7	55.0	**17705**	24925	1156
AlphaTrack [7]	2021	Fusion-based	70.4	**58.5**	18247	21126	718
BEVFusion [10]	2022	Fusion-based	**74.1**	40.3	19997	**19395**	**506**

TABLE III: Evaluation results of LiDAR-Based and Fusion-based MOT methods on NuScenes [16] test set. AMOTA and AMOTP follows the same definition as in comparison section. FP means "False Positive". FN means "False Negative". IDS means "ID Switch"

A. Comparison of LiDAR-based Methods

1) Coarse-to-fine Manner: Prediction. However, they both follow a "coarse-to-fine" manner, which is: firstly generating a coarse prediction of a 3D bounding box, then performing a certain type of prediction refinement method. Specifically, PTTR [12] performs local pooling for both search points and offset template points, then concatenates them to generate a final refined prediction. M^2-Track gets the coarse prediction and performs bounding box refinement with a motion-assisted shape completion strategy. Since both papers outperform many previous models, especially the ones without the prediction refinement step, we can reasonably assume that the prediction refinement step plays a crucial role in modern LiDAR-based methods. Methods along with a more accurate refinement step tend to perform better in 3D SOT tasks. Interestingly, among the aforementioned 3D MOT approaches, Yin [24] also adopts an MLP [27] approach as PTTR [12], so we assume that MLP could be a good try when dealing with prediction refinement problems.

2) Fusion-based Methods Tend to Perform Better: When comparing the results of fusion-based methods with those of Camera-based or LiDAR-based ones, we may find out that fusion-based methods usually have a better tracking performance than the other two methods. The difficulty when implementing fusion-based is that they require a more complex structure, to fuse data from cameras and LiDARs. Why does fusion-based achieve better performance? The reason could be: First, cross-modality data usually contain richer information which contributes to better 3D bounding box prediction. Second, after the coarse prediction of the 3D bounding box, data from different sensors can help to better refine the prediction.

B. Comparison of Camera-based methods

1) Tracking-by-Detection Paradigm: The papers of Yang [5] , Li [42], He [47] and Ren [11] follows basically the same overall structure. The overall structure of their models can be summarized as follows: 1. Vehicle Detection 2. ReID Feature Extraction 3. Single-Camera Multi-Object Tracking 4. Cross-Camera Trajectory Association. Furthermore, they have all applied a tracking-by-detection paradigm in the Single-Camera Multi-Object Tracking stage. However, they are proposing novel methods or models which often appear in the third and fourth part of the overall structure, which leads their models to different performances. To be more specific, in the Single-Camera Multi-Object Tracking stage: Yang [5] uses a DeepSORT [30] backbone with several modifications such as offline-relink, which contributes to solving the issue caused by frequent track ID switches. The main novelty of his work lies in a Box-grained distance matrix construction and optimization. Li [42] bases his tracking method on the existing SOFT models, but he proposes an augmented tracking prediction method by applying a feature-dropout filter and a multi-level matching process. To tackle the probable missing of low-confidence objects without extra data annotation, He [47] combines state-of-the-art ByteTrack and a novel cascaded matching strategy. Ren [11] uses DeepSORT [30], a detection-based-tracking paradigm for the tracking stage of his paper. As for the Cross-Camera Trajectory Association stage: Yang [5] uses the K-Reciprocal Nearest Neighbor algorithm as a matching strategy to find convincing pairs. Li [42] performs a 'zone-based single camera tracklet merging strategy'. To achieve precise association of trajectories from different cameras, He [47] develops zone-based and time-delay-based matching mechanisms, the main novelty lies in the time-decaying idea. Ren [11] has done feature matching under multiple cameras by proposing a low-cost MTMC strategy.

2) Use of Temporal and Spatial Information: The traditional cross-camera matching strategy which only exploits appearance feature matching has shown low efficiency and mismatching instance often occur. Consequently, we notice that several papers have proposed a time and space-related approach to improving matching precision. For example, Ren [11] uses a spatial and temporal filter in his matching strategy. Li [42] has adopted spatial and temporal information to perform clustering in his tracklet merging strategy. He [47] has also used the same information to make his association of trajectories under different cameras more precise. Judging from the test result, we can assume that with the introduction of extra spatial and temporal information the model can achieve better tracking performance.

Name	Year	Sensor	Method	Dataset	IDF1
Box-Grained Reranking [5]	2022	Camera	MCMT	CityFlowV2	0.8486
CenterTrack [48]	2021	Camera	MCMT	CityFlowV2	0.8449
Vehicle Tracking System [42]	2022	Camera	MCMT	CityFlowV2	0.8437
Space-Time-Appearance [47]	2022	Camera	MCMT	CityFlowV2	0.8371
Guided by Crossroad Zones [6]	2021	Camera	MCMT	CityFlowV2	0.8095
Spatial-Temporal Filtering [11]	2021	Camera	MCMT	CityFlowV2	0.5763
Graph Auto-Encoder [51]	2022	Camera	MCMT	CityFlow	0.7721
Reid camera link [49]	2020	Camera	MCMT	CityFlow	0.7493
SemanticAttribute Parsing [50]	2020	Camera	MCMT	CityFlow	0.4400

TABLE IV: Summary of Aforementioned MCMT Tracking Methods

VII. Discussion

In this part, we will briefly discuss the current challenges and future developing trends for the tracking tasks in autonomous driving considering the above sections.

A. Current challenges

Accurate and consistent vulnerable road user detection remains one of the most challenging perception tasks for autonomous vehicles. One of the most complex outstanding issues is partial occlusion [54], where a sensor has only a partial view of the target object due to a foreground object that partially obscures the target. Occlusion exists in various forms ranging from partial occlusion to heavy occlusion. In the automotive environment, target objects can be occluded by static objects such as buildings and lampposts. Dynamic objects such as moving vehicles or other road users may inter-occlude (occlude one another) such as in crowds, and self-occlude where parts of a pedestrian or cyclist overlap.

The frequency and variation of occlusion in the automotive environment are vast and can also be impacted by cultural and environmental factors [55]. To achieve the performance required for safety in autonomous driving an algorithm or set of algorithms, must consistently generalize to reach the state of the art performance in all benchmarks, cultures, and environmental conditions.

Additionally, and perhaps most challengingly, any successful approach must also have the computational efficiency to robustly identify objects in real time. The process of accurately assessing algorithm performance for the detection of partially occluded objects is a difficult one. There are a wide variety of test datasets available for object detection based on the desired target, environmental conditions, and sensing methods. Inconsistency between each dataset's definition and annotation level of occluded targets and the metrics used, present difficulties when attempting to accurately quantify performance.

B. Future trends

1) Cooperative Perception: Optical sensors and learning algorithms for autonomous vehicles have dramatically advanced in the past few years. Nonetheless, the reliability of today's autonomous vehicles is hindered by the limited line-of-sight sensing capability and the brittleness of data-driven methods in handling extreme situations. With recent developments in telecommunication technologies, cooperative perception with vehicle-to-vehicle communications has become a promising paradigm to enhance autonomous driving in dangerous or emergencies.

2) Sensor fusion: A multi-sensor-based perception system [56] has the potential to improve the perceiving performance by taking advantage of complementary sensor data with appropriate fusion techniques. An infrastructure-based perception system has more flexible conditions for multi-sensor equipment and is capable of empowering high-computational edge servers. Therefore, with the increasing application of deep learning, the relevant multi-sensor fusion strategy needs to be further improved, especially the serial network structure needs to be adjusted to adapt to the fusion of various sensor data. The purpose of multi-target tracking is to obtain the motion intention of the target and the environment reconstruction is to generate a safe driving area. However, it is difficult to do these tasks well with sensors alone, which need to be combined with location sensors, maps, and V2X. Besides, to form the perception ability to look around, AD vehicles need the cooperation of a variety of similar sensors, which need to cooperate closely with the whole system, and future work should explain further.

VIII. Conclusion

In this paper, We first introduced the current status of autonomous driving technology based on infrastructure sensors and expounded on its technological development and application scenarios. Then, we present the top-ranked studies in the 2021 and 2022 AI city challenges and introduce the novelties in these studies. Next, we compared the above-mentioned innovative methods with the databases used in those researches. At last, we analyze the challenges and future development trends of the current roadside sensor-based autonomous driving technology.

References

[1] National Center for Statistics and Analysis. Overview of motor vehicle crashes in 2019. National Highway Traffic Safety Administration. (Traffic Safety Facts Research Note. Report No. DOT HS 813 060). 2020

[2] INRIX, "INRIX: Congestion Costs Each American 97 hours, $1,348 A Year," Inrix, 2018. https://inrix.com/press-releases/scorecard-2018-us/

[3] L. Liu et al., "Deep Learning for Generic Object Detection: A Survey," International Journal of Computer Vision, vol. 128, no. 2, pp. 261–318, Oct. 2019, doi: 10.1007/s11263-019-01247-4.

[4] E. Arnold, O. Y. Al-Jarrah, M. Dianati, S. Fallah, D. Oxtoby, and A. Mouzakitis, "A Survey on 3D Object Detection Methods for Autonomous Driving Applications," IEEE Transactions on Intelligent Transportation Systems, vol. 20, no. 10, pp. 3782–3795, Oct. 2019, doi: 10.1109/tits.2019.2892405.

[5] X. Yang et al., "Box-Grained Reranking Matching for Multi-Camera Multi-Target Tracking," 2022.

[6] C. Liu et al., "City-Scale Multi-Camera Vehicle Tracking Guided by Crossroad Zones," 2022.

[7] Y. Zeng, C. Ma, M. Zhu, Z. Fan, and X. Yang, "Cross-modal 3D object detection and tracking for auto-driving," 2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2021.

[8] Z. Bai, G. Wu, X. Qi, Y. Liu, K. Oguchi, and M. J. Barth, "Infrastructure-Based Object Detection and Tracking for Cooperative Driving Automation: A Survey," arXiv:2201.11871, Mar. 2022.

[9] J. Deng, S. Shi, P. Li, W. Zhou, Y. Zhang, and H. Li, "Voxel R-CNN: Towards High Performance Voxel-based 3D Object Detection", AAAI, vol. 35, no. 2, pp. 1201-1209, May 2021.

[10] Liu, Zhijian, et al. "BEVFusion: Multi-Task Multi-Sensor Fusion with Unified Bird's-Eye View Representation." arXiv preprint arXiv:2205.135432.

[11] P. Ren et al., "Multi-Camera Vehicle Tracking System Based on Spatial-Temporal Filtering," 2021.

[12] C. Zhou et al., "PTTR: Relational 3D Point Cloud Object Tracking With Transformer," 2022.

[13] S. R. E. Datondji, Y. Dupuis, P. Subirats and P. Vasseur, "A Survey of Vision-Based Traffic Monitoring of Road Intersections," in IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 10, pp. 2681-2698, Oct. 2016, doi: 10.1109/TITS.2016.2530146.

[14] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," The International Journal of Robotics Research, vol. 32, no. 11, pp. 1231–1237, Aug. 2013, doi: 10.1177/0278364913491297.

[15] P. Sun et al., "Scalability in Perception for Autonomous Driving: Waymo Open Dataset," 2020.

[16] H. Caesar, V. Bankiti, and A. H. Lang, "nuScenes: A multimodal dataset for autonomous driving," 2019.

[17] H. Yu, Y. Luo, and M. Shu, "DAIR-V2X: A Large-Scale Dataset for Vehicle-Infrastructure Cooperative 3D Object Detection," 2022.

[18] T. Lin et al., "Microsoft COCO: Common Objects in Context", 2022.

[19] Z. Tang et al., "CityFlow: A City-Scale Benchmark for Multi-Target Multi-Camera Vehicle Tracking and Re-Identification," 2019.

[20] C. Qi, L. Yi, H. Su and L. Guibas, "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space", 2022.

[21] C. Zheng et al., "Beyond 3D Siamese Tracking: A Motion-Centric Paradigm for 3D Single Object Tracking in Point Clouds", 2022.

[22] X. Weng, J. Wang, D. Held and K. Kitani, "3D Multi-Object Tracking: A Baseline and New Evaluation Metrics", 2022.

[23] Trawny, Nikolas, Stergios I. Roumeliotis. Indirect Kalman filter for 3D attitude estimation. University of Minnesota, Dept. of Comp. Sci. Eng., Tech. Rep 2 (2005): 2005.

[24] T. Yin, X. Zhou, and P. Krahenbuhl, "Center-based 3D object detection and tracking," 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021.

[25] Y. Zhou and O. Tuzel, "VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection", 2022.

[26] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. CVPR, 2019.

[27] B. Jiang, R. Luo, J. Mao, T. Xiao and Y. Jiang, "Acquisition of Localization Confidence for Accurate Object Detection", 2022.

[28] A. Kim, A. Osep, and L. Leal-Taixe, "EagerMOT: 3D multi-object tracking via sensor fusion," 2021 IEEE International Conference on Robotics and Automation (ICRA), 2021.

[29] Qi, C.R., Su, H., Mo, K. and Guibas, L.J., 2017. Pointnet: C. Qi, H. Su, K. Mo and L. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation", 2022.

[30] 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

[31] K. He, X. Zhang, S. Ren and J. Sun, "Deep Residual Learning for Image Recognition", 2022.

[32] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional Neural Networks," Communications of the ACM, vol. 60, no. 6, pp. 84–90, 2017.

[33] 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.

[34] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 2117–2125, 2017.

[35] Jingdong Wang, Ke Sun, Tianheng Cheng, Borui Jiang, Chaorui Deng, Yang Zhao, Dong Liu, Yadong Mu, Mingkui Tan, Xinggang Wang, et al. Deep high-resolution representation learning for visual recognition. IEEE transactions on pattern analysis and machine intelligence, 43(10):3349– 3364, 2020.

[36] Saining Xie, Ross Girshick, Piotr Dollar, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 1492–1500, 2017

[37] Shang-Hua Gao, Ming-Ming Cheng, Kai Zhao, Xin-Yu Zhang, Ming-Hsuan Yang, and Philip Torr. Res2net: A new multi-scale backbone architecture. IEEE transactions on pattern analysis and machine intelligence, 43(2):652–662, 2019.

[38] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feicht- enhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s.

[39] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft. Simple online and realtime tracking. In 2016 IEEE International Conference on Image Processing (ICIP), pages 3464–3468, 2016

[40] H. W. Kuhn, "The Hungarian method for the assignment problem," Naval Research Logistics, vol. 52, no. 1, pp. 7–21, 2005.

[41] Y. Zhang, P. Sun, Y. Jiang, D. Yu, F. Weng, Z. Yuan, P. Luo, W. Liu, and X. Wang, "ByteTrack: Multi-object tracking by associating every detection box," 2022.

[42] F. Li et al., "Multi-Camera Vehicle Tracking System for AI City Challenge 2022," 2022.

[43] Glenn Jocher, "ultralytics/yolov5: v6.2 - YOLOv5 Classification Models, Apple M1, Reproducibility, ClearML and Deci.ai integrations". Zenodo, Aug. 17, 2022. doi: 10.5281/zenodo.7002879.

[44] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," 2016 IEEE International Conference on Image Processing (ICIP), 2016.

[45] M. Danelljan, G. Bhat, F. S. Khan, and M. Felsberg, "ECO: Efficient convolution operators for tracking," 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2017.

[46] Z. Kalal, K. Mikolajczyk, and J. Matas. Tracking-learning-detection. IEEE transactions on pattern analysis and machine intelligence, 34(7):1409–1422, 2011.

[47] H. Yao et al., "City-Scale Multi-Camera Vehicle Tracking based on Space-Time-Appearance Features," 2022.

[48] V. Kocur and M. Ftáčnik, "Multi-Class Multi-Movement Vehicle Counting Based on CenterTrack.", 2021

[49] H.-M. Hsu, Y. Wang, and J.-N. Hwang, "Traffic-aware multi-camera tracking of vehicles based on Reid and Camera link model," Proceedings of the 28th ACM International Conference on Multimedia, 2020.

[50] He, J. Han, W. Yu, X. Hong, X. Wei, and Y. Gong, "City-scale multi-camera vehicle tracking by semantic attribute parsing and cross-camera Tracklet matching," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2020.

[51] H.-M. Hsu, Y. Wang, J. Cai, and J.-N. Hwang, "Multi-target multi-camera tracking of vehicles by graph auto-encoder and self-supervised camera link model," 2022 IEEE/CVF Winter Conference on Applications of Computer Vision Workshops (WACVW), 2022.

[52] H. Qi, C. Feng, Z. Cao, F. Zhao, and Y. Xiao, "P2B: Point-to-box network for 3D object tracking in point clouds," 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[53] Caesar, H., Bankiti, V., Lang, A.H., Vora, S., Liong, V.E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G. and Beijbom, O., 2020. nuscenes: A multimodal dataset for autonomous driving. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 11621-11631).

[54] S. Gilroy, E. Jones and M. Glavin, "Overcoming Occlusion in the Automotive Environment—A Review", IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 1, pp. 23-35, 2021.

[55] Y. Yu, R. Vasudevan, and M. Johnson-Roberson, "Occlusion-aware risk assessment for autonomous driving in Urban Environments," IEEE Robotics and Automation Letters, vol. 4, no. 2, pp. 2235–2241, 2019.
[56] Z. Wang, Y. Wu, and Q. Niu, "Multi-sensor fusion in Automated Driving: A survey," IEEE Access, vol. 8, pp. 2847–2868, 2020.