Cyber Mobility Mirror: A Deep Learning-Based Real-World Object Perception Platform Using Roadside LiDAR

Zhengwei Bai, *Graduate Student Member, IEEE*, Saswat P. Nayak, Xuanpeng Zhao, Guoyuan Wu, *Senior Member, IEEE*, Matthew J. Barth, *Fellow, IEEE*, Xuewei Qi, *Member, IEEE*, Yongkang Liu, Emrah Akin Sisbot, *Member, IEEE*, and Kentaro Oguchi

***Abstract*— Object perception plays a fundamental role in Cooperative Driving Automation (CDA) which is regarded as a revolutionary promoter for next-generation transportation systems. However, the vehicle-based perception may suffer from the limited sensing range and occlusion as well as low penetration rates in connectivity. In this paper, we propose *Cyber Mobility Mirror* (*CMM*), a next-generation real-world object perception system for 3D object detection, tracking, localization, and reconstruction, to explore the potential of roadside sensors for enabling CDA in the real world. The CMM system consists of six main components: i) the data pre-processor to retrieve and preprocess the raw data; ii) the roadside 3D object detector to generate 3D detection results; iii) the multi-object tracker to identify detected objects; iv) the global locator to generate geo-localization information; v) the mobile-edge-cloud-based communicator to transmit perception information to equipped vehicles, and vi) the onboard advisor to reconstruct and display the real-time traffic conditions. An automatic perception evaluation approach is proposed to support the assessment of data-driven models without human-labeling requirements and a CMM field-operational system is deployed at a real-world intersection to assess the performance of the CMM. Results from field tests demonstrate that our CMM prototype system can achieve 96.99% precision and 83.62% recall for detection and 73.55% ID-recall for tracking. High-fidelity real-time traffic conditions (at the object level) can be geo-localized with a root-mean-square error (RMSE) of 0.69m and 0.33m for lateral and longitudinal direction, respectively, and displayed on the GUI of the equipped vehicle with a frequency of $3-4Hz$.**

***Index Terms*— Field operational system, 3D object detection, multi-object tracking, localization, deep learning, cooperative driving automation.**

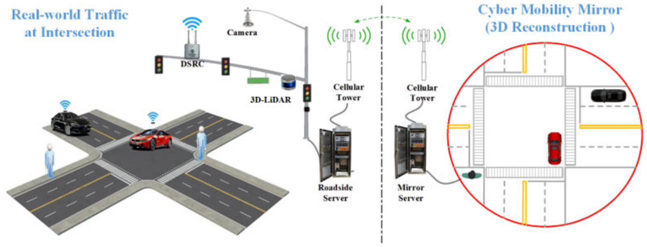

Real-world Traffic at Intersection
Camera
DSRC
3D-LiDAR
Cellular Tower
Roadside Server
Cellular Tower
Mirror Server
Cyber Mobility Mirror (3D Reconstruction)

Fig. 1. Illustration for CMM concept at an intersection scenario.

I. Introduction

WITH the rapid growth of travel demands, the transportation system is facing increasingly serious traffic-related challenges, such as improving traffic safety, mitigating traffic congestion, and reducing mobile source emissions. Taking advantage of recent strides in advanced sensing, wireless connectivity, and artificial intelligence, Cooperative Driving Automation (CDA) is attracting more and more attention over the past few years and is regarded as a transformative solution to the aforementioned challenges [1]. In the past few decades, several projects or programs have been conducted to explore the feasibility and potential of CDA. For instance, the California PATH program showed throughput improvement by a fully connected and automated platoon [2]. In the European DRIVE C2X project, the cooperative traffic system was assessed by large-scale field operational tests for various connected vehicle applications [3]. Recently, the U.S. Department of Transportation is leading the CARMA Program [4] for research on CDA, leveraging emerging capabilities in both connectivity and automation to enable cooperative transportation system management and operations (TSMO) strategies. Additionally, the Autonet2030 Program led by EUCar is working on Cooperative Systems in Support of Networked Automated Driving by 2030 [5]. However, most of the aforementioned projects assume an ideal scenario, i.e., all vehicles are connected and automated. Because the presence of mixed traffic (with different types of connectivity and levels of automation) would be the norm, in the long run, one of the popular ways to enhance CAVs' adaptability in such a complicated environment is to improve their situation-awareness capability. For example, vehicles are equipped with more and more high-resolution onboard sensors and upgraded with powerful onboard computers to better perceive the surroundings and make decisions by themselves, a similar path to highly automated vehicles (HAVs) [6]. However, this roadmap

Manuscript received 7 April 2022; revised 21 January 2023 and 1 April 2023; accepted 10 April 2023. Date of publication 5 May 2023; date of current version 30 August 2023. This work was supported by Toyota Motor North America InfoTech Labs. The contents of this paper reflect the views of the authors, who are responsible for the facts and the accuracy of the data presented herein. The contents do not necessarily reflect the official views of Toyota Motor North America. The Associate Editor for this article was C. Wen. *(Corresponding author: Zhengwei Bai.)*

Zhengwei Bai, Saswat P. Nayak, Xuanpeng Zhao, Guoyuan Wu, and Matthew J. Barth are with the Department of Electrical and Computer Engineering, University of California at Riverside, Riverside, CA 92507 USA (e-mail: zbai012@ucr.edu).

Xuewei Qi, Yongkang Liu, Emrah Akin Sisbot, and Kentaro Oguchi are with Toyota Motor North America, Research and Development Labs, InfoTech Labs, Mountain View, CA 94043 USA.

Digital Object Identifier 10.1109/TITS.2023.3268281

is facing a couple of major challenges: 1) the cost of large-scale real-world implementation is prohibitive; and 2) the detection ranges are limited for onboard sensors, which also suffer from occlusion partially due to mounting heights and positions [7].

Recently, roadside sensor-assisted perception is attracting a significant amount of attention for CAVs and is regarded as a promising way to unlock numerous opportunities for cooperative driving automation applications [8]. Current roadside sensing systems are mainly camera-based, which are cost-effective and well-developed for traffic surveillance such as turning movement counts, but hard to provide reliable object-level high-fidelity 3D information due to lighting conditions and shadow effects [9].

Considering its capability to determine an accurate 3D location based on point cloud data, LiDAR gets more popular in infrastructure-based traffic surveillance. Previous studies validated the performance of roadside LiDAR for vehicle detection, vehicle tracking, lane identification, pedestrian near-crash warning, and other applications [10]. These studies laid the foundation for applications with roadside LiDAR-based perception systems. However, most of these systems are deployed upon traditional perception pipelines [11], consisting of background filtering, point cloud clustering, object classification, and object tracking. Such pipelines may generate stable results but suffer from uncertainties and generality [12]. With the development of computer vision, deep learning-based perception models show great potential to overcome the above issues. However, few studies applied deep learning-based perception algorithms to roadside LiDAR systems.

The main contributions of this paper can be summarized as follows:

1) To the best of the authors' knowledge, this paper is the first attempt to comprehensively build a deep learning-based real-world platform, called Cyber Mobility Mirror (CMM), for 3D object-level cooperative perception at a signalized intersection using the roadside LiDAR.
2) A mobile-edge-cloud (MEC) framework is designed and implemented for real-world prototyping, with the consideration of scalability.
3) An onboard system is designed and developed for real-time object reconstruction and display.
4) An automatic perception evaluation approach is proposed for model assessment without the involvement of human-labeling efforts.

The CMM platform can serve as the stepping stone to enabling various cooperative driving automation (CDA) applications.

The rest of this paper is organized as follows: related work is firstly introduced in Section II. Section III shows the concept and structure of CMM, followed by a detailed description of the associated field operational system in Section IV. The results and analyses are discussed in Section V and the last section concludes this paper with further discussion.

II. Background

Situation awareness is one of the fundamental building blocks for Driving Automation. Specifically, 3D object detection and tracking play a crucial role in perceiving the environment. Meanwhile, traffic object reconstruction helps drivers better understand traffic conditions. Hence, in this section, related work about the detection, tracking, and reconstruction of traffic objects is presented.

A. Traffic Object Detection

Object detection is a fundamental task of environment perception and has also gone through a rapid development process in the past several decades. Back twenty years ago, a vision-based traffic detection system made an impressive achievement using statistical methods [13]. Aslani and Mahdavi-Nasab [14] proposed an optical flow-based moving object detection method for traffic surveillance. However, these model-based methods cannot provide high-fidelity detection results for more delicate applications, e.g., precise localization and object-level tracking.

With the tremendous progress of convolutional neural networks (CNNs) in vision-based tasks, CNN-based object detection methods have attracted a significant amount of attention in traffic surveillance [15]. *You Only Look Once* (*YOLO*) [16] and its variants, due to an impressive performance in real-time multi-object detection, get very popular in high-resolution traffic monitoring scenarios. *Faster RCNN* [17] is another generic epoch-making detection method, utilizing the region proposal ideology. To further improve the object detection performance for Faster-RCNN, Li et al. [18] proposed a cross-layer fusion structure based on Faster RCNN to achieve a nearly 10% higher average accuracy in complex traffic environments.

Except for the general object detection task applied in traffic scenes, many studies focus on specific perception cases. For instance, considering that existing traffic surveillance systems were made up of costly equipment with complicated operational procedures, Mhalla et al. [19] designed an embedded computer-vision system for multi-object detection in traffic surveillance. For small object detection, Lian et al. [20] proposed an attention feature fusion block to better integrate contextual information from different layers that could achieve much better performance.

To support object-level cooperative operations, detecting the objects in a 3D format is a straightforward and promising way to high-fidelity situation awareness. Hence, owing to the capability of generating 3D point clouds with spatial information, it is increasingly popular for deploying 3D LiDAR to traffic environment perception. Wu et al. [21], proposed a revised *Density-Based Spatial Clustering of Applications with Noise* (3D-DBSCAN) method to detect vehicles based on roadside LiDAR sensors under rainy and snowy conditions. Using a roadside LiDAR, Zhang et al. proposed a three-stage inference pipeline, called *GC-net* [22], including the gridding, clustering, and classification. To distinguish the moving object from the point cloud, Song et al. proposed a hierarchical searching method based on the feature distribution of point clouds to achieve background filtering and object detection [11]. Although 3D LiDAR has innate advantages to dealing with 3D object detection, the lack of labeled roadside datasets significantly limits the potential for applying deep learning-based detectors to roadside LiDAR sensors. Hence in this paper, an point cloud encoder-decoder method is proposed to enable the detection model to work on the roadside point clouds with training on the onboard dataset.

B. Traffic Object Tracking

Deploying CDA in urban environments poses a series of difficult technological challenges, out of which object tracking is arguably one of the most significant since it provides the identification information for other subsequent technical models [23]. Object tracking can be classified into two categories in terms of the number of objects tracked at one time: one is single-object tracking (SOT) and the other is multi-object tracking (MOT). SOT has been investigated over several decades and the *Kalman filtering*-based methods have been developed widely [24] for this type of task. For MOT tasks, some approaches have been proposed with the focus on improving accuracy and real-time performance. Bewley et al. [25] proposed *Simple Online and Real-time Tracking* (SORT) that can achieve MOT in a high frame rate without much-compromising accuracy. Based on SORT, Bewley and Paulus [26] proposed a multi-object tracker – *DeepSORT*, which was capable of tracking objects with longer periods of occlusions and effectively reducing the number of identity switches by integrating the appearance features. However, DeepSORT does not apply to 3D objects.

Chen et al. proposed a camera-based edge traffic flow monitoring scheme using DeepSORT [27]. Recent advances in LiDAR technology enable it to hold a place in traffic object tracking tasks, by leveraging the point cloud data. For instance, Cui et al. [28] provided a simple *global nearest neighbor* (GNN) method to track multiple vehicles based on the spatial distance between consecutive frames. *Adaptive probabilistic filtering* was utilized by Kampker et al. [29] to handle uncertainties due to sensing limitations of 3D LiDARs and the complexity of targets' movements.

C. Traffic Object Reconstruction

Traffic reconstruction, traditionally, means rebuilding the traffic scenarios or parameters based on recorded sensor data, such as loop detectors and surveillance cameras [30], [31]. These traffic-level reconstruction data are valuable for macroscopic traffic management. In this paper, nevertheless, the object-level reconstruction means rebuilding the 3D location or shape of certain objects based on sensor data, which can more concrete information to support subsequent CDA applications. Several studies have been conducted in this emerging area. Cao et al. [32] developed a camera-based 3D object reconstruction method on the Internet of Vehicles (IoV) environment. Rao and Chakraborty [33] proposed a LiDAR-based monocular 3D shaping to reconstruct the surrounding objects for onboard display, which has a similar purpose to the reconstruction work in this paper.

III. Cyber Mobility Mirror (CMM)

To explore the potential of the roadside sensing system, we propose a novel infrastructure-based object-level perception system, named *Cyber Mobility Mirror*. In this section, the core concept of CMM and the associated platform implemented in the real world are introduced.

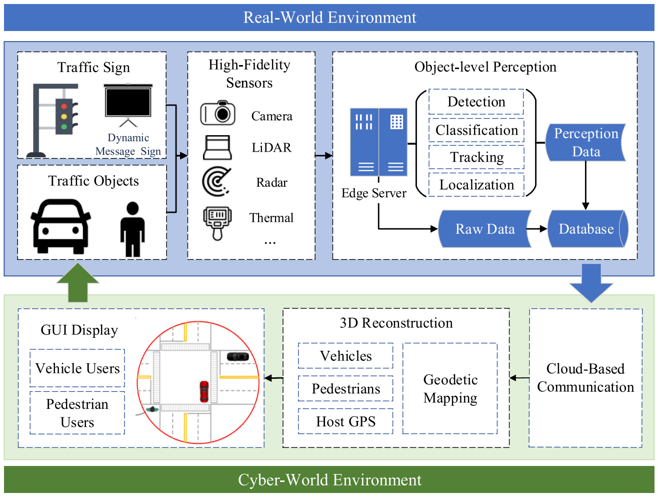


Fig. 2. Systematic diagram for the core concept of CMM.

A. Core Concept of CMM

CMM aims to enable real-time object-level traffic perception and reconstruction to empower various cooperative driving automation (CDA) applications, such as *Collision Warning* [34], *Eco-Approach, and Departure* (EAD) [35], and *Cooperative Adaptive Cruise Control* (CACC) [36]. In the CMM system, traffic conditions (i.e., "*mobility*") are detected by high-fidelity sensors and advanced perception methods, such as object detection, classification, and tracking. In the "*cyber*" world, digital replicas (i.e., "*mirrored*" objects) are built to reconstruct the traffic in *real-time* via high-definition 3D perception information, such as the detected objects' geodetic locations (rendered on the satellite map), 3D dimensions, speeds, and moving directions (or headings). Then, this "*mirror*" can act as the perception foundation for numerous CDA applications in a real-world transportation system.

Specifically, Fig. 2 illustrates the system diagram for the core concept of CMM. Traffic objects can be detected by high-fidelity sensors equipped on the infrastructure side and the sensing data is processed by an edge server to generate object-level information and enable various functions, such as detection, classification, tracking, and geodetic localization. The perception information is also transmitted to a cloud server for distribution and 3D reconstruction. The reconstructed traffic environment can be displayed on the GUI of connected road users to support various CDA applications.

B. Systematic Structure of CMM

In the real-world traffic environment, the system architecture of the CMM system is designed by following the core concept. Specifically, the CMM system can be divided into two main parts: the CMM RoadSide System (CMM-RSS) and the CMM Onboard System (CMM-OBS).

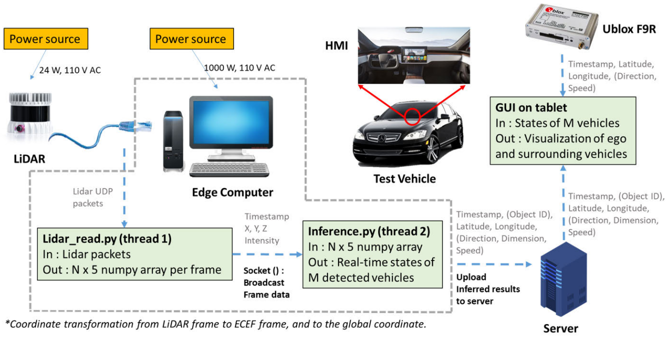

Power source
24 W, 110 V AC
Power source
1000 W, 110 V AC
LiDAR
Edge Computer
HMI
Test Vehicle
Ublox F9R
Timestamp, Latitude, Longitude, (Direction, Speed)
GUI on tablet
In : States of M vehicles
Out : Visualization of ego and surrounding vehicles
Lidar UDP packets
Lidar_read.py (thread 1)
In : Lidar packets
Out : N x 5 numpy array per frame
Timestamp X, Y, Z Intensity
Socket () : Broadcast Frame data
Inference.py (thread 2)
In : N x 5 numpy array
Out : Real-time states of M detected vehicles
Timestamp, (Object ID), Latitude, Longitude, (Direction, Dimension, Speed)
Upload Inferred results to server
Timestamp, (Object ID), Latitude, Longitude, (Direction, Dimension, Speed)
Server
*Coordinate transformation from LiDAR frame to ECEF frame, and to the global coordinate.

Fig. 3. The architecture for CMM field operational prototype system.

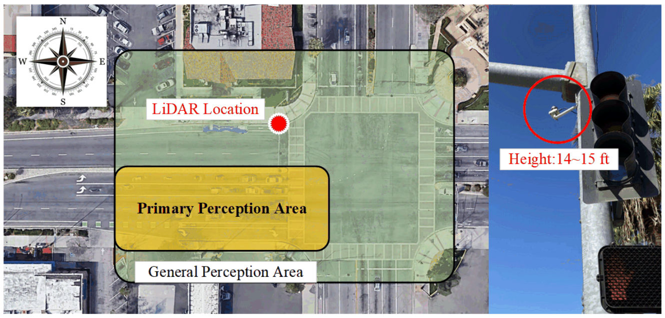


Fig. 4. LiDAR installation and tentative coverage area.

1) CMM Roadside System: CMM-RSS consists of 1) roadside sensors, e.g., LiDAR in this study, to perceive traffic conditions and generate high-fidelity sensor data; 2) edge computing-based real-time perception pipeline to achieve sensor fusion (if appropriate), object detection, classification, and tracking tasks; and 3) communication devices to receive information from other road users, infrastructure or even "clouds", and share perception results with them via different kinds of protocols (the communication protocols used in this paper is introduced in Section IV-G).

2) CMM Onboard System: For CAVs, CMM-OBS can receive the object-level perception data from CMM-RSS and then act as the perception inputs to support various CDA applications, such as CACC, cooperative merging, cooperative eco-driving; and for Connected Human-driven Vehicles (CHVs), CMM-OBS can also provide them with real-time traffic information via the human-machine interface (HMI) to improve driving performance or to avoid possible crashes due to occlusion.

In this paper, the CMM concept is implemented in the real world and a field operational system is developed for real-world testing, which will be discussed in Section IV.

IV. CMM Field Operational System

A. System Overview

The system overview for the CMM Field Operational System (FOS) is shown in Fig. 3. The FOS mainly consists of a roadside 3D LiDAR for data collection, an edge-computing system for data processing, a cloud server for data distribution, and a test vehicle equipped with connectivity and Graphic User Interface (GUI). To be specific, the LiDAR is installed on the signal pole high enough to achieve better coverage. The edge computer retrieves 3D point cloud data from the roadside LiDAR and then generates high-definition perception information (i.e., 3D object detection, classification, and tracking results) which is transmitted to the cloud server via Cellular Network. A CHV equipped with the CMM OBUs (including a GPS receiver, onboard communication device, and a tablet) can receive the perception information, and reconstruct and display the object-level traffic condition on GUI in real-time.

The whole system follows an edge-cloud structure where the edge server and cloud are mainly responsible for raw data processing and message distribution respectively. It is notable that computing on the cloud and communicating on edge (e.g., DSRC) is also a theoretically feasible structure. However, for the Lidar-based CMM system, the raw Lidar data transmission requires a 50+ MBps data transmitting rate with low latency which is hard to be satisfied by the cloud, and the edge-based communication is limited by communicating range and occlusion.

B. System Initialization

As demonstrated by Fig. 4, the LiDAR is installed at the northwest corner of the intersection (marked as the red circle) of University Ave. and Iowa Ave. in Riverside, California. In this work, an OUSTER®64-Channel 3D LiDAR is used as a major roadside sensor, mounted on a signal pole at the height of 14-15 ft above the ground with the appropriate pitch and yaw angles to cover the monitoring area enclosed by the orange rectangle in Fig. 4. The edge computer at the intersection retrieves the data stream from the LiDAR in the form of UDP packets. Point cloud attributes such as 3D location, i.e., x, y, z, and the intensity, i, of each point are bundled into an $N \times 4$ array to be processed for generating 3D detection, tracking, and localization results.

Additionally, two types of perception areas are defined in Fig. 4. Since University Ave. (horizontal) is the main focus of the previous Riverside Innovation Corridor project [37], a primary perception area is identified as the yellow box in Fig. 4. For the rest of the area that is perceived from general purpose, we define it as the general perception area shown in the green box in Fig. 4.

C. Data Retrieving and Preprocessing

The raw point cloud data is generated by a 64-channel 3D LiDAR and then the edge computer retrieves the raw data through an Ethernet cable via UDP communication. In this paper, the detection range Ω for the roadside LiDAR is defined as a $102.4m \times 102.4m$ area centered on the location of LiDAR. The raw point cloud data can be described by:

$$\mathcal{P} = \{[x, y, z, i] \mid [x, y, z] \in \mathbb{R}^3, i \in [0.0, 1.0]\}. \quad (1)$$

Then, $\mathcal{P}$ is geo-fenced by:

$$\mathcal{P}_\Omega = \{[x, y, z, i]^T \mid x \in \mathcal{X}, y \in \mathcal{Y}, z \in \mathcal{Z}\} \quad (2)$$

where $\mathcal{P}_\Omega$ represents the 3D point cloud data after geofencing.

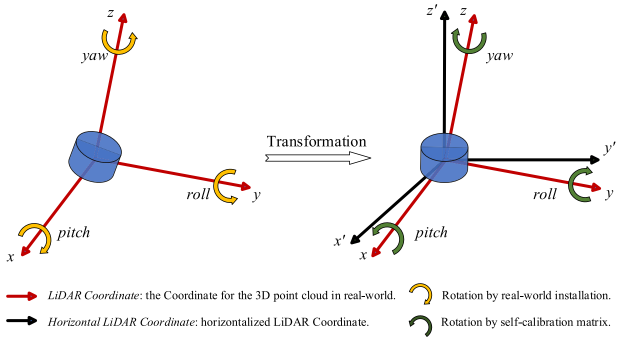

z
yaw
roll
y
pitch
x
Transformation
z'
z
yaw
y'
roll
y
x'
x
pitch
LiDAR Coordinate: the Coordinate for the 3D point cloud in real-world.
Horizontal LiDAR Coordinate: horizontalized LiDAR Coordinate.
Rotation by real-world installation.
Rotation by self-calibration matrix.

Fig. 5. Description of the initial transformation for LiDAR point cloud data.

D. 3D Object Detection From Roadside LiDAR

1) Roadside Data Transformation: Considering the LiDAR's limited vertical field of view (FOV), it is installed with an adjusted rotation angle including pitch, yaw, and roll to cover the desired surveillance area as shown in Fig. 5. To build the system cost-effectively, we try to use an open-source dataset to train our detection model, e.g., Nuscenes [38]. However, these available datasets are collected based on a vehicle-equipped LiDAR. These LiDAR sensors have different spatial configurations from ours. The domain gap between Nuscenes and our roadside data will lead to performance degradation if the model trained on these datasets is applied to our roadside point clouds directly.

Therefore, to empower the model with the capability of domain adaptation – training on onboard datasets while inference on the roadside data – we propose the Roadside Data Transformation (RDT). The main purpose of RDT is to transform roadside point clouds into a space in which the model trained on the onboard datasets can work out. The transformation process of the RDT is described in Fig. 5.

To achieve the transformation, we propose a self-calibration approach for the roadside-LiDAR pose by using Least Square Regression (LSR) to the point clouds. The coordinate for roadside point clouds is defined as *LiDAR Coordinate (L-Coor)* and the coordinate of point clouds after encoding, is defined as *Horizontal Coordinate (H-Coor)*. Using LSR, the least square plane is generated to represent the $x-y$ plane of the *L-Coor*. Then the 3D rotation matrix can be generated as $\mathcal{P}_{Cali}$, which is shown as:

$$\mathcal{P}_{Cali} = \begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \tag{3}$$

where $a, \ldots, i$ are the parameters generated from LSR. For translation, the vertical offset Δz is defined as:

$$\Delta z = z_{roadside} - z_{onboard} \tag{4}$$

where $z_{roadside}$ and $z_{onboard}$ represent the heights of the roadside LiDAR and the onboard LiDAR (used in the training dataset), respectively.

The whole encoding process is defined by:

$$\mathcal{P}_{\mathcal{H}} = \mathcal{P}_{\Omega} \cdot \begin{bmatrix} \mathcal{P}_{Cali} & 0 \\ 0 & 1 \end{bmatrix} + [0, 0, \Delta z, 0] \tag{5}$$

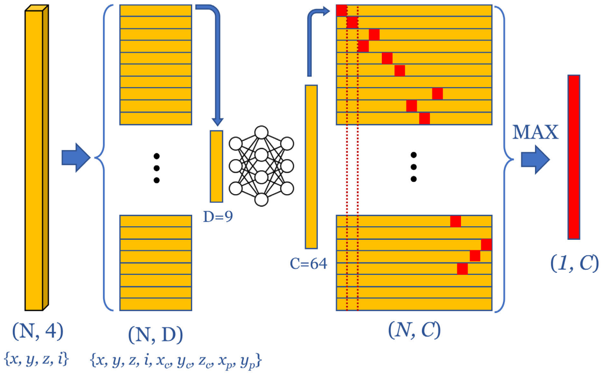


Fig. 6. Process for the feature extraction and compression.

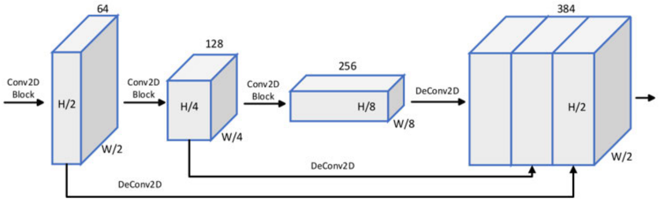


Fig. 7. Deep neural network backbone for hidden feature extraction.

2) Object Detection Network: Although the roadside point cloud is transformed into the coordinate suitable for training on the onboard dataset. The detection model has still required a large tolerance for the difference in data. Since there is a large shifting, i.e., near $3m$, along $z-$axis, to make the model not too sensitive for $z-$axis data, we voxelized the point cloud following the strategy applied in [39], i.e., only voxelization on the $x-y$ plane to generate point cloud pillars. Then data aggregation, as shown in Fig. 6, is designed to extract and compress the features which will be sent to the deep neural network for generating predicted bounding boxes.

After the data aggregation, Fig 7 shows the designed feature pyramid network (FPN) followed by a 3D anchor-based detection head [40] to generate predicted bounding boxes. The FPN consists of two sub-networks: 1) one 2D convolutional (Conv2D) layer-based network that generates the extracted features with decreasingly spatial resolution; and 2) one deconvolutional (DeConv2D) layer-based network that generates output features by performing upsampling and concatenation. Each Conv2D block consists of one Conv2D layer with the kernel of (3, 2, 1), followed by several Conv2D layers with kernels of (3, 1, 1). Specifically, the numbers of Conv2D layers in each block are 4, 6, and 6, respectively.

For the loss functions, localization and classification are considered. To be specific, ground targets (GT) and anchors are defined by an 8-dimensional vector $(x, y, z, w, l, h, \theta)$. The localization regression residuals between ground truth and anchors are defined by:

$$\Delta x = \frac{x^{gt} - x^{a}}{d^{a}}, \quad \Delta y = \frac{y^{gt} - y^{a}}{d^{a}}, \quad \Delta z = \frac{z^{gt} - z^{a}}{h^{a}}, \tag{6}$$

$$\Delta w = \log \frac{w^{gt}}{w^{a}}, \quad \Delta l = \log \frac{l^{gt}}{l^{a}}, \quad \Delta h = \log \frac{h^{gt}}{h^{a}}, \tag{7}$$

$$\Delta \theta = sin(\theta^{gt} - \theta^{a}) \tag{8}$$

where the superscript gt and a represent the ground truth and anchor, respectively, and d^a is defined by:

$$d^a = \sqrt{(w^a)^2 + (l^a)^2}. \tag{9}$$

The total localization loss is:

$$\mathcal{L}_{loc} = \sum_{b \in (x,y,z,w,l,h,\theta)} \text{SmoothL1}(\Delta b) \tag{10}$$

Inspired by [41], a softmax classification loss, $\mathcal{L}_{dir}$, is used to distinguish flipped boxes. The object classification is enabled by the focal loss [42], which is shown as:

$$\mathcal{L}_{cls} = -\alpha_a (1 - p^a)^\gamma \log p^a, \tag{11}$$

where p^a is the class probability of an anchor, and α and β are set as the same as the original paper. Hence, the total loss is:

$$\mathcal{L} = \frac{1}{N_{pos}} (\beta_{loc}\mathcal{L}_{loc} + \beta_{cls}\mathcal{L}_{cls} + \beta_{dir}\mathcal{L}_{dir}), \tag{12}$$

where N_{pos} is the number of positive anchors and β_{loc}, β_{cls} and β_{dir} are set as 2, 1, and 0.2.

E. 3D Multi-Object Tracking

For real-time 3D MOT, we propose *3DSORT* by adding 3D object matching on DeepSORT [27]. To be specific, 2D location information is filtered from the 3D detection results, and the 2D location data is fed into the DeepSORT model to generate the 2D MOT results, i.e., unique identification (ID) number for each object. Then, a Euclidean distance-based 3D object-matching algorithm is designed to generate enhanced 3D MOT results.

Algorithm 1 The Description for 3DSORT

```
Input: The instant 3D object detection results: $Dobj = \{D^{(i)}(x, y, z, w, l, h, \theta)|i = 1, 2, \dots, N_{Dbbx}\}$;
Output: The multi-object tracking results: $Tobj = \{T^{(i)}(x, y, z, w, l, h, \theta, id)|i = 1, 2, \dots, N_{Dbbx}\}$;
function 3D DEEPSORT($Dobj$)
    $Dobj_{2d} \leftarrow D^{(i)}(x, y, w, l)|i = 1, 2, \dots, N_{Dbbx}$;
    $Tobj_{2d} = \{T_{2d}^{(j)}(x, y, w, l, id)|j = 1, 2, \dots, N_{Tbbx}\} \leftarrow DeepSORT(Dobj_{2d})$;
    for $Dobj_{2d}^{(i)} \in Dobj_{2d}$ do
        for $Tobj_{2d}^{(j)} \in Tobj_{2d}$ do
            if Euclidean distance of $(Dobj_{2d}^{(i)}, Tobj_{2d}^{(j)}) < d_o$ then
                $T^i \leftarrow [D^{(i)}, Tobj_{2d}(id)]$; Continue;
            end if
        end for
    end for
    $Tobj = \{T^{(i)}|i = 1, 2, \dots, N_{Dbbx}\}$
    return $Tobj$;
end function
```

Algorithm 1 demonstrates the details of 3DSORT where N_{Dbbx} and N_{Tbbx} are the numbers of the detection bounding boxes and 2D tracking boxes, respectively. Additionally, id represents the tracking identification number for each unique object. d_o is the matching distance which is defined as $0.2m$.

F. Geo-Localization

To endow the perception data with more generality, the geo-referencing of the point cloud is developed in this work. However, the output T_{boxes} from the 3D MOT is calculated based on the Horizontal-LiDAR Coordinate, i.e., a Cartesian Coordinate centered with the sensor installed evenly. Thus, the input of the geo-localization data, i.e., the T_{boxes} from Algorithm 1, is then fed into a multi-step transformation process to transform the object location information to Geodetic Coordinate, i.e., latitude, longitude, and altitude. There are three steps: 1) from the horizontal-LiDAR coordinate to the real LiDAR coordinate; 2) from the real LiDAR coordinate to the Geocentric Earth-centered Earth-fixed (ECEF) coordinate; and 3) from ECEF coordinate to the geodetic coordinate (i.e., latitude, longitude, and altitude). Specifically, the World Geodetic System 1984 (WGS84) is applied for the geo-transformation. The transformation from the Horizontal LiDAR coordinate to the ECEF coordinate system is shown in Eq. 13.

$$\begin{bmatrix} X_{ecef} \\ Y_{ecef} \\ Z_{ecef} \\ 1 \end{bmatrix}^T = \begin{bmatrix} X_{hor} \\ Y_{hor} \\ Z_{hor} \\ 1 \end{bmatrix}^T \cdot \mathcal{P}_{Cali}^{-1} \cdot \mathcal{P}_{ECEF} \tag{13}$$

where $\mathcal{P}_{Cali}^{-1} \in \mathcal{R}^{4\times4}$ and $\mathcal{P}_{ECEF} \in \mathcal{R}^{4\times4}$ are the inverse of the LiDAR calibration matrix, and the ECEF transformation matrix, respectively. X_{hor}, Y_{hor}, and Z_{hor} represent the coordinates of 3D points concerning Horizontal LiDAR Coordinate. The $\mathcal{P}_{ECEF}$ matrix responsible for transforming points in LiDAR coordinate frame to the geocentric coordinate frame (ECEF) is calculated using the Ground Control Point surveying technique [43].

The longitude (λ) is calculated from the ECEF position using Eq. 14,

$$\lambda = \arctan(\frac{Y_{ecef}}{X_{ecef}}) \tag{14}$$

The geodetic latitude (ϕ) is calculated using Bowring's method by solving Eq. 15 and Eq. 16 in an iterative manner,

$$\overline{\beta} = \arctan(\frac{Z_{ecef}}{(1-f)s}) \tag{15}$$

$$\overline{\phi} = \arctan(\frac{Z_{ecef} + e^2(\frac{1-f}{1-e^2})R(\sin\beta)^3}{s - e^2R(\cos\beta)^3}) \tag{16}$$

where R, f, and $e^2 = 1-(1-f)^2$ are the equatorial radius, flattening of the planet, and the square of first eccentricity, respectively. s is defined as $s = \sqrt{X_{ecef}^2 + Y_{ecef}^2}$. The altitude ($h_{ego}$, height above ellipsoid) is given by,

$$h_{ego} = s\cos\phi + (Z_{ecef} + e^2N\sin\phi)\sin\phi - N \tag{17}$$

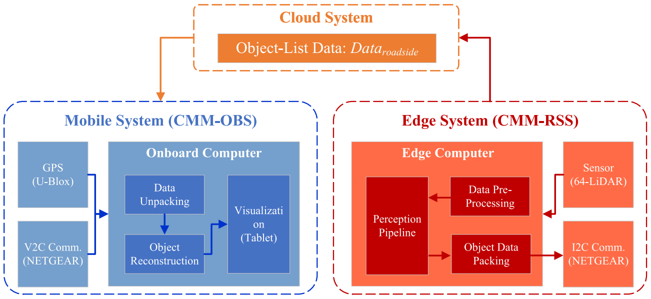


Fig. 8. Illustration of the Mobile-Edge-Cloud communication structure.

where N, the radius of curvature in the vertical prime, is defined as

$$N = \frac{R}{\sqrt{1 - e^2(\sin\phi)^2}} \tag{18}$$

Then the geo-referenced perception information (ϕ, λ, h_{ego}) along with other data will be transmitted to the cloud server for distribution and the final data is packaged as:

$$Data_{roadside} = \{M^{(i)}(t, id, \phi, \lambda, h_{ego}, w, l, h, \theta)\}_{i=1}^{N_{Dbbx}} \tag{19}$$

G. Mobile-Edge-Cloud Communication

To make the CMM capable of future extension, a Mobile-Edge-Cloud communication framework is designed, as shown in Fig. 8. The cloud system is applied to cope with a dynamic number of edge systems or mobile systems (e.g., data synchronization algorithms can be deployed in the cloud system to align all the timestamps of different subsystems.).

Fig. 8 illustrates the inter-system and intra-system communications. Specifically, the CMM-OBS retrieves traffic perception data from the cloud server and GPS location data from a GPS receiver. Then the onboard unit reconstructs the traffic conditions based on the multi-source data and displays it on the graphical user interface (GUI) in real time. In our field implementation, a Samsung Galaxy Tab A7 tablet serves as the onboard computer, running a designed application to retrieve data from the GPS receiver and displaying the reconstructed object-level traffic information on the GUI. To have accurate GPS measurements, we utilize a C102-F9R U-Blox unit with an embedded Inertial Measurement Unit (IMU) which provides an 8Hz update frequency on the GPS location and heading.

For Vehicle-to-Cloud (V2C) and Infrastructure-to-Cloud (I2C) communication, We applied the NETGEAR AirCard 770S mobile hotspots which are equipped with 4G/LTE sim cards and can provide V2C/I2C communication between the cloud server and CMM-OBS/CMM-RSS.

H. Multi-Object Reconstruction

An application is designed to visualize the location of vehicles perceived by the roadside unit (RSU) and the ego vehicle provided by the OBU. To achieve that, we first locate the monitored area at the intersection and crop it from the Google Earth Pro satellite view. We leverage the cropped image as a background map for visualizing the reconstructed traffic. Firstly, we calculate the distance between two reference GPS points using the Haversine formula as shown followed.

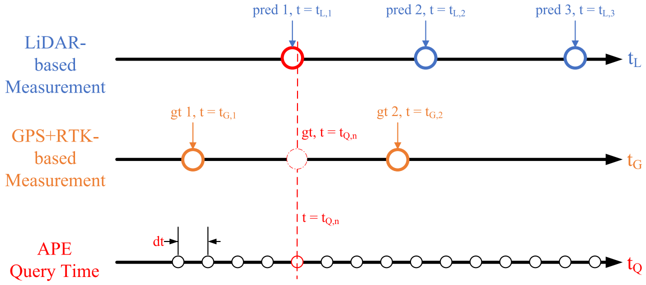


Fig. 9. The diagram illustrating the APE approach.

$$\begin{aligned} a &= sin^2(\Delta lat/2) + cos(lat_{ref1}) \\ &\quad \cdot cos(lat_{ref2}) \cdot sin^2(\Delta lon/2) \\ c &= 2 \cdot atan2(\sqrt{a}, \sqrt{1-a}) \\ d &= R \cdot c \end{aligned} \tag{20}$$

where lat_{ref1} and lat_{ref2} are latitudes of two reference GPS points, Δlat is the latitude difference between two GPS points, Δlon is the longitude difference between two GPS points, R is the radius of the earth, and d is the distance computed between two GPS points. Based on the number of pixels between their displayed pixel coordinates on the tablet, we can calculate the transfer ratio between them.

$$\frac{Pix_{ref1} - Pix_{ref2}}{Dis_{ref1} - Dis_{ref2}} = \alpha \tag{21}$$

where, Pix_{ref1} and Pix_{ref2} are the pixel coordinates of two reference points, Dis_{ref1} and Dis_{ref2} are the distance between two reference points, and α is the transfer ratio. By now, we can create an object and display it on the desired pixel coordinates based on its GPS location.

I. Automatic Perception Evaluation

In this section, we propose a novel automatic perception evaluation (APE) approach that doesn't require human-labeled data for evaluating the detection, tracking, and localization performance of data-driven models. Specifically, rather than evaluating all objects in each frame, which requires costly human-labeling effort, evaluating the perception performance of the ego-vehicle (GPS+RTK enabled) can be statistically regarded as a sampling evaluation process. The APE approach can be illustrated in Fig. 9.

Since the GPS data and LiDAR data are asynchronous data, to associate each perception frame with an accurate ground truth, time interpolation and motion model are adopted in our APE approach as shown in Fig. 9. Considering the future extension of more sensors, a query time t_Q is designed to increasingly traverse all the LiDAR-based measurements $Dobj$. To cope with the non-synchronization issue, two adjacent GPS measurements $G^{ego}(t_{G,1}, t_{G,2})$, before and after the LiDAR-based measurement, are extracted and a linear motion model is applied to estimate the associated ground truth ego_{gt}

at time t_L. The detailed description of the APE approach is shown in Algorithm 2, where δt is set as 0.02 s and t_{end} is set as the final time of GPS-based measurement.

Algorithm 2 The Description for APE Approach

```
Input: The LiDAR-based objects measurement: $Dobj = \{D^{(i)}(t_L, x_{pred}, y_{pred}, z_{pred}, id_{pred}) | i = 1, 2, \ldots, N_{Dbbx}\}$; The GPS-based ego-vehicle measurement: $Gobj = \{G^{ego}(t_G, x_{gt}, y_{gt}, z_{gt}, id_{gt})\}$;
Output: The APE information matrix: $\mathcal{S} = \{D^{ego}(t_L, x_{pred}, y_{pred}, z_{pred}, id_{pred}, x_{gt}, y_{gt}, z_{gt}, id_{gt})\}$;
function APE($Dobj$, $Gobj$)
    initialize idx_lidar_buffer, $\mathcal{S}$, $t_{end}$;
    while $t \leq t_{end}$ do
        idx_lidar ← index of the most recent LiDAR frame
        if idx_lidar $\neq$ idx_lidar_buffer then
            $Dobj(t)$ ← retrieve objects at this frame
            $t_L \leftarrow Dobj(t)$
            Two adjacent GPS measurement before and after the $t_L$: $t_{G,1}, t_{G,2} \leftarrow G^{ego}$
            Estimating the velocity at the query time: $v_{t_Q} \leftarrow G^{ego}(t_{G,1}, t_{G,2})$
            Estimating the ground truth $ego_{gt}$ at the query time: $ego_{gt} = \{x_{gt}, y_{gt}, z_{gt}, idgt\} \leftarrow v_{t_Q}, G^{ego}(t_{G,1}, t_{G,2}), Dobj(t)$
            Find the ego-vehicle's measurement $ego_{pred}$ from euclidean distance to the $bbox_{gt}$
            Append $\{t_L, ego_{pred}, ego_{gt}\}$ to $\mathcal{S}$
            idx_lidar_buffer ← idx_lidar
        end if
        $t \leftarrow t + \delta t$
    end while
    return $\mathcal{S}$;
end function
```

V. Field Testing and Results Analysis

A. Feasibility

Object-level perception information acts as the building block for CMM, which requires high-fidelity data retrieved from high-resolution sensors, such as LiDARs. Nevertheless, it could be costly, time-consuming, and to some extent, restricted by policies and protocols, to deploy these sensors directly in the real world. Thus, it is necessary to evaluate the feasibility of the system at the early stage of this work.

To find an efficient and cost-effective way to validate the feasibility of CMM, we emulated a CMM system in a simulation platform, i.e., a CARLA-based co-simulation system [44], before the real-world implementation. As demonstrated in Fig. 10, the basic idea is to emulate the real-world traffic environment via one CARLA simulator [45] and run the entire perception process within the emulated real-world environment. Then the other CARLA simulator is applied to emulate the cyber world, i.e., to reconstruct the traffic objects and then display them. Owing to the capability of CARLA to model high-fidelity sensors, the evaluation results of the emulated CMM in the co-simulation platform can lay the foundation for real-world CMM implementation.

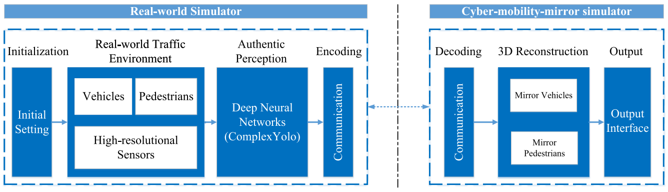


Fig. 10. Structure for the CMM-based co-simulation platform.

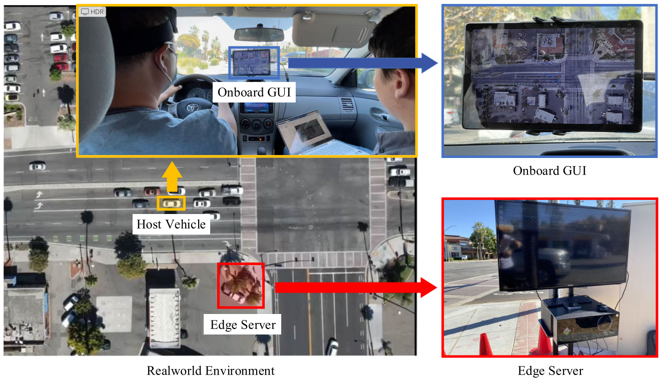


Fig. 11. Illustration of CMM field operational test from different views from a drone, host vehicle, onboard GUI, and edge server.

After the feasibility check in the simulation environment, we implement the CMM field operational system (FOS) at a real-world intersection of University Ave. & Iowa Ave. in Riverside, California. Fig. 11 depicts the field system from different views. Multi-view videos are captured along the test including drone's view, in-vehicle views (including driver perspective, backseat passenger perspective, and GUI), roadside view, and point cloud data-based bird's-eye view (BEV). A video clip is edited with descriptive annotations to show the whole online process, which is available at https://www.youtube.com/watch?v=0egpmgkzyG0). The video demonstrates the feasibility of the CMM FOS and the following sections will show the results of detection accuracy and real-time performance.

B. Experimental Setup

1) General Setting: For the purpose of evaluating and investigating the real-time perception performance of our CMM system, the perceiving area is set as the intersection of University Ave. & Iowa Ave., Riverside, CA, USA as shown in Figure 4. Specifically, due to the effective ground truth range in our training dataset (i.e., ±51.2m for the nuScenes dataset), the geo-fencing area of the point cloud data is set as the range of $x \in [-51.2m, 51.2m]$ and $y \in [-51.2m, 51.2m]$. For the object detection network, the spatial size of the voxel is set as $[0.25m, 0.25m, 8.0m]$ and the maximum number of voxels is set to 30, 000 and 40, 000 during training and testing, respectively.

For the score threshold during the inference pipeline setting, two factors are considered: 1) the objects at the most right

TABLE I

DATASET ANALYSIS FOR MODEL TRANSFERABILITY

Dataset	LiDAR Structure	Horizontal FOV	Vertical FOV	Channel	Channel Under Horizon	Inter-beam Angle
KITTI	Single	180°	$[-24.9°, +2°]$	64	59	0.420°
NuScenes	Single	360°	$[-30.0°, +10°]$	32	24	1.250°
Waymo	Multiple (1 mid-range, 4 short range)	360°	$[-17.6°, +2.4°]/[-90°, 30°]$	64	56.32/48	0.313°/1.875°
CMM (Ours)	Single	360°	$[-22.5°, 22.5°]$	64	32	0.703°

lane of University Ave. (the southwest horizontal lane in Fig. 4) should be able to be detected (because the approaching vehicles from the west direction will be further used for future research), and 2) the objects within the intersection range shouldn't have too many False Positive (FP) detections. Based on the principles above, we finally set the score threshold to 0.3, which can recall most of the vehicles in our tentative coverage area with satisfied precision performance.

2) Dataset Selection: To decide the Dataset used for training our model, we further conducted several experiments to analyze the model transferability by utilizing different training datasets, e.g., KITTI [46], NuScenes [38], or Waymo Open Dataset [47]. Specifically, we trained the detection model on KITTI, NuScenes, and Waymo Open Dataset and applied the same domain adaptation techniques that we proposed in Section IV-D.1. However, we found that the model trained on KITTI and Waymo Open Dataset can barely return correct detection results. By analyzing the detailed specification of these Datasets in Table I, we found that KITTI and Waymo Open Dataset have bigger domain gaps than the NuScenes Dataset if compared with our data.

In Table I, from the perspective of LiDAR structure, we can find that PCD from Waymo Open Dataset is generated from the multi-LiDAR system while PCD from KITTI/NuScenes is generated from a single-LiDAR system which is closer to ours. From the perspective of FOV, KITTI only has a 180° horizontal FOV which is not enough for our scenario in which a panoramic horizontal view is required. Additionally, Way e from ours in terms of the vertical FOV. For the resolution of the LiDAR, although Nuscenes has only 32 channels of the laser beam, it is the only one whose inter-beam angle is larger than ours. In other words, the PCD collected by our LiDAR only satisfies the resolution standard of the NuScenes dataset, which leads to better transferability. For instance, the data from KITTI has more beams under the horizon and denser vertical resolution compared with ours, which would cause data insufficiency when we transfer a KITTI-trained model to the CMM scenario. So the data collected from our LiDAR will not face the data insufficiency issue when used for the model trained on NuScenes. Thus, we applied the NuScenes dataset to train our detection model.

C. Detection

Fig. 12 demonstrates several frames of the CMM FOS testing results. The ego vehicle equipped with the CMM onboard system is marked by a red rectangle in each figure. In the GUI, the orange icons represent the GPS locations of the ego vehicle, while the blue ones denote vehicles detected by the roadside LiDAR. Additionally, pedestrians are also detected and shown in the GUI with top-view pedestrian icons (shown in the video). The detection accuracy is evaluated by the *Confusion Matrix*. Specifically, the detection results can be categorized into four classes:

TABLE II

DETECTION PERFORMANCE IN PRIMARY PERCEPTION AREA

Ground Truth	TP	FP	Precision	Recall	Miss
1661	1389	43	96.99%	83.62%	16.38%

- **True Positive (TP)**: the number of cases predicted as positive by the classifier when they are indeed positive, i.e., a vehicle object is detected as a vehicle.
- **False Positive (FP)** = the number of cases predicted as positive by the classifier when they are indeed negative, i.e., a non-vehicle object is detected as a vehicle.
- **True Negative (TN)** = the number of cases predicted as negative by the classifier when they are indeed negative, i.e., a non-vehicle object is detected as a non-vehicle object.
- **False Negative (FN)** = the number of cases predicted as negative by the classifier when they are indeed positive, i.e., a vehicle is detected as a non-vehicle object.

Precision is the ability of the detector to identify only relevant objects, i.e., vehicles and pedestrians in this paper. It is the proportion of correct positive predictions and is given by

$$Precision = \frac{TP}{TP + FP} = \frac{TP}{\text{\# of all detections}} \tag{22}$$

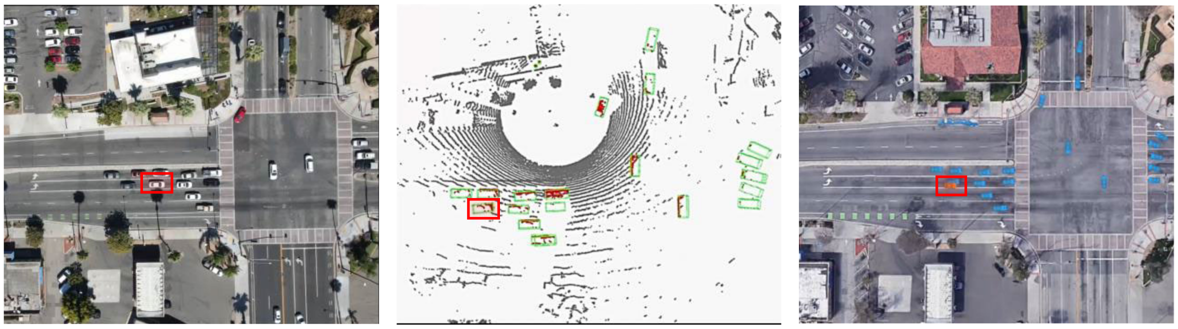
Recall is a metric that measures the ability of the detector to find all the relevant cases (that is, all the ground truths). It is the proportion of true positive detected among all ground-truth (i.e., real vehicles) and is defined as

$$Recall = \frac{TP}{TP + FN} = \frac{TP}{\text{\# of all ground truth}} \tag{23}$$

In terms of the perspective for traffic surveillance, we define another metric named *Miss* which measures the portion of "missing" vehicles (that are not detected) and is defined by

$$Miss = \frac{FN}{TP + TN} = \frac{\text{\# of all missing vehicles}}{\text{\# of all ground truth}} \tag{24}$$

To evaluate detection performance in the primary perception area, we randomly select 130 frames of testing data and manually label them based on the drone's view. A total of 1661 vehicles are labeled as the ground truth and the detection accuracy is evaluated based on the three aforementioned parameters. Table II summarizes the evaluation results.



(a) Drone's view -- ground truth. (b) View of 3D bounding box. (c) GUI view -- reconstruction.

Fig. 12. Examples of the CMM FOS testing results from different perspectives (The ego-vehicle is marked by red boxes).

TABLE III
DETECTION PERFORMANCE IN GENERAL PERCEPTION AREA

# of GT	# of TP under different error condition			
	err = 3m	err = 2m	err = 1m	err = 0.5m
1365	1181	1174	871	722
Recall	86.52%	86.01%	63.81%	52.89%
Miss	13.48%	13.99%	36.19%	47.11%

TABLE IV
TRACKING PERFORMANCE IN GENERAL PERCEPTION AREA

# of GT	# of IDTP under different error condition			
	err = 3m	err = 2m	err = 1m	err = 0.5m
1365	1004	1000	747	663
ID-Recall	73.55%	73.26%	54.73%	48.57%
ID-Miss	26.45%	26.74%	45.27%	51.43%

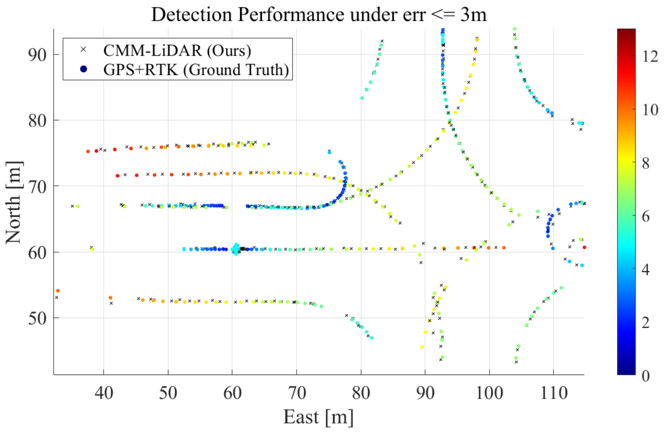


Fig. 13. Detection performance in general perception area and vehicle speed (m/s) represented by the color bar.

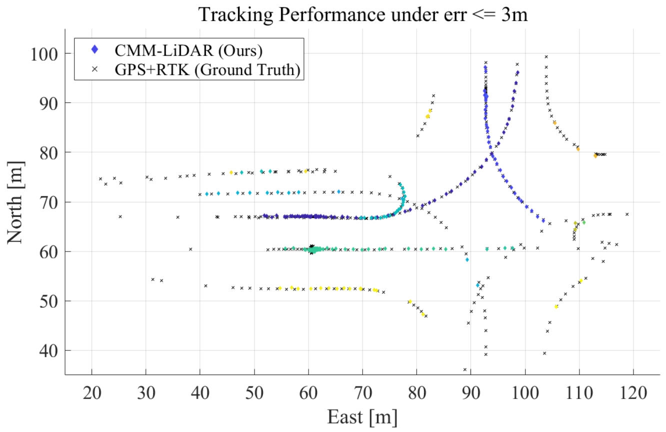


Fig. 14. Tracking performance in general perception area (id represented by colors).

For the detection performance in the general perception area, the APE process proposed in Section IV-I is applied. Since APE cannot generate FP data, Recall and Miss are evaluated and demonstrated in Table III. Furthermore, the qualitative performance of detection under different velocities is shown in Fig. 13.

D. Tracking

For evaluating the tracking performance, information matrix $\mathcal{S}$ generated by the APE approach is applied. Specifically, the tracking performance is calculated using the ID-Recall matrix, which is defined below:

$$\text{ID-Recall} = \frac{\text{IDTP}}{\text{IDTP} + \text{IDFN}} \tag{25}$$

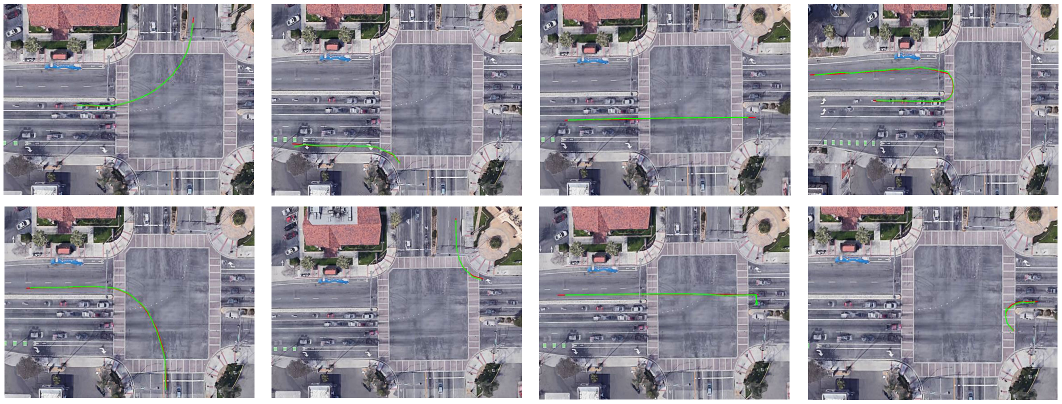
where IDTP and IDFN represent the number of TP matching and FN matching, respectively.

Table IV shows the numerical evaluation of the tracking performance. Under a 2 to 3-meter localization-error condition, the tracking ID-Recall can achieve over 73%. Even under sub-meter level localization requirements, our CMM system can still give a tracking recall performance of around 50%.

More detailed tracking performance with respect to the ground truth is shown in Fig. 14. Tracking performance shows a significant drop when it locates away from the LiDAR (roughly at [70 East, 85 North]). Thus, a second LiDAR would be required at the opposite corner to make the whole intersection be covered at a satisfactory tracking performance.

E. Localization

This section analyzes the localization performance of our CMM field operational system. To evaluate localization accuracy, a multi-sensor-based localization system is applied to



(a) Left Turn (b) Right Turn (c) Straight (d) U Turn

Fig. 15. Trajectories of different driving scenarios for CMM FOS (green) and ground truth (red).

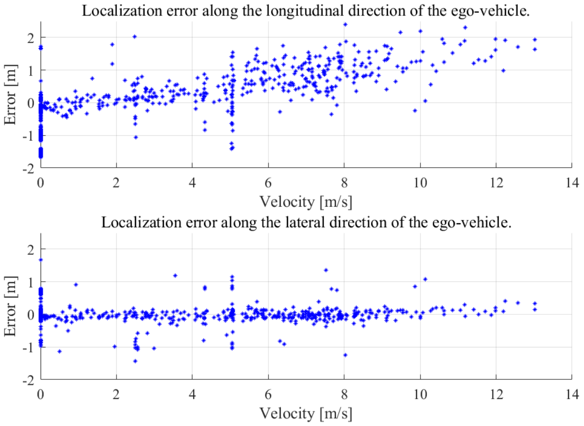


Fig. 16. Localization error along the longitudinal and lateral direction of the ego-vehicle.

measure the ground truth location of the ego-vehicle. This multi-sensor system consists of a GPS receiver enabled with Real-Time Kinematic (RTK) positioning and an Inertial Measurement Unit (IMU). Since this system can achieve centimeter-level positioning, the measurement generated by this GPS-RTK-IMU positioning system is used as the ground truth to assess the CMM system.

Root-mean-square error (RMSE) is applied for evaluating the localization performance of the CMM system. The RMSE for longitudinal localization and lateral localization, $RMSE_{lon}$, and $RMSE_{lat}$ are $0.69m$ and $0.33m$ respectively. The overall RMSE for the CMM localization system $RMSE$ is $0.76m$. Meanwhile, we also provide the evaluation results using mean absolute error (MAE), which are $0.19m$ for lateral MAE, and $0.47m$ for longitudinal MAE.

Fig. 16 demonstrates the localization error along the longitudinal and lateral direction of the ego-vehicle. A positive correlation can be identified in the longitudinal localization error, which can be explained by the non-synchronization between the GPS and LiDAR systems. In another word, the time gap is amplified by the longitudinal motion, because of its higher speed than the lateral movement.

Additionally, visualization results are shown in terms of different driving scenarios, including 1) left turn, 2) right turn, 3) going straight, and 4) U-turn. The trajectories of ego-vehicle with four driving scenarios are extracted and visualized in Fig. 15. The trajectories generated by our CMM system (green curves) highly match the ground truth generated by the onboard GPS-RTK-IMU positioning system (red curves under the green one).

F. Latency

As for a field operation system (FOS), it is of great significance to analyze the latency of the whole system. As depicted in Fig. 17, the latency of the whole CMM FOS pipeline to process one frame of data can be analyzed by breaking down the whole workflow into three main phases:

Phase 1 – Sensor Side: Time elapsed from the start till the edge server receives the sensor data. Specifically, in the *sensor processing* stage, the sensor collects the raw data and processes it into a transformable format via its embedded system. For data retrieving, the processed data can be transmitted to the edge server via the Local Area Network (LAN). The time consumption is certified by the manufacturer.

Phase 2 – Edge-Server Side: Time elapsed from the moment when sensor data is received by the edge server till the instance when perception data is encoded and sent out to the cloud server. The edge server is responsible for generating the object-level perception data, including 3D object detection, 3D multi-object tracking, and geodetic localization. Since these modules are running in chronological order, the time consumption for each module is measured by the starting and ending timestamps of each function.

Phase 3 – Cloud & Onboard Side: Time elapsed from the moment when perception data is sent from the edge server till the instance when reconstructed traffic environments are displayed on the onboard GUI. Since the CMM system tends to serve all the road users with connectivity, a cloud server is used for data acquisition, synchronization, and distribution of processed data (after edge computing). The onboard computer, i.e., the tablet utilized in this study, decodes the perception data, reconstructs the traffic environment, and displays it on the GUI. Time consumption for this phase is measured by the timestamps from the onboard end to the edge-server end.

As shown in Fig. 17, the total latency is about $285ms - 335ms$, whose variance mainly results from the

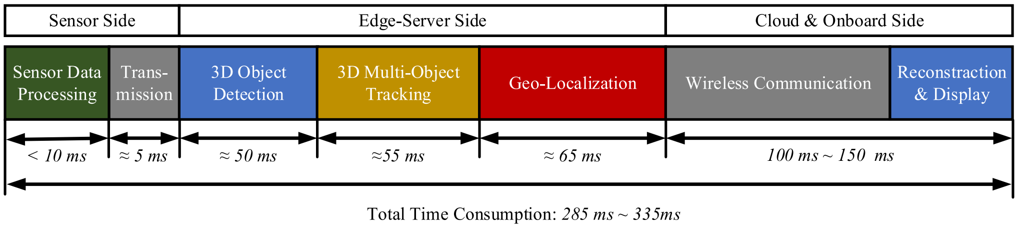


Fig. 17. The diagram of time consumption per frame at different stages in CMM FOS.

fluctuation of communication. However, during the field testing, we find out that the time consumption of every single computational module may vary within a certain range. For example, the object-tracking and geo-localization modules have a larger variance compared with the object detection model, which may be caused by the change in the number of detected objects.

To reduce the latency of the whole system, there are several ways that can be applied in the future. For example, several *for loops* and *external python packages* are implemented in the software for tracking and localization parts which mainly account for the surprisingly high computational cost at the perception end. Therefore, programming optimization can be applied to further reduce computational time. Another way to speed up the whole process is to improve the hardware's computational performance for edge servers and onboard computers.

VI. Conclusion and Discussion

In this study, we introduce the concept of Cyber Mobility Mirror (CMM) and develop a CMM Field Operational System at a real-world intersection as a prototype for enabling Cooperative Driving Automation (CDA). It leverages high-fidelity roadside sensors (e.g., LiDAR) to detect, classify, track and reconstruct object-level traffic information in real time, which can lay a foundation of environment perception for various kinds of CDA applications in mixed traffic. Testing results prove the feasibility of the CMM concept and also demonstrate satisfactory system performance in terms of real-time high-fidelity traffic surveillance. The overall perception can achieve 96.99% precision and 83.62% recall for detection and 73.55% ID-recall for tracking. Additionally, the average geo-localization error of the system is $0.19m$, and $0.47m$ for lateral and longitudinal direction and real-time traffic conditions can be displayed at a frequency of $3-4Hz$.

Based on this prototype CMM FOS, several future directions for improving the system performance may include:

- ***Perception Accuracy***: The current domain adaptation approach to data transformation is preliminary, which can be further improved by box distribution normalization [48] or pseudo labels [49]. Additionally, Style Transfer [50] can also be a promising solution to improving the capabilities of domain adaptation and cost-effectiveness;
- ***Perception Range***: The current CMM FOS only involves one LiDAR sensor and thus can only cover a limited area of the whole intersection. To extend the perception range of the CMM system, we plan to set up several sensors including both LiDARs and cameras to cover multiple intersections to achieve a corridor-level cooperative perception system;
- ***Real-time Performance***: The time consumption can be mainly reduced from the edge-server side, i.e., optimizing the software programming in the tracking and localization parts. Besides, upgrading the hardware equipment can also improve the real-time processing speed.

This paper intends to provide a field operational system of a novel concept of the roadside sensor-based high-fidelity cooperative perception system, named CMM, which can provide foundations and inspirations for future work. By leveraging the high-fidelity roadside sensing information available from the CMM system, plenty of subsequent CDA applications (e.g., CACC, advanced intersection management, cooperative eco-driving) can be revisited for real-world implementation in the mixed traffic environment.

References

[1] D. J. Fagnant and K. Kockelman, "Preparing a nation for autonomous vehicles: Opportunities, barriers and policy recommendations," *Transp. Res. A, Policy Pract.*, vol. 77, pp. 167–181, Jul. 2015.

[2] J. A. Misener and S. E. Shladover, "PATH investigations in vehicle-roadside cooperation and safety: A foundation for safety and vehicle-infrastructure integration research," in *Proc. IEEE Intell. Transp. Syst. Conf.*, Sep. 2006, pp. 9–16.

[3] R. Stahlmann, A. Festag, A. Tomatis, I. Radusch, and F. Fischer, "Starting European field tests for CAR-2-X communication: The DRIVE C2X framework," in *Proc. 18th ITS World Congr. Exhib.*, 2011, p. 12.

[4] USDOT. (May 2021). *Carma Program Overview*. [Online]. Available: https://highways.dot.gov/research/operations/CARMA

[5] EUCAR. (May 2021). *Autonet2030*. [Online]. Available: https://www.autonet2030.eu/

[6] 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 Trans. Intell. Transp. Syst.*, vol. 20, no. 10, pp. 3782–3795, Oct. 2019.

[7] E. Yurtsever, J. Lambert, A. Carballo, and K. Takeda, "A survey of autonomous driving: Common practices and emerging technologies," *IEEE Access*, vol. 8, pp. 58443–58469, 2020.

[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," 2022, *arXiv:2201.11871*.

[9] S. R. E. Datondji, Y. Dupuis, P. Subirats, and P. Vasseur, "A survey of vision-based traffic monitoring of road intersections," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 10, pp. 2681–2698, Oct. 2016.

[10] 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," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2022, pp. 1366–1373.

[11] Y. Song, H. Zhang, Y. Liu, J. Liu, H. Zhang, and X. Song, "Background filtering and object detection with a stationary LiDAR using a layer-based method," *IEEE Access*, vol. 8, pp. 184426–184436, 2020.

[12] Z. Zou, K. Chen, Z. Shi, Y. Guo, and J. Ye, "Object detection in 20 years: A survey," 2019, *arXiv:1905.05055*.

[13] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, "Statistic and knowledge-based moving object detection in traffic scenes," in *Proc. IEEE Intell. Transp. Syst.*, Oct. 2000, pp. 27–32.

[14] S. Aslani and H. Mahdavi-Nasab, "Optical flow based moving object detection and tracking for traffic surveillance," *Int. J. Elect., Comput., Energetic, Electron. Commun. Eng.*, vol. 7, no. 9, pp. 1252–1256, 2013.

[15] A. Boukerche and Z. Hou, "Object detection using deep learning methods in traffic scenarios," *ACM Comput. Surv.*, vol. 54, no. 2, pp. 1–35, Mar. 2022.

[16] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 779–788.

[17] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 28, 2015, pp. 91–99.

[18] C.-J. Li, Z. Qu, S.-Y. Wang, and L. Liu, "A method of cross-layer fusion multi-object detection and recognition based on improved faster R-CNN model in complex traffic environment," *Pattern Recognit. Lett.*, vol. 145, pp. 127–134, May 2021.

[19] A. Mhalla, T. Chateau, S. Gazzah, and N. E. B. Amara, "An embedded computer-vision system for multi-object detection in traffic surveillance," *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 11, pp. 4006–4018, Nov. 2019.

[20] J. Lian, Y. Yin, L. Li, Z. Wang, and Y. Zhou, "Small object detection in traffic scenes based on attention feature fusion," *Sensors*, vol. 21, no. 9, p. 3031, Apr. 2021.

[21] J. Wu, H. Xu, J. Zheng, and J. Zhao, "Automatic vehicle detection with roadside LiDAR data under rainy and snowy conditions," *IEEE Intell. Transp. Syst. Mag.*, vol. 13, no. 1, pp. 197–209, Spring 2021.

[22] L. Zhang, J. Zheng, R. Sun, and Y. Tao, "GC-Net: Gridding and clustering for traffic object detection with roadside LiDAR," *IEEE Intell. Syst.*, vol. 36, no. 4, pp. 104–113, Jul. 2021.

[23] SAE. (2021). *Taxonomy and Definitions for Terms Related to Cooperative Driving Automation for on-Road Motor Vehicles J3216_202005*. [Online]. Available: https://www.sae.org/standards/content/j3216_202005/

[24] E. V. Cuevas, D. Zaldivar, and R. Rojas, "Kalman filter for vision tracking," Fachbereich Mathematik und Informatik; Serie B, Informatik, Freie Universität Berlin, Berlin, Germany, Tech. Rep. 05-12, 2005.

[25] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2016, pp. 3464–3468.

[26] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3645–3649.

[27] C. Chen, B. Liu, S. Wan, P. Qiao, and Q. Pei, "An edge traffic flow detection scheme based on deep learning in an intelligent transportation system," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 3, pp. 1840–1852, Mar. 2020.

[28] Y. Cui, H. Xu, J. Wu, Y. Sun, and J. Zhao, "Automatic vehicle tracking with roadside LiDAR data for the connected-vehicles system," *IEEE Intell. Syst.*, vol. 34, no. 3, pp. 44–51, May/Jun. 2019.

[29] A. Kampker, M. Sefati, A. S. A. Rachman, K. Kreisköther, and P. Campoy, "Towards multi-object detection and tracking in urban scenario under uncertainties," in *Proc. 4th Int. Conf. Vehicle Technol. Intell. Transp. Syst.*, 2018, pp. 156–167.

[30] J. C. Herrera and A. M. Bayen, "Traffic flow reconstruction using mobile sensors and loop detector data," UC Berkeley, Univ. California Transp. Center, Berkeley, CA, USA, Tech. Rep. 01-18, 2007. [Online]. Available: https://escholarship.org/uc/item/6v40f0bs

[31] D. Jiang, W. Wang, L. Shi, and H. Song, "A compressive sensing-based approach to end-to-end network traffic reconstruction," *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 507–519, Jan. 2020.

[32] M. Cao, L. Zheng, W. Jia, and X. Liu, "Joint 3D reconstruction and object tracking for traffic video analysis under IoV environment," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 6, pp. 3577–3591, Jun. 2021.

[33] Q. Rao and S. Chakraborty, "In-vehicle object-level 3D reconstruction of traffic scenes," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 12, pp. 7747–7759, Dec. 2021.

[34] J. Wu, H. Xu, Y. Zhang, and R. Sun, "An improved vehicle-pedestrian near-crash identification method with a roadside LiDAR sensor," *J. Saf. Res.*, vol. 73, pp. 211–224, Jun. 2020.

[35] Z. Bai, P. Hao, W. Shangguan, B. Cai, and M. J. Barth, "Hybrid reinforcement learning-based eco-driving strategy for connected and automated vehicles at signalized intersections," *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 15850–15863, Sep. 2022.

[36] Z. Wang, Y. Bian, S. E. Shladover, G. Wu, S. E. Li, and M. J. Barth, "A survey on cooperative longitudinal motion control of multiple connected and automated vehicles," *IEEE Intell. Transp. Syst. Mag.*, vol. 12, no. 1, pp. 4–24, Dec. 2020.

[37] D. Oswald et al., "Development of an innovation corridor testbed for shared electric connected and automated transportation," Nat. Center Sustain. Transp. (NCST)(UTC), Center Environ. Res. Technol., Univ. California, Riverside, Riverside, CA, USA, Tech. Rep. NCST-UCR-RR-21-20, 2021.

[38] H. Caesar et al., "NuScenes: A multimodal dataset for autonomous driving," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11621–11631.

[39] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "PointPillars: Fast encoders for object detection from point clouds," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 12697–12705.

[40] W. Liu et al., "SSD: Single shot MultiBox detector," in *Proc. Eur. Conf. Comput. Vis.* Cham, Switzerland: Springer, 2016, pp. 21–37.

[41] Y. Yan, Y. Mao, and B. Li, "SECOND: Sparsely embedded convolutional detection," *Sensors*, vol. 18, no. 10, p. 3337, 2018.

[42] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal loss for dense object detection," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2980–2988.

[43] S. Schuhmacher and J. Boehm, "Georeferencing of terrestrial laser-scanner data for applications in architectural modeling," Virtual Reconstruct. Vis. Complex Archit., Venice, Italy, Tech. Rep. 3D-ARCH 2005, Aug. 2005.

[44] Z. Bai, G. Wu, X. Qi, Y. Liu, K. Oguchi, and M. J. Barth, "Cyber mobility mirror for enabling cooperative driving automation in mixed traffic: A co-simulation platform," 2022, *arXiv:2201.09463*.

[45] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proc. Conf. Robot Learn.*, 2017, pp. 1–16.

[46] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3354–3361.

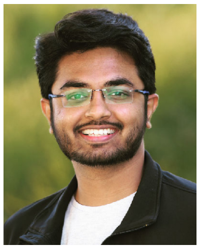
[47] P. Sun et al., "Scalability in perception for autonomous driving: Waymo open dataset," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 2446–2454.

[48] Y. Wang et al., "Train in Germany, test in the USA: Making 3D object detectors generalize," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 11713–11723.

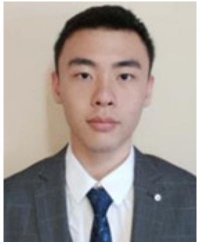
[49] J. Yang, S. Shi, Z. Wang, H. Li, and X. Qi, "ST3D: Self-training for unsupervised domain adaptation on 3D object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 10368–10378.

[50] C.-T. Lin, S.-W. Huang, Y.-Y. Wu, and S.-H. Lai, "GAN-based day-to-night image style transfer for nighttime vehicle detection," *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 2, pp. 951–963, Feb. 2021.

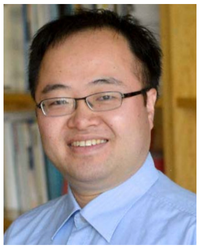
Zhengwei Bai (Graduate Student Member, IEEE) received the B.E. and M.S. degrees from Beijing Jiaotong University, Beijing, China, in 2017 and 2020, respectively. He is currently pursuing the Ph.D. degree in electrical and computer engineering with the University of California at Riverside, Riverside, CA, USA. His research interests include object detection and tracking, cooperative perception, decision-making, motion planning, and cooperative driving automation (CDA). He is a member of the ASCE Artificial Intelligence in Transportation Committee. He serves as a Review Editor for *Urban Transportation Systems and Mobility*.



Saswat P. Nayak received the B.Tech. degree in electrical engineering from the National Institute of Technology, Rourkela, India, in 2018. He is currently pursuing the Ph.D. degree with the Center of Environmental Research and Technology (CE-CERT), University of California at Riverside, Riverside, CA, USA. He was a Project Associate with the Department of Aerospace Engineering, Indian Institute of Technology Kanpur, India, from 2018 to 2019. His main research interests include vehicle positioning and localization in mixed traffic scenarios, multi-sensor fusion, and connected vehicle applications.



Xuanpeng Zhao received the B.E. degree in electrical engineering from Shanghai Maritime University in 2019 and the M.S. degree in electrical engineering from the University of California at Riverside, Riverside, CA, USA, where he is currently pursuing the Ph.D. degree in electrical and computer engineering. His research interests include cybersecurity and connected and automated vehicle technology.



Guoyuan Wu (Senior Member, IEEE) received the Ph.D. degree in mechanical engineering from the University of California at Berkeley, Berkeley, CA, USA, in 2010. He is currently an Associate Researcher and an Associate Adjunct Professor with the Bourns College of Engineering–Center for Environmental Research and Technology (CE-CERT), Department of Electrical and Computer Engineering, University of California at Riverside, Riverside, CA, USA. His research interests include the development and evaluation of sustainable and intelligent transportation system (SITS) technologies. He is a member of the Vehicle-Highway Automation Standing Committee (ACP30) of the Transportation Research Board (TRB), a Board Member of the Chinese Institute of Engineers Southern California Chapter (CIE-SOCAL), and a member of the Chinese Overseas Transportation Association (COTA). He was a recipient of the Vincent Bendix Automotive Electronics Engineering Award. He serves as an Associate Editor for journals, including IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, *SAE International Journal of Connected and Automated Vehicles*, and IEEE OPEN JOURNAL OF INTELLIGENT TRANSPORTATION SYSTEMS.

Matthew J. Barth (Fellow, IEEE) received the M.S. and Ph.D. degrees in electrical and computer engineering from the University of California at Santa Barbara, Santa Barbara, CA, USA, in 1985 and 1990, respectively. He is currently the Yeager Families Professor with the College of Engineering, University of California at Riverside, Riverside, CA, USA. He is also the Director of the Center for Environmental Research and Technology. His current research interests include ITS and the environment, transportation/emissions modeling, vehicle activity analysis, advanced navigation techniques, electric vehicle technology, and advanced sensing and control. He has been active in the IEEE Intelligent Transportation System Society for many years, serving as a Senior Editor for IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS and IEEE TRANSACTIONS ON INTELLIGENT VEHICLES. He served as the IEEE ITSS President from 2014 to 2015. He is the IEEE ITSS Vice President of Education.

Xuewei Qi (Member, IEEE) received the M.S. degree in engineering from the University of Georgia, USA, in 2013, and the Ph.D. degree in electrical and computer engineering from the University of California at Riverside, Riverside, CA, USA, in 2016. He is currently the Principal AI Researcher with Toyota Motor North America, Research and Development Labs. His recent research interests include deep learning, autonomous vehicles, perception and sensor fusion, reinforcement learning, and decision-making. He is serving as a member for several standing committees of the Transportation Research Board.

Yongkang Liu received the M.S. and Ph.D. degrees in electrical engineering from The University of Texas at Dallas in 2017 and 2021, respectively. He is currently a Research Engineer with Toyota Motor North America, Research and Development Labs, InfoTech Labs. His current research interests include in-vehicle systems and advancements in intelligent vehicle technologies.

Emrah Akin Sisbot (Member, IEEE) received the Ph.D. degree in robotics and artificial intelligence from Paul Sabatier University, Toulouse, France, in 2008. He was a Post-Doctoral Research Fellow with LAAS-CNRS, Toulouse, and with the University of Washington, Seattle, WA, USA. He is currently a Principal Engineer with Toyota Motor North America, Research and Development Labs, InfoTech Labs, Mountain View, CA, USA. His current research interests include real-time intelligent systems, robotics, and human–machine interaction.

Kentaro Oguchi received the M.S. degree in computer science from Nagoya University. He is currently the Director of Toyota Motor North America, Research and Development Labs, InfoTech Labs. His team is responsible for creating intelligent connected vehicle architecture that takes advantage of novel AI technologies to provide real-time services to connected vehicles for smoother and efficient traffic, intelligent dynamic parking navigation and vehicle guidance to avoid risks from anomalous drivers. His team also creates technologies to form a vehicular cloud using V2X technologies.