

Robust Roadside Perception: an Automated Data Synthesis Pipeline Minimizing Human Annotation

Rusheng Zhang¹, Depu Meng¹, Lance Bassett², Shengyin Shen³, Zhengxia Zou⁴ and Henry X. Liu^{1,5*}

Abstract—Recently, advancements in vehicle-to-infrastructure communication technologies have elevated the significance of infrastructure-based roadside perception systems for cooperative driving. This paper delves into one of its most pivotal challenges: data insufficiency. The lacking of high-quality labeled roadside sensor data with high diversity leads to low robustness, and low transfer-ability of current roadside perception systems. In this paper, a novel solution is proposed to address this problem that creates synthesized training data using Augmented Reality. A Generative Adversarial Network is then applied to enhance the reality further, that produces a photo-realistic synthesized dataset that is capable of training or fine-tuning a roadside perception detector which is robust to different weather and lighting conditions.

Our approach was rigorously tested at two key intersections in Michigan, USA: the Mcity intersection and the State St./Ellsworth Rd roundabout. The Mcity intersection is located within the Mcity test field, a controlled testing environment. In contrast, the State St./Ellsworth Rd intersection is a bustling roundabout notorious for its high traffic flow and a significant number of accidents annually. Experimental results demonstrate that detectors trained solely on synthesized data exhibit commendable performance across all conditions. Furthermore, when integrated with labeled data, the synthesized data can notably bolster the performance of pre-existing detectors, especially in adverse conditions.

I. INTRODUCTION

Recently, with the rapid development in vehicle-to-infrastructure (V2I) communications technologies, the infrastructure-based perception system to support autonomous driving has attracted significant attention. Sensors installed on the roadside detect vehicles in the region-of-interest in real-time, and forward the perception results to Connected Automated Vehicles (CAVs) with short latency via V2I communications, broadcasting Basic Safety messages defined in SAE J2735 [1], [2] or Sensor Data Sharing Message defined in SAE J3224 [3]. The roadside sensors are usually positioned at fixed locations on the roadside, typically at elevated heights, providing a broader perspective, fewer obstructed objects, blind spots, and less environmental variability compared to the sensors onboard the vehicles. Consequently, the perception results obtained from the

This work is part of US DoT Smart Intersection Project.

¹Rusheng Zhang, Depu Meng, and Henry X. Liu are with the Department of Civil and Environmental Engineering, University of Michigan, Ann Arbor.

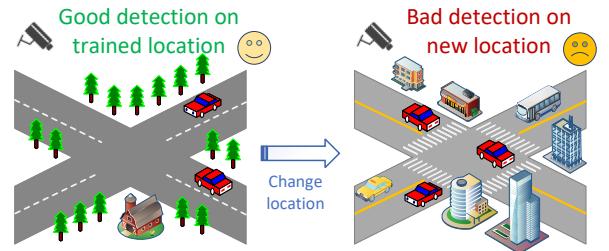
²Lance Bassett is with the Department of Electrical Engineering and Computer Science, University of Michigan, Ann Arbor.

³Shengyin Shen is with University of Michigan Transportation Research Institute.

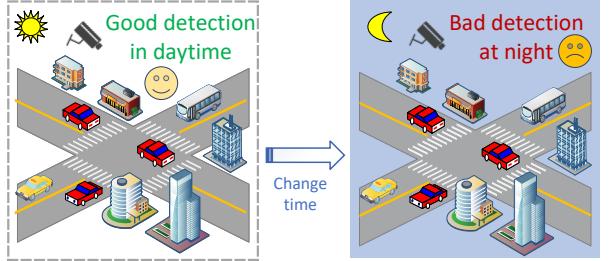
⁴Zhengxia Zou is with the Beihang University.

⁵Henry X. Liu is also with Mcity.

*Corresponding author - Henry X. Liu (henryliu@umich.edu).



(a) A detector trained in one location may perform poorly when deploying the same detector to another location.



(b) A detector trained with day time data may perform poorly on the same location at night.

Fig. 1: An illustrative figure that shows the issues current roadside perception systems face due to the data insufficiency.

roadside sensors can complement the onboard perception of CAVs, resulting in a more comprehensive, consistent, and accurate understanding of the surrounding scene, particularly in complex scenarios and challenging weather and lighting conditions.

Although it is commonly believed that roadside perception is less complex than onboard perception due to the significantly lower environmental variability and fewer occluded objects, roadside perception presents its own unique set of challenges. One of the most crucial challenges is data insufficiency, specifically the lack of high-quality and diverse labeled sensor data collected from the roadside. Acquiring roadside data with a sufficient level of diversity (through the deployment of numerous sensors along the roadside) is expensive compared to onboard perception, primarily due to the high cost of installation. Furthermore, obtaining large quantities of labeled data is even more costly, given the high expense associated with labeling. Currently, high-quality labeled roadside perception data are generally obtained from few locations with limited environmental diversity.

The aforementioned challenge of data insufficiency presents significant practical issues in real-world deployment. Figure 1

provides illustrative examples on these issues. In Figure 1a, the performance of the detector trained on data from one location is severely compromised when applied to a new location. Similarly, in Figure 1b, the training dataset lacks images taken at night, resulting in poor performance during nighttime, even at the same location. Clearly, these challenges impede the widespread deployment of roadside perception systems. Moreover, since roadside perception is considered a supplementary and enhancing approach to onboard vehicle detection, it is expected to have higher requirements for robustness and accuracy compared to onboard perception. Consequently, such demanding nature of roadside perception exacerbates the aforementioned challenge of data-insufficiency.

In this paper, we propose an automated pipeline that generates photo-realistic synthesized dataset for roadside perception using Augmented Reality (AR) [4] and Generative Adversarial Network (GAN) [5]. This synthesized dataset can be used either to train a high-performance roadside perception system, or to fine-tune an existing detector with minimum human labeling effort. Since the data synthesizing quality could be further improved with more unlabeled data, this work can be considered a step forward to large-scale real-world deployment.

The contributions of this paper are as follows:

- 1) Propose an AR rendering pipeline for a roadside perception system, including camera pose estimation, realistic vehicle positioning and heading simulation, and AR rendering. This pipeline generates physically realistic images with annotations.
- 2) Propose a GAN based reality enhancement strategy that processes the physical realistic images obtained from AR and converts them to photo-realistic images.
- 3) Report a thorough field evaluation of the model obtained from the aforementioned pipeline under different lighting and weather conditions that demonstrates the viability of this method in real-world, large-scale deployment.

II. RELATED WORKS

A. Roadside Camera for Vehicle Detection

The existence of roadside sensor-based surveillance systems can be traced back as early as 1986. Early systems aim at traffic monitoring and abnormal behavior detection [6]. These systems consist of single or multiple cameras mounted at a high elevated position. A large number of algorithms have been explored during the last decade for roadside vehicle detection with cameras, to name a few, background subtraction [7], frame difference [8], feature-based detection [9], KanadeLucas-Tomasi tracking [10], cascading classifiers [11] and many more [6]. Recently, deep learning based detection has become the trend [12], [13], [14], [15].

B. Training on Synthesized Data

Acquiring a substantial quantity of data is a costly endeavor, and obtaining annotations further escalates the expenses. As a result, the insufficiency of data presents a widespread challenge in the field of deep learning. To address this challenge,

several synthesized datasets and simulation tools, such as Carla [16], AirSim [17], SYNTHIA [18], GTA5 [19] and VIPER [20], are developed, so that researchers can have access to sufficient annotated data. However, the neural networks trained on these synthesized datasets often perform poorly in the real-world.

There have been many proposals on using synthesized data together with real-world data to mitigate such effects [21]. For example, [22] trains neural networks with synthesized data generated with domain randomization and shows that the network yields better results using the augmented data than with real-world data alone. Generative Adversarial Networks (GANs) [5] represent another promising approach that has recently been explored for data augmentation due to their potential to generate photo-realistic images. In a study conducted by [23], various data augmentation schemes were evaluated, and the use of GANs to generate images in different styles was proposed. [24] introduced SurfelGAN, a method that synthesizes realistic images reconstructed from actual sensor data. Additionally, GANs have been utilized by [25] to translate clear images into different weather conditions, thereby enhancing the robustness of autonomous vehicles in adverse weather. Finally, [26] proposed an image composition method that employs both local and global discriminators to achieve realistic shadow and texture effect.

In the field of autonomous driving, data augmentation with image composition has recently become an interesting topic. Several researches focus on data augmentation for onboard vehicle perception. [27] proposes GeoSim, a geometry-aware image composition process that can augment existing image with dynamic objects. [28] proposes CADSim that recovers photo-realistic 3D traffic participant model from sparse sensor data, to create a large set of 3D model library which can be further used in data augmentation and simulations. These methods can generate photo-realistic augmented data for perception and other safety-critic downstream applications. While these methods can potentially be applied directly, their need for 3D awareness makes them best suited for scenarios where both point cloud and camera images are accessible, especially given the current limited availability of infrastructure data that combines both.

Given the static nature of the background in roadside sensor data, the process of image composition becomes considerably simpler. Unlike onboard dynamic scenes that are replete with challenges such as occlusion, variable lighting, and unpredictable motion, static backgrounds afford a stable canvas on which augmented elements can be added with relative ease. This simplicity has led to the emergence of various methods capable of accomplishing such tasks. In fact, many of the current methods have advanced to a degree that their capabilities surpass the needs of simple roadside data augmentation. It's important to note that the main intent of this paper isn't to claim superiority in this specific domain of data augmentation. Instead, our focus is to establish a methodology pipeline that demonstrates the efficacy of data synthesis in the context of roadside perception. By leveraging the strengths of existing technologies and adapting them for this niche, we aim to open new avenues in the field of roadside cooperative perception.

III. METHODOLOGY

A. Hardware Setup



(a) One of the mast arm in Mcity where the camera is being installed.



(b) The roundabout at the intersection of Ellsworth St. and State St. in Ann Arbor, Michigan, USA

Fig. 2: Two sets of cameras are leveraged at two different locations in this paper. Four cameras are installed at an intersection in Mcity, facing four approaches respectively (Figure 2a) and another four cameras are installed at four corners of a two-lane roundabout (Figure 2b).

The experiments reported in this paper are conducted in two location, one intersection in Mcity, and another roundabout at the intersection of Ellsworth St. and State St. in Ann Arbor, Michigan, USA. Mcity is the world's first controlled environment specifically designed to test the performance and safety of connected and automated vehicle technologies [29]. Four pinhole cameras are installed on the mast arms of the intersection at Mcity, each facing one approach. The two-lane roundabout is located at Ellsworth St. and State St. in Ann Arbor, Michigan, USA. Four pinhole cameras are installed at the four corner of the roundabout, recording real car flow. Figure 2 shows images of these two experimental position.

B. Core Idea

Our solution to create a roadside perception system includes two main steps:

- 1) a data synthesizing pipeline that generates labeled training data
- 2) deep learning based vehicle detection and localization model trained with the synthesized data

The data synthesis pipeline renders realistic vehicles onto the background images obtained from the cameras and simultaneously generates the corresponding annotations. The components in this step are discussed in section III-C, III-D, III-E, and III-F.

The synthesized data are used to train a YOLOX detector [30]. The vehicle detection pipeline utilizes the trained YOLOX detector and generates 3D vehicle location information. The components in this step are discussed in section III-G.

C. The Data Synthesizing Pipeline

Figure 3 illustrates the overall data synthesis pipeline responsible for generating the training dataset. Initially, vehicle locations and headings are generated with a traffic simulator. Then, an AR renderer is employed to project 3D vehicle models onto the background images. During this rendering

process, annotation information for each rendered vehicle is also generated. To bridge the disparity between our synthesized data and the real world, a reality enhancer based on GAN technology is applied to each rendered vehicle. This enhancer effectively translates the vehicles into a more realistic style, thereby reducing the gap between our synthesized data and real-world scenarios.

a) Background images: The background images can be easily estimated with a temporal median filter [31]. One can gather the background images under different conditions to cover the variability of the background for each camera (i.e., different weather and lighting conditions).

b) Traffic simulation: A traffic simulation is performed to generate realistic vehicle trajectories; the heading and location information of the simulated vehicles is further used for AR rendering. This task is accomplished with SUMO, an open-source microscopic mobility simulator [32]. The road map information can be directly imported to SUMO from OpenStreetMap [33], and constant car flows are generated for all maneuvers at the intersection. However, as SUMO only creates vehicles at the center of the lane with fixed headings, a domain randomization step is implemented to introduce variability. This involves applying a random positional and heading offset to each vehicle. The positional offset follows a normal distribution with a variance of 0.5 meters in both vehicles' longitudinal and latitudinal directions, while the heading offset follows an uniform distribution from -5° to 5° .

c) 3D vehicle models: The 3D vehicle models used in this work are obtained from the Shapenet repository, which is an ongoing effort to establish a richly-annotated, large-scale dataset of 3D shapes [34]. In total, we pick more than 200 vehicle 3D models to yield a diverse model set. For each vehicle in SUMO simulation, a random model will be assigned and rendered onto the background images.

D. Camera Pose Estimation

To correctly render 3D models onto the background images, the camera pose, including the camera rotation and camera translation in the world coordinate system, needs to be estimated. Standard camera extrinsic calibration with a large checkerboard requires on-site operation by experienced technicians, which will add complexity to the deployment pipeline, particularly in scenarios involving large-scale deployment. In this paper, we introduce a landmark based camera pose estimation method for roadside cameras that eliminates the need for field operation. Figure 4 provides an overview of this method. Our method considers a few landmarks, marked as $P_1, P_2, P_3, \dots, P_n$ in the figure, that are both observable from the camera view and the satellite image. These landmarks provide a set of correspondences between the world coordinate system and their projections on the camera plane. Since camera intrinsic parameters are known, the camera pose can be solved with a Perspective-n-Point (PnP) solver using the n pairs of the world-to-image correspondences obtained by these landmarks [35]. Figure 5 shows the landmarks we used for one of our deployed cameras (the westbound approach view) at

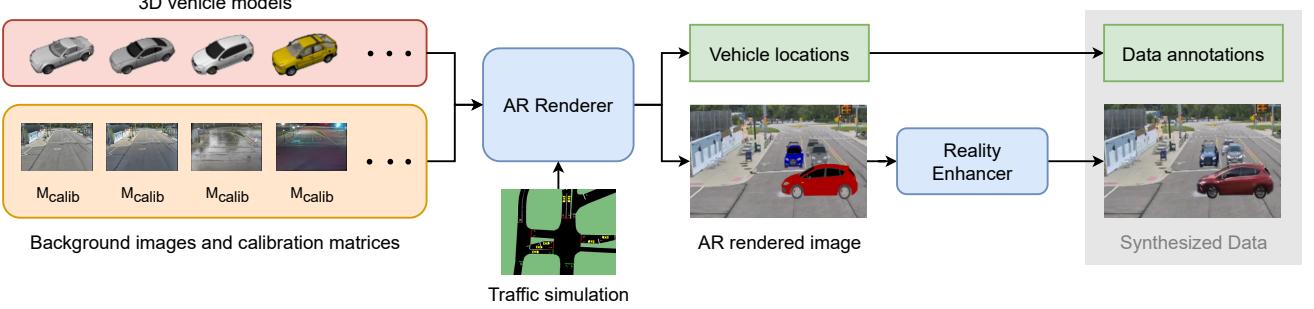


Fig. 3: Data synthesizing pipeline to generate realistic data: an AR renderer renders 3D model onto the real background with traffic simulation data, a GAN-based reality enhancer is then applied to make the rendered vehicle photo-realistic.

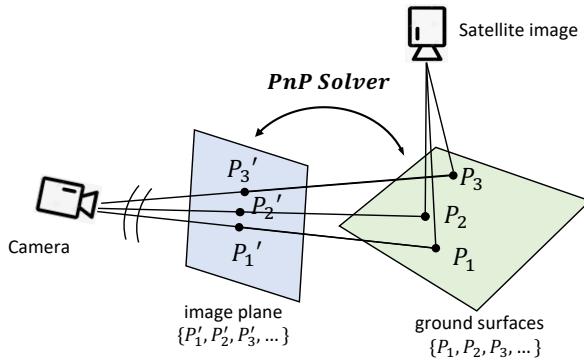
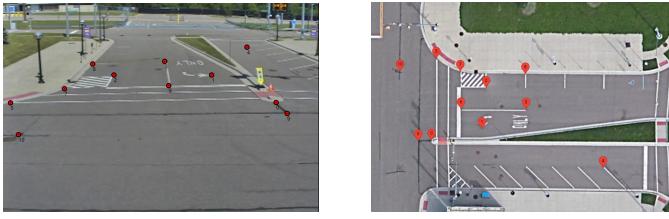


Fig. 4: Illustration figure for pose estimation.

the intersection of Mcity. As shown in the images, we employ approximately ten landmarks for each camera to accomplish the pose estimation task. To alleviate the typically tedious nature of this process for human operators that manually match landmarks with image pixels, we have developed a user-friendly online UI interface for public use, which simplifies this task [36].



(a) Landmarks selected for camera pose estimation.

(b) The corresponding landmarks from satellite image.

Fig. 5: Pose estimation landmark selection on one of the camera (westbound approach of the intersection).

E. Augmented Reality Rendering

For each camera, the camera intrinsic matrix K is known; the extrinsic matrix $[R|T]$ can be estimated using the method described in section III-D. Here, R is a 3×3 rotation matrix and T is a 3×1 translation matrix. For any point in the world coordinate system, the corresponding image pixel location can be found with the classic camera transformation [37]:

$$Y = K \times [R|T] \times X \quad (1)$$

where X is a homogeneous world 3D coordinate of size 4×1 , and Y is a homogeneous 2D coordinate of size 3×1 . Equation (1) is used for rendering models onto the image, as well as generating ground-truth labels that maps each vehicle's 3D bounding box in the image. In this work, the AR rendering task is accomplished with Pyrender, a light-weight AR rendering module for Python [38]. Figure 6 (top row) shows some rendering results.

F. Reality Enhancement

The AR method from section III-E creates vehicles in the foreground over the real-world background images. These foreground vehicles are rendered from 3D models, which are not realistic enough and may negatively impact the real-world performance of the trained detector. To address this issue, we have incorporated a GAN-based reality enhancement component that converts the AR-generated foreground vehicles to a more realistic appearance. Specifically, we have utilized the Contrastive Unpaired Translation (CUT) technique to translate the AR-generated foreground to a realistic image style [39]. The realistic image styles have been learned from BAAI-Vanhee dataset [40], which comprises 2000 roadside camera images. To remove the backgrounds of all images in the dataset, we have employed a salient object detector known as TRACER [41]. The vehicles in the images have been cropped according to the annotation, enabling the CUT model to focus solely on translating the vehicle style rather than the background style. The AR-rendered vehicles have been translated individually and re-rendered to the same position. Figure 6 displays some sample images. The top row of four images are the AR rendered images, the bottom row are the corresponding images after reality enhancement, arranged in the order of eastbound approach, northbound approach, southbound approach and westbound approach camera views.

G. Vehicle Detection and Localization

In the applications for autonomous vehicles, vehicle 3D coordinates are needed. This requires an extra step that lifts the 2D detection results to 3D. This paper adopts a previously reported method to achieve this goal [13], [14]. Figure 7 illustrates an overall pipeline for estimating vehicle 3D coordinates. The 2D detector is trained to detect the vehicles' bottom centers in the image. These bottom center points are mapped

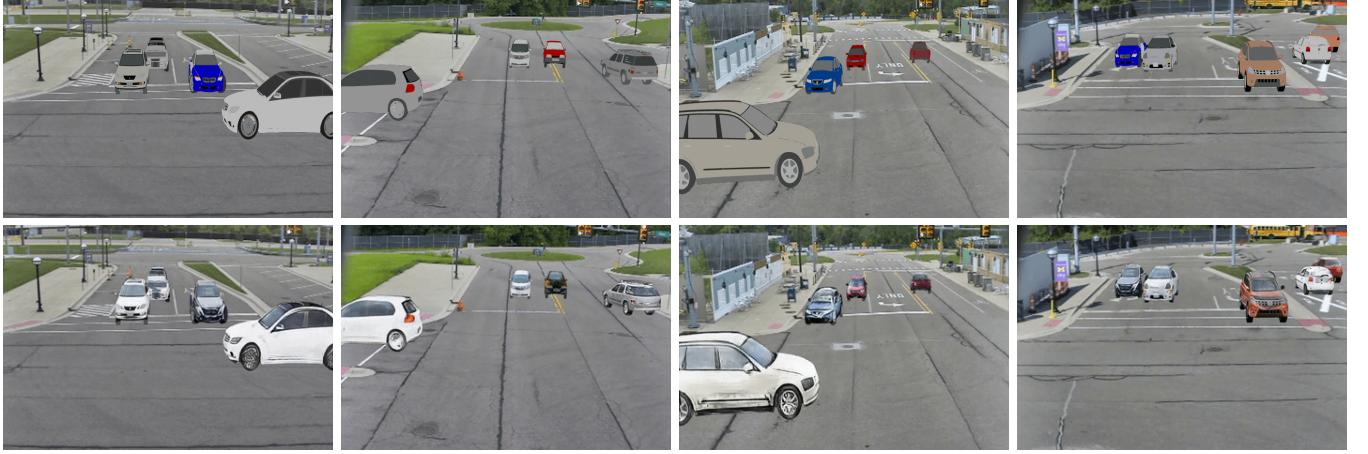


Fig. 6: Examples of (top) Augmented Reality (AR) rendered images, and (bottom) the same images after reality enhancement.

with a homography map to the road plane. The homography matrix can be obtained from the same correspondence set between the real-world and image described in section III-D. We use a YOLOX detector to perform the 2D bottom center detection task [30]. To train our YOLOX model to detect vehicle bottom centers, the data annotations generated in section III-E record bounding boxes that tightly frame the vehicles' 3D bottom in the image. In this way, one can recover the bottom center position from the YOLOX detection results simply by using the center point of the bounding box. As this method has been thoroughly tested, reported and proven to be viable ([13], [14]), and with accurate homography map, the 3D localization evaluation is equivalent to pixel localization performance. in this paper, our main focus is on 2D detection and pixel-level localization, framed with yellow box in figure 7. While the detection and localization technologies are crucial components of our study, it is important to note that our primary focus is not on these technologies per se. The method we employ here, which is among the state-of-the-art in detection and localization, serves as a representative example. Our presented pipeline is designed to be universally compatible with a variety of other advanced methods, including but not limited to YOLO [42], SSD [43], and Faster-RCNN [44].

H. Deployment Strategy

As previously discussed, the roadside vehicle detection strategy presented in this paper lessens the dependence on human labeling. The deployment strategy is as follows:

- Step 1: Install cameras at the roadside, determine the camera's intrinsic matrix, and perform pose estimation.
- Step 2: Following camera installation, begin collecting background images.

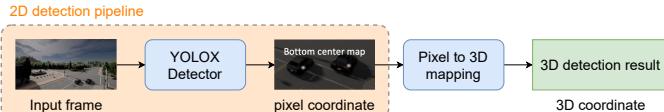


Fig. 7: Complete detection pipeline for 3D vehicle localization and its relationship with our trained YOLOX Detector

- Step 3: Superimpose rendered vehicles onto these collected background images, along with their corresponding labels, to create a synthesized dataset.
- Step 4: Use this synthesized dataset to train a deep learning-based detector. One might opt to initialize training using a pretrained detector from another location or dataset as a starting checkpoint.
- Step 5: Roll out the detector trained on the synthesized dataset.
- Step 6: Periodically amass additional background images to expand the synthesized dataset, thereby enhancing background diversity coverage. Iteratively refine the detector using this continually augmented synthesized dataset.

It's noteworthy that, theoretically, this entire deployment pipeline could operate with full automation, allowing the detector to update and evolve autonomously without, or with minimum human intervention.

IV. PERFORMANCE EVALUATION

A. Training Dataset

Mcity Intersection Mcity is a closed testing facility that does not have urban traffic flow. Therefore, we use pure synthesized images for the detector training. Our training dataset contains 4,000 images in total. We synthesize 1,000 images for each camera view (north, south, east and west). The background images for synthesis are captured and sampled from roadside camera clips with 720×480 resolution over 5 days. For the foreground, we consider all kinds of vehicles (cars, buses, trucks, etc) to be in the same 'vehicle' category. The backgrounds for synthesizing data and those used in the final evaluation are intentionally selected from different days to ensure no repetition. We choose similar weather conditions and made the backgrounds resemble those in the evaluation dataset. We introduce uniformly sample 100 background images from 8am to 8pm for data synthesizing.

State St/Ellsworth Rd Roundabout We also tested our approach on the roundabout at State Street and Ellsworth Road at Ann Arbor, Michigan. A perception system trained with 1K manually labeled images has already been deployed in fall



Fig. 8: Visualization of detection results on various conditions (left two: sunny weather, center two: rainy weather, right two: dark lighting condition). The detector is trained on our synthesized dataset. Red dots represent detected vehicle bottom centers.



Fig. 9: Comparison of detection results of the detector trained on human-labeled real data (top), and the detector trained on a mixed dataset including human-labeled data and synthesized data (bottom) at the State Street/Ellsworth Road intersection. Synthesized data improves the detection quality in harsh conditions (snow weather). Red dots represent detected vehicle bottom centers.

(September, 2022). The performance of which decades in winter (December, 2022). Therefore, we use our synthesis method to generate 1K more images to improve the performance in the winter. The background images for synthesis are captured with different conditions (sunny, cloudy, snow) from different days.

B. Evaluation Metrics

Our evaluation method aligns closely with the standard approach used in general object detection [45]. However, traditional evaluation methods do not sufficiently emphasize localization precision, a critical factor in cooperative driving. Drawing inspiration from the evaluation metrics proposed by nuScenes [46] and CLEAR metrics [47], we have developed an evaluation method similar to general object detection, with a key distinction: we identify false positives using a distance threshold.

Specifically, our metrics are based on the pixel distance between the bottom centers of vehicles. We calculate the center distance between the detected vehicle and ground truth, denoted as d . We set a distance error tolerance of θ and consider detections with $d < \theta$ as true positives and those with $d \geq \theta$ as false positives. Detections are sorted in descending order of confidence scores for Average Precision (AP) calculation. We calculate AP for $\theta = 2, 5, 10, 15, 20, 50$ pixels, as well as the mean average precision (mAP). We report mAP, AP@20 (AP with $\theta = 20$ pixels), AP@50 (AP with $\theta = 50$ pixels), and the average recall (AR).

C. Training Settings

We follow the training pipeline provided by YOLOX [30] with some modifications to fit our dataset. We use YOLOX-Nano as the default model in our experiments. We train

the model for 150 epochs in total with 15 warm-up epochs included, and drop the learning rate by a factor of 10 after 100 epochs. The initial learning rate is set to be $4e - 5$ and the weight decay is set to be $5e - 4$. The Adam [48] optimizer is used. We train the model with a mini-batch size 8 on one NVIDIA RTX 3090 GPU.

For data augmentation, we first resize the input image such that the long side is at 640 pixels, and then pad the short side to 640 pixels. Random horizontal flips are applied with probability 0.5 and a random Hue, Saturation, and Value (HSV) augmentation is applied with a gain range of [5, 30, 30].

D. Experiments and Evaluation at Mcity

To thoroughly test the robustness of the proposed perception system, a total of six field-tests were carried out at Mcity during the months of July and August, 2022. During the field tests, vehicles adhered to traffic and lane rules while traversing the intersection for at least 15 minutes per trial. To ensure an adequate level of diversity, over 20 distinct vehicles were used in the experiment. These six trials encompassed a wide range of environmental variations including different weather (sunny, cloudy, light raining, heavy raining) and lighting (daytime and nighttime) conditions.

Two evaluation datasets were built from the field tests described above: normal condition evaluation dataset and harsh condition evaluation dataset. The normal condition dataset contains 217 images with real vehicles in the intersection during the daytime under good weather conditions. The harsh condition dataset contains 134 images with real vehicles in the intersection under adverse conditions. 15 images are under light raining conditions, 39 images are collected at twilight or dusk, 50 images are collected under heavy raining conditions, 30 images are collected in sunshine after raining conditions.

Training dataset	#images	Normal Condition Evaluation				Harsh Condition Evaluation			
		mAP	AP@20	AP@50	AR	mAP	AP@20	AP@50	AR
COCO [49]	118K	47.5	70.3	88.9	62.1	38.3	54.4	85.2	57.6
KITTI [50]	8K	46.4	76.2	89.5	62.8	33.6	54.7	75.8	53.6
BAAI-Vanhee [40]	2K	42.5	65.3	84.9	62.6	34.7	48.7	80.6	57.2
DAIR-V2X [51]	7K	39.7	60.1	71.6	60.1	34.1	51.0	62.4	54.3
Ours	4K	49.1	78.0	92.4	63.6	44.4	72.1	89.8	59.1

TABLE I: Comparison of model trained on our synthesis dataset to models on other existing datasets. The model trained on our synthesis dataset achieves the best performance on both normal conditions and harse conditions.

Settings				Normal Condition Evaluation				Harsh Condition Evaluation			
AR	AR + RE	Single bg.	Diverse bg.	mAP	AP@20	AP@50	AR	mAP	AP@20	AP@50	AR
✓		✓		34.8	63.0	84.8	54.4	29.9	50.1	77.7	49.8
✓			✓	40.1	66.1	88.5	57.9	37.0	62.7	85.8	53.9
✓	✓	✓		43.4	73.4	89.1	57.4	37.1	64.4	82.4	54.8
✓			✓	49.1	78.0	92.4	63.6	44.4	72.1	89.8	59.1

TABLE II: Ablation study. In the settings, **AR** means to directly use Augmented Reality to render vehicles. **AR+RE** means to use Augmented Reality with Reality Enhancement for vehicle generation. **Single bg.** means to use only one single background for dataset generation. **Diverse bg.** means to use diverse backgrounds for dataset generations.

We compare YOLOX-Nano trained on our synthesized data to the same model trained on other datasets, including the general object detection dataset COCO [49], the vehicle-side perception dataset KITTI [50], and the roadside perception datasets BAAI-Vanhee [40] and DAIR-V2X [51]. Since we evaluate the vehicle bottom center position, while the following datasets only provide the object bounding box in their 2d annotations, for models trained on COCO, KITTI, BAAI-Vanhee, and DAIR-V2X, we manually apply a center shift to roughly map the predicted vehicle center to vehicle bottom center by $x_{bottom} = x, y_{bottom} = y + 0.35h$. Here (x_{bottom}, y_{bottom}) is the estimated vehicle bottom center after mapping, and the (x, y) is the predicted object center by the detector.

Table I shows the comparison between the model trained on our dataset and on other datasets. We use the pretrained YOLOX weight as the initial weights of our training process, this is a standard practice for YOLOX detector fine-tuning. The model trained on our dataset outperforms all other datasets on both normal conditions and harsh conditions. On normal conditions, our model achieves 1.6 mAP improvement and 1.5 AR improvement over the second best model (trained on COCO). On harsh conditions, our model achieves 6.1 mAP improvement and 1.5 AR improvement over the model trained on COCO. One can observe that our method enhances performance under both normal and harsh conditions, with a more significant improvement noted in harsh conditions. This improvement is due to the lack of harsh condition data in the original training dataset, which our synthesized dataset compensates for. Notably, the performance enhancement is greater at a 20-pixel threshold than at a 50-pixel threshold. This indicates that our method effectively guides the detector to improve localization accuracy, a critical aspect in the context of cooperative driving.

For other datasets, one can see that the models trained on roadside perception datasets (BAAI-Vanhee and DAIR-V2X)

Training dataset	#images	Normal	Snow
Real data (human labeled)	1K	51.3	39.7
Mixed (real + synthesized)	1K + 1K	53.6	45.8

TABLE III: Comparison of model trained on 1K labeled data and model trained on the dataset of mixed labeled and synthesized data. mAP is reported.

Diversity of backgrounds		Normal		Harsh	
Weather diversity	Time diversity	mAP	AR	mAP	AR
✗	✗	43.4	57.4	37.1	54.8
✓	✗	46.8	54.7	40.1	57.3
✗	1 day, 8am to 8pm	47.4	60.3	41.8	56.8
✓	5 day, 8am to 8pm	49.1	63.6	44.4	59.1

TABLE IV: Ablation study on diversity of backgrounds. Adding weather diversity and time diversity both improve the detection performance on all conditions. Improvement on harsh conditions is more significant.

are worse than COCO and KITTI on normal conditions. This implies that the roadside perception datasets might have a weaker transfer-ability than general object detection datasets. One possible reason might be the poses of the camera are fixed. On harsh conditions, none of the existing datasets achieve satisfactory performance.

E. Experiments at State St/Ellsworth Rd Roundabout

As mentioned earlier, a roadside perception system has already been deployed [14] with training data collected and annotated in the fall. It has been found that the perception system degrades dramatically during winter, especially in snowy conditions. To mitigate this issue, we use the synthesizing pipeline introduced in this paper to enrich training samples with winter and fall background.

For evaluation, we collected 343 image data from the round-about site in normal weather conditions and snow weather conditions. For normal weather conditions, we collected and

Pretrain dataset	Normal		Harsh	
	mAP	AR	mAP	AR
—	43.7	63.8	34.7	60.6
KITTI [50]	48.2	66.2	41.6	60.4
BAAI-Vanhee [40]	42.4	59.0	40.6	56.3
DAIR-V2X [51]	44.8	61.7	40.3	58.1
COCO [49]	49.1	63.6	44.4	59.1

TABLE V: Ablation study on pretraining. Pretraining on existing datasets improves mAP on both normal conditions and harsh conditions. AR is not improved by pretraining.

annotated 111 images. For snow weather conditions, 232 images are contained for evaluation.

As shown in Table III, We compare the model trained with real data only (collected in the fall), and the model trained with mixed data (real data collected in fall, and synthesized data collected in both fall and winter). The model trained on mixed data outperforms the model trained on real data on both normal condition and snowy weather condition. Under snowy conditions, the improvement of 6.1 mAP is obtained. Figure 9 shows the comparison of the two models. Both detectors perform decently in the fall. Under snowy weather conditions, the model trained with mixed data performs better.

F. Ablation Study

Analysis on components of the pipeline. In our data synthesis analysis, we focus on two key components: GAN-based reality enhancement (RE) and diverse backgrounds. In Table II, we compare four settings: AR only with a single background, AR only with diverse backgrounds, AR + RE with a single background, and AR + RE with diverse backgrounds. The ‘diverse background’ refers to the shuffled selection from multiple backgrounds used in our training dataset, as mentioned in previous section. In contrast, the ‘single background’ dataset consistently uses the same background for synthesizing data. We observe that applying diverse backgrounds to AR-only data improves mAP by 5.3 in normal conditions and 7.1 in harsh conditions. Additionally, when compared to AR-only with a single background, adding RE leads to an improvement of 8.6 mAP in normal conditions and 7.3 mAP in harsh conditions. The combination of diverse backgrounds and reality enhancement further enhances performance by over 5 mAP in normal conditions and 7 mAP in harsh conditions.

Analysis on diversity of backgrounds. Using diverse backgrounds in image rendering is the key to achieve robust vehicle detection over different lighting conditions and weather conditions. Table IV shows the analysis on diversity of backgrounds. We can clearly observe that both weather diversity and time diversity improve the detection performance. An interesting finding is that the performance on normal conditions is also greatly improved by the diverse backgrounds.

Analysis on the Impact of Different Pretrained Weights. In Table V, we show that our method can also benefit from pre-training on existing datasets. On normal conditions, pretraining on COCO dataset or KITTI dataset improves the detection

performance by over 4 mAP, while pretraining on BAAI-Vanhee or DAIR-V2X dataset shows no significant improvement. One possible reason is that the BAAI-Vanhee dataset and DAIR-V2X dataset are roadside datasets captured in Chinese intersections. The generalization ability to U.S. intersections might be limited. On harsh conditions, pretraining on all datasets shows decent mAP improvement.

V. DISCUSSION AND FUTURE WORK

From section IV, it can be observed that the model’s performance is enhanced after tuning on our synthesized dataset, especially in terms of precision under harsh conditions. It is worth noting that the improvement in recall is relatively marginal in most cases. An intuitive explanation can be provided here. By incorporating a substantial number of background images into the training dataset, the model is able to rectify instances where it incorrectly identifies backgrounds as vehicles. However, in order to enhance recall, the model must address cases where it misclassifies vehicles as backgrounds. In our specific case, there still exists a disparity between the synthesized vehicles in our dataset and real-world vehicles. As part of future work, we intend to enhance the quality of realistic vehicle synthesis. One promising direction is the utilization of Stable Diffusion [52] to render realistic images and achieve style translation. To be efficient, this approach will require a substantially larger dataset. Such a dataset could be feasibly acquired post-deployment in the real world, as part of the Smart Intersection Project (SIP) [53]. Furthermore, We plan to extend the investigation on more practical factors such as traffic flow conditions, shadows, and lighting, enriching the understanding on how different factors influence the performance.

VI. CONCLUSION

In this study, we introduce a groundbreaking AR and GAN-based data synthesis pipeline, specifically designed to tackle the prevalent and crucial challenge of data insufficiency in current roadside vehicle perception systems. This innovative pipeline facilitates the training or fine-tuning of detectors, enabling them to seamlessly adapt to new locations, weather conditions, or lighting scenarios with minimal or virtually no human manual effort. This approach marks a significant leap forward in the real-world deployment of roadside detectors for autonomous driving, especially in the context of large-scale, robust deployment. A comprehensive evaluation is performed at different locations, under multiple weather and lighting conditions is reported in this paper. We demonstrate that our synthesized dataset can train a detector from scratch or fine-tune detectors trained from other datasets and improve the precision and recall under multiple lighting and weather conditions, yielding a much more robust perception system.

REFERENCES

- [1] S. Draft, “J2735 dedicated short range communications (dsrc) message set dictionary,” *Rev 0.7, Jan*, 2006.
- [2] J. B. Kenney, “Dedicated short-range communications (dsrc) standards in the united states,” *Proceedings of the IEEE*, vol. 99, no. 7, pp. 1162–1182, 2011.

- [3] S. A. A. T. Committee *et al.*, “V2x sensor-sharing for cooperative & automated driving,” *SAE J3224*. Available online: <https://www.sae.org/servlets/works/committeeHome.do>, 2019.
- [4] M. Billinghurst, A. Clark, G. Lee, *et al.*, “A survey of augmented reality,” *Foundations and Trends® in Human–Computer Interaction*, vol. 8, no. 2-3, pp. 73–272, 2015.
- [5] A. Creswell, T. White, V. Dumoulin, K. Arulkumaran, B. Sengupta, and A. A. Bharath, “Generative adversarial networks: An overview,” *IEEE signal processing magazine*, vol. 35, no. 1, pp. 53–65, 2018.
- [6] S. R. E. Datondji, Y. Dupuis, P. Subirats, and P. Vasseur, “A survey of vision-based traffic monitoring of road intersections,” *IEEE transactions on intelligent transportation systems*, vol. 17, no. 10, pp. 2681–2698, 2016.
- [7] T. Furuya and C. J. Taylor, “Road intersection monitoring from video with large perspective deformation,” Ph.D. dissertation, University of Pennsylvania, 2014.
- [8] S. Messelodi, C. M. Modena, and M. Zanin, “A computer vision system for the detection and classification of vehicles at urban road intersections,” *Pattern analysis and applications*, vol. 8, no. 1, pp. 17–31, 2005.
- [9] N. Saunier and T. Sayed, “A feature-based tracking algorithm for vehicles in intersections,” in *The 3rd Canadian Conference on Computer and Robot Vision (CRV'06)*. IEEE, 2006, pp. 59–59.
- [10] C. Li, A. Chiang, G. Dobler, Y. Wang, K. Xie, K. Ozbay, M. Ghahremani, J. Zhou, and D. Wang, “Robust vehicle tracking for urban traffic videos at intersections,” in *2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2016, pp. 207–213.
- [11] F. Faisal, S. K. Das, A. H. Siddique, M. Hasan, S. Sabrin, C. A. Hossain, and Z. Tong, “Automated traffic detection system based on image processing,” *Journal of Computer Science and Technology Studies*, vol. 2, no. 1, pp. 18–25, 2020.
- [12] A. Aboah, “A vision-based system for traffic anomaly detection using deep learning and decision trees,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 4207–4212.
- [13] R. Zhang, Z. Zou, S. Shen, and H. X. Liu, “Design, implementation, and evaluation of a roadside cooperative perception system,” *Transportation Research Record*, p. 03611981221092402, 2022.
- [14] Z. Zou, R. Zhang, S. Shen, G. Pandey, P. Chakravarty, A. Parchami, and H. X. Liu, “Real-time full-stack traffic scene perception for autonomous driving with roadside cameras,” in *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 890–896.
- [15] R. Zhang, D. Meng, S. Shen, Z. Zou, H. Li, and H. X. Liu, “Msight: An edge-cloud infrastructure-based perception system for connected automated vehicles,” 2023.
- [16] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, “Carla: An open urban driving simulator,” in *Conference on robot learning*. PMLR, 2017, pp. 1–16.
- [17] S. Shah, D. Dey, C. Lovett, and A. Kapoor, “Airsim: High-fidelity visual and physical simulation for autonomous vehicles,” in *Field and service robotics*. Springer, 2018, pp. 621–635.
- [18] G. Ros, L. Sellart, J. Materzynska, D. Vazquez, and A. M. Lopez, “The synthia dataset: A large collection of synthetic images for semantic segmentation of urban scenes,” in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016, pp. 3234–3243.
- [19] S. R. Richter, V. Vineet, S. Roth, and V. Koltun, “Playing for data: Ground truth from computer games,” in *European conference on computer vision*. Springer, 2016, pp. 102–118.
- [20] S. R. Richter, Z. Hayder, and V. Koltun, “Playing for benchmarks,” in *Proceedings of the IEEE International Conference on Computer Vision*, 2017, pp. 2213–2222.
- [21] V. Seib, B. Lange, and S. Wirtz, “Mixing real and synthetic data to enhance neural network training—a review of current approaches,” *arXiv preprint arXiv:2007.08781*, 2020.
- [22] J. Tremblay, A. Prakash, D. Acuna, M. Brophy, V. Jampani, C. Anil, T. To, E. Cameracci, S. Boochoon, and S. Birchfield, “Training deep networks with synthetic data: Bridging the reality gap by domain randomization,” in *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, 2018, pp. 969–977.
- [23] L. Perez and J. Wang, “The effectiveness of data augmentation in image classification using deep learning,” *arXiv preprint arXiv:1712.04621*, 2017.
- [24] Z. Yang, Y. Chai, D. Anguelov, Y. Zhou, P. Sun, D. Erhan, S. Rafferty, and H. Kretzschmar, “Surfelgan: Synthesizing realistic sensor data for autonomous driving,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11118–11127.
- [25] T. Rothmeier and W. Huber, “Let it snow: On the synthesis of adverse weather image data,” in *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, 2021, pp. 3300–3306.
- [26] F. Zhan, S. Lu, C. Zhang, F. Ma, and X. Xie, “Adversarial image composition with auxiliary illumination,” in *Proceedings of the Asian Conference on Computer Vision*, 2020.
- [27] Y. Chen, F. Rong, S. Duggal, S. Wang, X. Yan, S. Manivasagam, S. Xue, E. Yumer, and R. Urtasun, “Geosim: Realistic video simulation via geometry-aware composition for self-driving,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2021, pp. 7230–7240.
- [28] J. Wang, S. Manivasagam, Y. Chen, Z. Yang, I. A. Bărsan, A. J. Yang, W.-C. Ma, and R. Urtasun, “Cadsim: Robust and scalable in-the-wild 3d reconstruction for controllable sensor simulation,” in *6th Annual Conference on Robot Learning*, 2022.
- [29] MCity, “Home: MCity,” <https://mcity.umich.edu/>, 2022.
- [30] Z. Ge, S. Liu, F. Wang, Z. Li, and J. Sun, “YOLOX: exceeding YOLO series in 2021,” *CoRR*, vol. abs/2107.08430, 2021. [Online]. Available: <https://arxiv.org/abs/2107.08430>
- [31] R. C. Gonzalez, *Digital image processing*. Pearson education india, 2009.
- [32] P. A. Lopez, M. Behrisch, L. Bieker-Walz, J. Erdmann, Y.-P. Flötteröd, R. Hilbrich, L. Lücken, J. Rummel, P. Wagner, and E. Wießner, “Microscopic traffic simulation using sumo,” in *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018. [Online]. Available: <https://elib.dlr.de/124092/>
- [33] OpenStreetMap contributors, “Planet dump retrieved from https://planet.osm.org ,” <https://www.openstreetmap.org> , 2017.
- [34] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, *et al.*, “Shapenet: An information-rich 3d model repository,” *arXiv preprint arXiv:1512.03012*, 2015.
- [35] E. Marchand, H. Uchiyama, and F. Spindler, “Pose estimation for augmented reality: a hands-on survey,” *IEEE transactions on visualization and computer graphics*, vol. 22, no. 12, pp. 2633–2651, 2015.
- [36] “User interface for map-to-pixel calibration,” 2019, accessed: 12/30/2023. [Online]. Available: <https://map2picturecalibration.net/>
- [37] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.
- [38] M. Matl, “Pyrender,” <https://github.com/mmatl/pyrender>, 2019.
- [39] T. Park, A. A. Efros, R. Zhang, and J.-Y. Zhu, “Contrastive learning for unpaired image-to-image translation,” in *European conference on computer vision*. Springer, 2020, pp. 319–345.
- [40] Y. Deng, D. Wang, G. Cao, B. Ma, X. Guan, Y. Wang, J. Liu, Y. Fang, and J. Li, “BAAI-VANJEE roadside dataset: Towards the connected automated vehicle highway technologies in challenging environments of china,” *CoRR*, vol. abs/2105.14370, 2021. [Online]. Available: <https://arxiv.org/abs/2105.14370>
- [41] M. S. Lee, W. Shin, and S. W. Han, “Tracer: Extreme attention guided salient object tracing network (student abstract),” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 11, 2022, pp. 12993–12994.
- [42] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 779–788.
- [43] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “Ssd: Single shot multibox detector,” in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11–14, 2016, Proceedings, Part I 14*. Springer, 2016, pp. 21–37.
- [44] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” *Advances in neural information processing systems*, vol. 28, 2015.
- [45] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” *International journal of computer vision*, vol. 88, pp. 303–338, 2010.
- [46] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liou, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, “nuscenes: A multimodal dataset for autonomous driving,” in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13–19, 2020*. Computer Vision Foundation / IEEE, 2020, pp. 11618–11628.
- [47] K. Bernardin and R. Stiefelhagen, “Evaluating multiple object tracking performance: the clear mot metrics,” *EURASIP Journal on Image and Video Processing*, vol. 2008, pp. 1–10, 2008.

- [48] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, Y. Bengio and Y. LeCun, Eds., 2015. [Online]. Available: <http://arxiv.org/abs/1412.6980>
- [49] T. Lin, M. Maire, S. J. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft COCO: common objects in context," in *Computer Vision - ECCV 2014 - 13th European Conference, Zurich, Switzerland, September 6-12, 2014, Proceedings, Part V*, ser. Lecture Notes in Computer Science, D. J. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds., vol. 8693. Springer, 2014, pp. 740–755.
- [50] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.
- [51] H. Yu, Y. Luo, M. Shu, Y. Huo, Z. Yang, Y. Shi, Z. Guo, H. Li, X. Hu, J. Yuan, and Z. Nie, "DAIR-V2X: A large-scale dataset for vehicle-infrastructure cooperative 3d object detection," *CoRR*, vol. abs/2204.05575, 2022. [Online]. Available: <https://doi.org/10.48550/arXiv.2204.05575>
- [52] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 10684–10695.
- [53] "Smart Intersection Project," <https://sip.umtri.umich.edu/>, 2022.

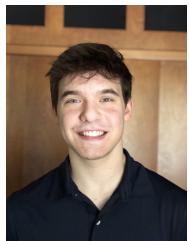
VII. BIOGRAPHY



Rusheng Zhang received the B.E. degree in micro electrical mechanical system and second B.E. degree in Applied Mathematics from Tsinghua University, Beijing, in 2013. He received an M.S. and PhD degree in electrical and computer engineering from Carnegie Mellon University, in 2015, 2019 respectively. His research areas include artificial intelligence, cooperative driving, cloud computing and vehicular networks.



Depu Meng (Member, IEEE) is a Post Doctoral Research Fellow at the Department of Civil and Environmental Engineering, University of Michigan. He received his B. E. degree from the Department of Electrical Engineering and Information Science at the University of Science and Technology of China in 2018. He received his Ph. D. degree from the Department of Automation at the University of Science and Technology of China. His research interests include computer vision and autonomous driving systems.



Lance Bassett received a B.S. and M.S.E in Computer Science from the University of Michigan, Ann Arbor in 2023. He did research in computer vision with the Michigan Traffic Lab as a graduate student, and currently works on real-time software and media in the automotive industry.

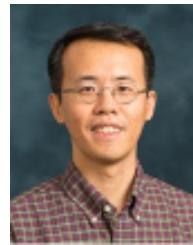


Shengyin (Sean) Shen works as a Research Engineer in the Engineering Systems Group at the University of Michigan Transportation Research Institute (UMTRI). Sean holds an MS degree in Civil and Environmental Engineering from the University of Michigan, Ann Arbor, and an MS degree in Electrical Engineering from the University of Bristol, UK. He also earned a BS degree from Beijing University of Posts and Telecommunications, China. Sean's research interests are primarily focused on cooperative driving automation and related applications that use

roadside perception, edge-cloud computing, and V2X communications to accelerate the deployment of automated vehicles. He has extensive experience in implementation of large-scale deployments, such as the Safety Pilot Model Deployment (SPMD), Ann Arbor Connected Vehicle Testing Environment (AACVTE), and Smart Intersection Project. Moreover, he has been involved in many research projects funded by public agencies such as USDOT, USDOE, and companies such as Crash Avoidance Metric Partnership (CAMP), Ford Motor Company, and GM Company, among others.



Zhengxia Zou Zhengxia Zou received his B.S. and Ph.D. degrees from the Image Processing Center, School of Astronautics, Beihang University, Beijing, China, in 2013 and 2018, respectively. He is currently a Professor at the School of Astronautics, Beihang University. From 2018 to 2021, he worked at the University of Michigan, Ann Arbor as a Post-Doctoral Research Fellow. His research interests include computer vision and related problems in autonomous driving and remote sensing. He has published more than 40 peer-reviewed papers in tier journals and conferences, including Nature, Nature Communications, PROCEEDINGS OF THE IEEE, IEEE TRANSACTIONS ON IMAGE PROCESSING, IEEE TRANSACTIONS ON GEOSCIENCE AND REMOTE SENSING, and IEEE/CVF Computer Vision and Pattern Recognition. Dr. Zou was selected as "World's Top 2% Scientists" by Stanford University in 2022.



Henry X. Liu (Member, IEEE) received the bachelor's degree in automotive engineering from Tsinghua University, China, in 1993, and the PhD. degree in civil and environment engineering from the University of Wisconsin-Madison in 2000. He is currently a professor in the Department of Civil and Environmental Engineering and the Director of Mcity at the University of Michigan, Ann Arbor. He is also a Research Professor at the University of Michigan Transportation Research Institute and the Director for the Center for Connected and Automated Transportation (USDOT Region 5 University Transportation Center). From August 2017 to August 2019, Prof. Liu served as DiDi Fellow and Chief Scientist on Smart Transportation for DiDi Global, Inc., one of the leading mobility service providers in the world. Prof. Liu conducts interdisciplinary research at the interface of transportation engineering, automotive engineering, and artificial intelligence. Specifically, his scholarly interests concern traffic flow monitoring, modeling, and control, as well as testing and evaluation of connected and automated vehicles. Prof. Liu is the managing editor of Journal of Intelligent Transportation Systems.