2022 IEEE International Conference on Robotics and Automation (ICRA)
May 23-27, 2022. Philadelphia, PA, USA

Real-time Full-stack Traffic Scene Perception for Autonomous Driving with Roadside Cameras

Zhengxia Zou[1], Rusheng Zhang[1], Shengyin Shen[1], Gaurav Pandey [2], Punarjay Chakravarty[2], Armin Parchami[2], and Henry X. Liu[1*]

***Abstract*— We propose a novel and pragmatic framework for traffic scene perception with roadside cameras. The proposed framework covers a full-stack of roadside perception pipeline for infrastructure-assisted autonomous driving, including object detection, object localization, object tracking, and multi-camera information fusion. Unlike previous vision-based perception frameworks rely upon depth offset or 3D annotation at training, we adopt a modular decoupling design and introduce a landmark-based 3D localization method, where the detection and localization can be well decoupled so that the model can be easily trained based on only 2D annotations. The proposed framework applies to either optical or thermal cameras with pinhole or fish-eye lenses. Our framework is deployed at a two-lane roundabout located at Ellsworth Rd. and State St., Ann Arbor, MI, USA, providing 7x24 real-time traffic flow monitoring and high-precision vehicle trajectory extraction. The whole system runs efficiently on a low-power edge computing device with all-component end-to-end delay of less than 20ms.**

I. INTRODUCTION

Infrastructure-assisted cooperative perception is an emerging research topic in autonomous driving and intelligent transportation. Recently, the rapid development of deep learning and computer vision technology has opened up new perspectives for assisting automated vehicles in complex driving environments. With roadside sensors, hazardous driving scenarios could be identified (e.g. objects hidden in the blind spot), and automated vehicles could be informed in advance.

In this paper, we propose a novel and pragmatic solution for roadside camera-based perception. As shown in Fig. 1, the proposed scheme covers a full-stack of roadside perception pipeline for infrastructure-assisted autonomous driving - from object detection, localization, tracking, to multi-sensor information fusion. To obtain the real-world object location from a 2D image, previous 3D detection methods [1], [2], [3] typically require camera calibration parameters or depth offset available at training so that a transformation between the image plane and the 3D world can be constructed. However, such information is difficult to obtain in data annotation phase. Particularly, the calibration of camera extrinsic parameters may rely heavily on other types of sensors (such as lidar) and may also involve the issues of joint calibration and multi-sensor synchronization [4].

[1]Zhengxia Zou, Rusheng Zhang, Shengyin Shen, and Henry X. Liu are with the Department of Civil and Environmental Engineering, University of Michigan, Ann Arbor.

[2]Gaurav Pandey, Punarjay Chakravarty, and Armin Parchami are with Ford Motor Company.

*Corresponding author, henryliu@umich.edu.

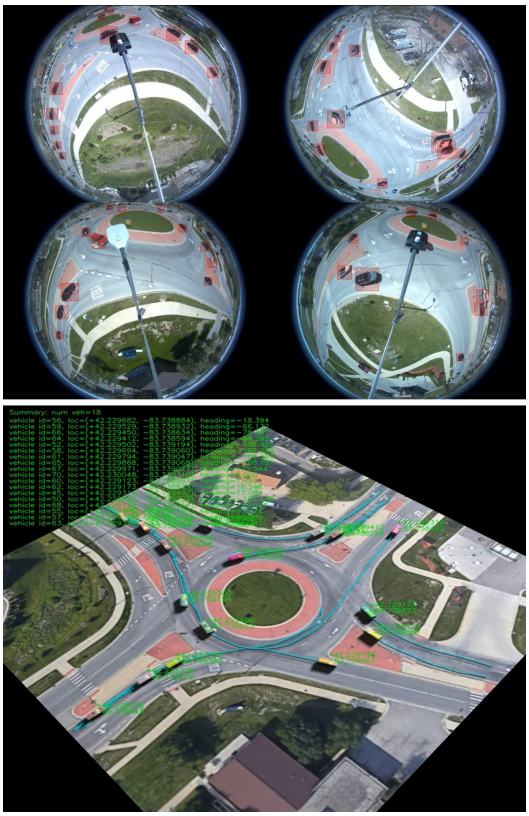

Fig. 1: We propose a vision-based solution for real-time traffic object detection, localization, information fusion, and tracking. The upper image shows real-time detection results with four 360° roadside fish-eye cameras and the lower one shows the vectorized location and trajectory of each object.

Instead of using multi-sensor joint calibration, we introduce a purely vision-based solution with a detection-localization decoupling design. In our method, a landmark-based object localization strategy is utilized that allows our detector to be trained solely based on 2D annotations. The detection results are then lifted to 3D with the landmark Homography and camera intrinsics. Our method can be applied to both optical and thermal cameras with pinhole or fish-eye lenses. Using a lightweight MobileNet-v2 [5] network backbone, our method can run efficiently in real-time on a low-power edge computing box. The all-component end-to-end perception delay is less than 20ms.

Our contributions are summarized as follows.

- We propose a novel framework for full-stack road-

2022 IEEE International Conference on Robotics and Automation (ICRA) | 978-1-7281-9681-7/22/$31.00 ©2022 IEEE | DOI: 10.1109/ICRA46639.2022.9812137

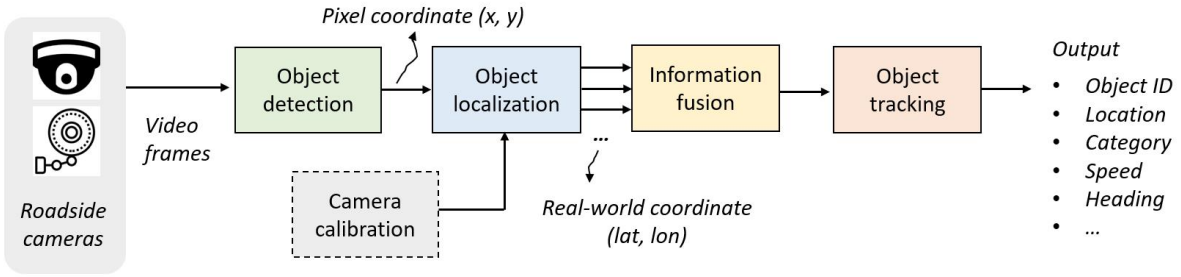

Fig. 2: An overview of the proposed framework for roadside vision-based traffic scene perception.

side assisted traffic scene perception, including object detection, 3D localization, tracking, and multi-camera information fusion. Our method is flexible and scalable - since the training of our model only requires 2D annotations, the whole framework can be deployed quickly and migrated elegantly at any new application scenarios.

- Most previous perception systems for autonomous driving focus on onboard perception only and rarely discuss roadside-based solutions. To our best knowledge, we are one of the first to propose and implement a fully established roadside framework for infrastructure-assisted autonomous driving.
- Our framework is deployed at a two-lane roundabout in Ann Arbor, MI, providing 7x24 traffic flow monitoring and hazardous driving warnings capabilities. For the entire 10000 m^2 roundabout area, our method achieves sub-meter-level localization accuracy with a single camera and 0.4m localization accuracy with information fusion of multiple cameras.

II. RELATED WORK

Roadside sensor-based perception system has a long history and can be traced back to 1980s [6]. To detect traffic objects and monitor their behavior, some early methods are developed based on traditional computer vision techniques such as background subtraction [7], frame difference [8], optical flow [9], etc. Recently, the fast development of deep learning technology has greatly promoted object detection and tracking research. Some representative approaches includes Faster R-CNN [10], [11], [12], SSD [13], and YOLO [14], [15], [16] for object detection; DeepSort [17] and Center Track [18] for object tracking. Some of these methods have been successfully applied to UAV-based traffic surveillance applications [19]. However, for roadside-based traffic perception, deep learning-based approaches are still in their infancy and have attracted increasing attention recently [20].

2D/3D object detection plays a central role in roadside traffic scene perception. The task of 2D object detection [12] is to find the pixel location of all objects of interest in the image and determine their bounding boxes and categories. In contrast to conventional 2D object detection, 3D object detection predicts 3D boxes (with 3D location, orientation, and size) from a single monocular image [21], [3], [1], [2] or stereo images [22], which has received great attention in autonomous driving recently. The proposed detection method is mostly related to Objects as Points [23], a recent popular 2D detection framework. We use a similar idea of point detection but extend this framework for 3D pose and 3D size estimation with additional output branches. Instead of predicting the center of 2D box, we predict the object's 3D bottom center and lift the prediction to 3D using a pre-calibrated plane-to-plane Homography. Compared to recent 3D object detection methods, our "point detection + 3D lifting" design makes our method neither requires depth information nor 3D annotation during the training, greatly reducing the cost of data annotation and collection. In addition, most current 3D object detection solutions of autonomous driving only focus on onboard perception and rarely discuss roadside-based perception. In contrast to previous onboard solutions [21], [3], [1], [2], we provide a new framework for roadside-based perception and have evaluated the effectiveness of our system at a two-lane roundabout with real-world connected and automated vehicles.

III. METHODOLOGY

The introduced framework is composed of four different modules: 1. object detection, 2. object localization, 3. information fusion, and 4. object tracking. Fig. 2 shows an overview of the proposed framework. The object detection operates directly on 2D images and generates 2D bounding boxes; the object localization lifts the 2D detection to the 3D world; detections from different sensors are fused; finally, individual ids will be assigned for all detected vehicles with tracking.

A. Object Detection

A single-stage center-aware detector is designed for joint object detection, pose estimation, and category recognition. As shown in Fig. 3, the proposed detector consists of a lightweight image encoder E, a feature decoder D, and four prediction heads (for bottom center prediction, box-size estimation, pose-estimation, and vehicle type recognition, respectively). To improve detection on small objects, we apply feature pyramid fusion [24] in our decoder and progressively upsample the feature map to the same spatial size as the input. In the following, we will introduce the four prediction heads accordingly.

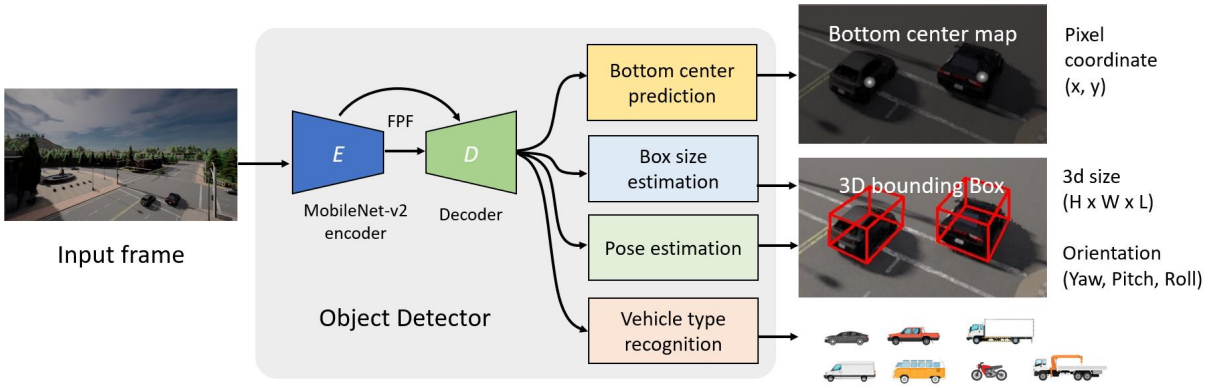

Fig. 3: Architecture of the proposed detection method. Our detector consists of a feature encoder [5], a feature decoder, and four output heads designed for vehicle bottom-center prediction, box-size estimation, pose estimation, vehicle type recognition.

1) Bottom-center prediction: The bottom-center prediction branch is trained to produce a heat-map with the same spatial size as the input. We define the loss function of the bottom-center prediction branch as a pixel-wise least-square loss between the prediction and ground truth:

$$\mathcal{L}_{center}(X) = \mathbb{E}_{X\sim\mathcal{D}}\{\text{TopK}(\|Y_{center} - \hat{Y}_{center}\|_2^2)\}, \quad (1)$$

where Y_{center} and $\hat{Y}_{center}$ are the prediction output and its 2D ground truth map. X and $\mathcal{D}$ are the input image and the dataset. TopK represents hard-example selection - in each training iteration, only the top 1% of pixels with the largest loss will be used for error back-propagation. In $\hat{Y}_{center}$, a larger pixel value means a larger probability the pixel belongs to the bottom center of an object. We generate the ground truth maps with a Gaussian function:

$$\hat{Y}_{center}(i,j) = \sum_{t}^{T} \exp\left(-d_t(i,j)^2/\sigma_t^2\right), \quad (2)$$

where (i,j) is the pixel location; T is the number of object in an image; $d_t(i,j)$ is the distance between the (i,j) to the bottom center of the t-th object; $\sigma_t = \frac{1}{2}\sqrt{l_t}$; l_t is the pixel bounding box diagonal length of the t-th object.

2) 3D Size and Pose Estimation: The 3D size prediction and pose estimation can be formulated as least square regression problems. The loss function of the 3D size branch and pose estimation branch are defined as follows:

$$\begin{aligned}\mathcal{L}_{size}(X) &= \mathbb{E}_{X\sim\mathcal{D}}\{\hat{Y}_{center}(\|\log Y_{size} - \log \hat{Y}_{size}\|_2^2)\},\\ \mathcal{L}_{pose}(X) &= \mathbb{E}_{X\sim\mathcal{D}}\{\hat{Y}_{center}(\|Y_{pose} - \hat{Y}_{pose}\|_2^2)\},\end{aligned} \quad (3)$$

where Y_{pose} and Y_{size} are the predicted pose and size maps. We apply log normalization to the predicted size for better convergence. $\hat{Y}_{pose}$ and $\log \hat{Y}_{size}$ are their ground truth. We use the ground truth bottom center $\hat{Y}_{center}$ as a pixel-wise weight map since the predictions only need to be focused on the object regions.

3) Object Category Recognition: The vehicle type recognition can be considered as a standard classification problem. We therefore define the loss as a standard cross-entropy distance between the probabilistic output and the ground truth. The loss function is defined as follows:

$$\mathcal{L}_{v-type}(X) = \mathbb{E}_{X\sim\mathcal{D}}\{-\hat{Y}_{center}\sum_{i}^{C}\hat{Y}_{type}^{(i)}\log Y_{type}^{(i)}\}, \quad (4)$$

where Y_{type} is the predicted category probability maps after softmax normalization; $\hat{Y}_{type}$ is the one-hot ground truth; C is the number of vehicle category.

4) Multi-task Loss: We finally train our detector by following multi-task loss function as follows:

$$\mathcal{L} = \mathcal{L}_{center} + \beta_1\mathcal{L}_{size} + \beta_2\mathcal{L}_{pose} + \beta_3\mathcal{L}_{v-type} \quad (5)$$

where β_1, β_2, and β_3 are predefined weights for balancing the loss terms from different prediction heads. Since all output branches are differentiable, we can train the whole detector in an end-to-end fashion.

5) Network configuration: We use a similar network configuration in all output branches. In each output, we use a stacked two convolutional layers on top of the decoder feature map for prediction. We choose Sigmoid output activation for bottom center prediction, Tanh for normalized pose prediction, ReLU for size prediction, and Softmax for category recognition.

B. Camera Calibration and Object Localization

Since our object detector is only trained with 2D annotations, to determine their real-world location, a mapping needs to be constructed between the pixel space and the 3D world. Here we introduce a simple and elegant solution for camera calibration and object localization. Instead of estimating the intrinsic/extrinsic camera matrices jointly with other sensors, we directly transform the image into a bird-eye view with an estimated Homography. In this way, the transformed view will have a uniform pixel resolution for the real-world longitude and latitude coordinate.

The area for perception is represented by a piece-wise segmented planar surface. We manually select a set of ground landmarks (e.g., pavement or roadside static objects) and annotate their pixel coordinate as well as real-world coordinate with Google Maps. For each segment, an Homography matrix $\boldsymbol{H}$ can be easily estimated with least square regression and RANSAC consensus between the two groups of landmark sets. A longitude mask M_{lon} and a latitude mask M_{lat} thus can be generated by projecting each pixel of the camera view to the real-world coordinate. Given the pixel location of any detected objects, their localization can be easily retrieved from lookup tables:

$$(x, y) = (M_{lon}^{(1,..,P)}(i, j), M_{lat}^{(1,..,P)}(i, j)), \tag{6}$$

where (i, j) is the bottom center pixel coordinate of an object and (x, y) is the estimated longitude and latitude value. P is the number of segmented planers.

The proposed solution also applies to fish-eye cameras. We assume the camera lens follow a generic radially symmetric model [25] $r(\theta) = k_1\theta + k_2\theta^3 + k_3\theta^5 + \ldots$. With the landmark pairs, the camera intrinsic matrix $\boldsymbol{K}$ and the distortion coefficients d_i can be estimated [26]. Then, by back-transforming the landmark points to an undistorted camera view, the Homography $\boldsymbol{H}^{(1,..P)}$ and the longitude/latitude masks can be generated in a way similar to pinhole cameras.

C. Object Tracking and Information Fusion

The object tracker is built on top of SORT (Simple Online and Realtime Tracking) [27], a popular online object tracking method. The basic idea is using a Kalman Filter [28] and the Hungarian Algorithm [29] for object state prediction and box matching. Instead of using pixel coordinates, we found using the world coordinate can better deal with camera distortions, especially when tested on fisheye cameras. The state of the Kalman Filter is defined as follows:

$$\boldsymbol{x} = [x_c, y_c, s, r, v_x, v_y, v_s, v_r]^T, \tag{7}$$

where (x_c, y_c) are the location of the object; s and r are the area and aspect-ratio of the bounding box; v_x, v_y, v_s v_r are the derivatives of x_c, y_c, s, and r. We set the maximum age of any consecutive un-detected objects to 3.

To fuse the detections from multiple cameras, we divide the map into several regions according to the camera location. The fusion is performed before the tracking, with only those high-certainty detection of each camera being used. Since the tracking is only performed based on the 3D locations, the proposed fusion design makes our system capable of tracking cross-camera moving objects with consistent identities.

IV. Experimental Results and Analysis

A. Experimental Setup

We evaluate our method in both simulation and real-world traffic environments:

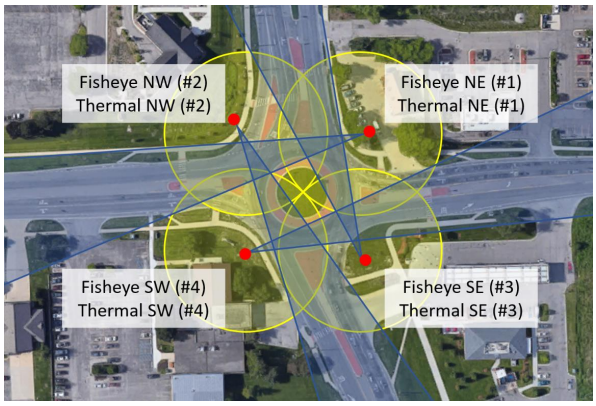

Fig. 4: Our real-world test environment: placement of four 360° fisheye cameras and four thermal cameras at State St. and W. Ellsworth Rd roundabout, Ann Arbor, MI.

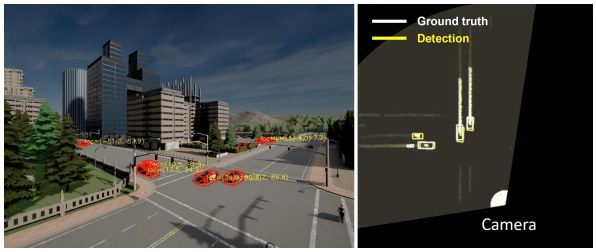

Fig. 5: Visualization of object detection, 3D localization, and tracking with a virtual camera placed at CARLA "Town 05".

1) Simulation Environment: We generate our synthetic dataset with CARLA Simulator [30]. We place four cameras at four corners of an intersection of CARLA "Town 05". For each camera, 16 video clips are collected, with $4 \times 16 \times 1000$ frames in total. Video #1 - #15 are used for training and video #16 is used for evaluation. We randomly generate 100 vehicles in each video clip. 3D bounding boxes of vehicles in both pixel coordinate and real-world coordinate are recorded. The clock rate is set to 2fps for training and 10fps for testing.

2) Real-world Environment: We evaluate our framework at a roundabout located at the intersection of W Ellsworth Rd and State St. in Ann Arbor, MI, with two groups of cameras - four 360 degree fisheye cameras and four long-range thermal cameras. The cameras are placed at the four corners of the roundabout. For each camera, we annotated 1000 images, with 90% for training and 10% for testing. The bottom rectangle of each vehicle is annotated. The annotation of all images took 400 man-hours in total. Fig. 4 shows the placement of the cameras.

3) Training Details: We use MobileNet-v2 [5] as the backbone of our detector. The detector is trained for 100 epochs using the Adam optimizer with batch_size=16 and learning_rate=0.0005. We set β_1=β_2=β_3=0.01. When training on the roundabout data, we ignore the vehicle height and predict 2D boxes in pixel size since we do not have their 3D ground truth. Training data augmentation is performed with random image clipping, random gamma correction, and random jittering. The image color is removed at the input of

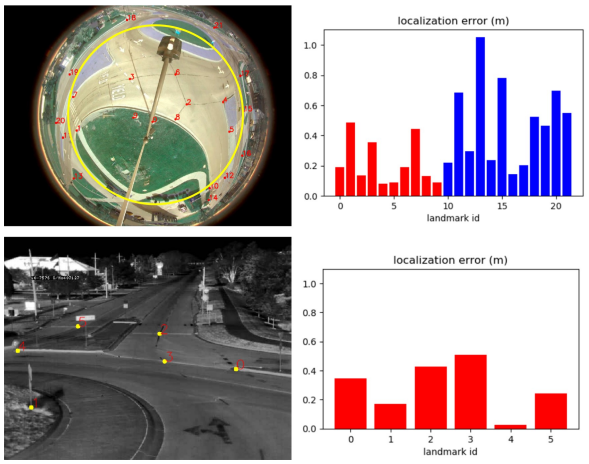

Fig. 6: Landmarks and their localization error (m). The red bars correspond to those in-ROI landmarks.

Fig. 7: Visualization of detection, localization, and tracking results with roundabout long-range thermal cameras.

the detector for better adapting to seasonal changes.

B. Bottom-center-aware Detection

The accuracy of the detector is evaluated on both synthetic images and real images. We follow VOC07 [31] detection metrics and calculate the mean average precision on different datasets. The VOC box-iou threshold is set to 0.5. Other thresholds are not reported here as bounding-box localization is not the focus of this paper. In IV-C, we will conduct a more detailed evaluation of 3D localization accuracy.

In Table I and Table II, we show box-level detection accuracy (VOC avg precision) and pose/3d-size error, respectively. In Fig. 5, Fig. 1, and Fig. 7, we show the detection + localization result with CARLA images, fish-eye images, and thermal images. When calculating the pose and size error, we only take into account those successful detections. Since we do not have the ground-truth of vehicle pose/size from real-world images, we only evaluate this part in CARLA simulation. An ablation study is also conducted where we remove the TopK selection in Eq. 1, and replace the bottom center prediction with 2D box center prediction. The top rows of Table I and Table II shows the ablation results. Observe when removing the Top-K selection, the training fails to converge. Also, replacing the bottom center prediction with a conventional 2D center prediction caused a noticeable decrease in the localization accuracy.

	CARLA	AA-Fisheye	AA-Thermal
w/o TopK select loss	— fail to converge —		
full implementation	0.880	0.912	0.886

TABLE I: 2D box-level accuracy (AP) of our detector on different sensors. The results are directly from the detection of individual images. No cross-camera fusion is adopted.

	Loc Err	Yaw Err	3D Size Err
w/o TopK select loss	— fail to converge —		
w/o bottom-center pred	3.210 (m)	9.512°	0.448 (m)
full implementation	0.984 (m)	9.510°	0.451 (m)

TABLE II: Pose and 3D size estimation error of our method with different configurations on the CARLA dataset.

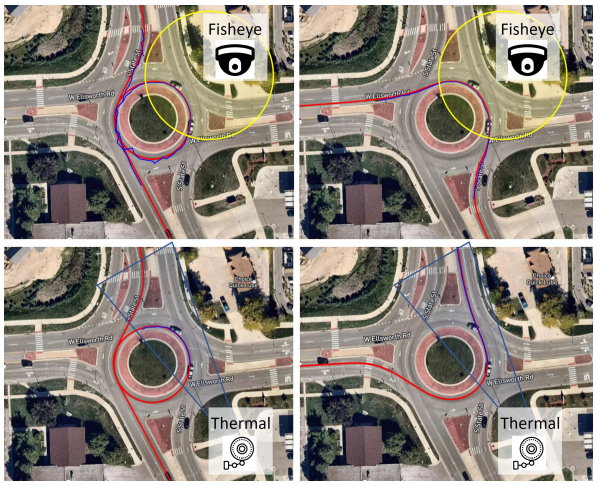

Fig. 8: Detected Lincoln MKZ trajectories (blue curve) and the trajectories recorded by RTK as ground truth (red curve). The above two images show the results from fish-eye cam #1 and the bottom two are from thermal cam #3. See Fig. 4 for the camera placement.

C. Localization

In this experiment, the calibration and end-to-end localization error are evaluated for both fisheye and thermal images.

1) Calibration Error Analysis: Every camera equipped at the roundabout is calibrated manually with 5-20 landmarks labeled on Google Maps. We set the number of segmented planer to one for pinhole camera and four for fisheye camera. We compare the longitude/latitude lookup values at the landmark locations with their ground truth. Fig. 6 shows the landmark distribution and their localization errors. Since we mainly care about the area underneath the camera (distant area can be covered by other cameras), we divide the map region into two groups: “region of interest (in-ROI)” and “ out of the region of interest (out-ROI)”. For a fish-eye camera, we define its ROI as a circular area centered at the camera location with a radius of 25 meters while for a long-range thermal camera, we define its ROI as the <200m area within its field of view. Fig. 6 shows the calibration error. For fisheye cameras, the average in-ROI error (within the yellow circle, marked as red in the bar-plot) is 0.219 $\pm$ 0.145 m.

	Fisheye		Thermal
	In-ROI (m)	Out-ROI (m)	In-ROI (m)
Trip #1	0.478 ± 0.248	1.038 ± 0.965	0.615 ± 0.340
Trip #2	0.377 ± 0.218	0.779 ± 0.747	0.339 ± 0.251
Trip #3	0.408 ± 0.167	1.154 ± 1.138	1.334 ± 0.792
Trip #4	0.217 ± 0.162	0.596 ± 0.559	0.373 ± 0.363
Trip #5	0.368 ± 0.195	0.969 ± 0.849	0.860 ± 0.622
Trip #6	0.401 ± 0.210	1.491 ± 2.031	0.837 ± 0.456
All	0.377 ± 0.207	0.964 ± 1.085	0.820 ± 0.546

TABLE III: Trajectory error between the detection and the ground truth (RTK) with fisheye and thermal cameras.

Fig. 9: (a) The edge device where our method is deployed and tested. (b) Our testing platform - a Hybrid Lincoln MKZ equipped with a high-precision RTK [32], [33].

The out-ROI error (marked as blue in the bar-plot) is 0.489 ± 0.268 m. For thermal cameras, the error is 0.288 ± 0.162 m.

2) Evaluation with Connected Vehicle: We deploy our system on an edge device (Jetson AGX XAVIER) at the City of Ann Arbor and provide 7x24 monitoring service of the roundabout traffic. A connected automated vehicle[1] — a Hybrid Lincoln MKZ [32], [33] equipped with a high-precision RTK and an Inertial Measurement Unit (IMU), is used to test our system. With the vehicle and sensors, we can measure the vehicle location in real-time. The vehicle and the edge device are shown in Fig. 9.

The vehicle is driven through the roundabout six times in two separate days: June 30th, 2021, and July 19th, 2021, recording the trajectories by RTK GPS as the ground truth. Fig. 8 shows the detected trajectories alongside with the ground-truth. Table III shows the localization error. For each trip, the error is calculated as the average project distance between the localization points and the ground truth trajectory. The average In-ROI error over 6 trials for fish-eye and thermal cameras are 0.377 m and 0.820 m respectively. Fig. 10 shows the localization error within the entire roundabout area before and after the fusion of all four fisheye cameras. With fusion, the average localization error is reduced from 0.834 m ± 1.037 m to 0.377 m ± 0.207 m. The fusion can therefore greatly improve both the localization accuracy and stability. Note that the large variance of the 6th trip error is caused by the camera shake in the wind. Nevertheless, we choose to report this non-ideal trip and include it in the performance analysis to give an end-to-end accuracy considering all practical issues.

[1]https://mcity.umich.edu/

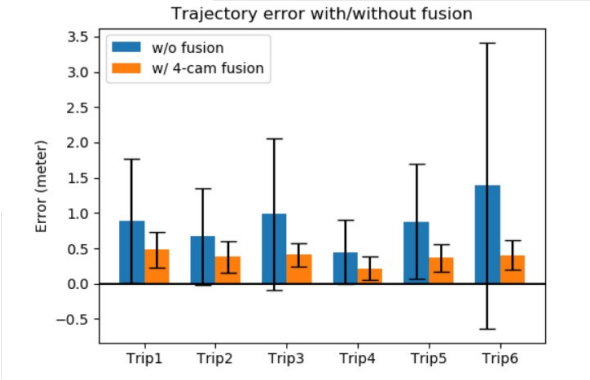

Fig. 10: Trajectory error with or without the fusion of the four fisheye cameras.

	1-way processing	4-way processing
I7-9700K + 2070S	160 fps / 6 ms	60×4 fps / 16 ms
I7-8750H + 1050Ti	60 fps / 16 ms	16×4 fps / 60 ms
Jetson AGX XAVIER	50 fps / 20 ms	18×4 fps / 55 ms

TABLE IV: The speed performance of (frames per second and delay) of the proposed framework on different devices.

D. Speed performance

We test the inference speed of our framework on multiple platforms with different computational capabilities. Table IV shows the detailed speed performance of our system. With half-precision inference speedup, the whole processing pipeline of our system (detection + localization + fusion + tracking) achieves 160fps on an I7-9700K+2070S desktop and 50fps on a Jetson AGX XAVIER edge device. When handling 4-way input video streams simultaneously, our system still achieves real-time processing speed, with 60×4 fps and 18×4 fps on the two platforms respectively.

V. CONCLUSIONS

We propose a vision-based traffic scene perception framework with object detection, localization, tracking, and sensor fusion. Owing to the decoupling design, the framework can be trained solely based on 2D annotations, which greatly overcomes difficulties in field deployment and migration. We tested our system with both real-world connected and automated vehicles and simulation environment, and achieve 0.4-meter localization accuracy within an entire 100x100 m^2 two-lane roundabout area. The all-components end-to-end perception delay is less than 20ms. The proposed method provides a novel solution for practical roadside perception and shows great potential in the cooperative perception of automated vehicles with infrastructure support.

ACKNOWLEDGMENT

This work was partially supported by Mcity and the College of Engineering of the University of Michigan and the Ford Motor Company.

REFERENCES

[1] Z. Liu, Z. Wu, and R. Tóth, "Smoke: Single-stage monocular 3d object detection via keypoint estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 996–997.

[2] F. Manhardt, W. Kehl, and A. Gaidon, "Roi-10d: Monocular lifting of 2d detection to 6d pose and metric shape," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 2069–2078.

[3] A. Simonelli, S. R. Bulo, L. Porzi, M. López-Antequera, and P. Kontschieder, "Disentangling monocular 3d object detection," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 1991–1999.

[4] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 3354–3361.

[5] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L.-C. Chen, "Mobilenetv2: Inverted residuals and linear bottlenecks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4510–4520.

[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] C. Li, A. Chiang, G. Dobler, Y. Wang, K. Xie, K. Ozbay, M. Ghandehari, 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.

[10] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 580–587.

[11] R. Girshick, "Fast r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 1440–1448.

[12] 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, pp. 91–99, 2015.

[13] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, "Ssd: Single shot multibox detector," in *European conference on computer vision*. Springer, 2016, pp. 21–37.

[14] 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.

[15] J. Redmon and A. Farhadi, "Yolo9000: better, faster, stronger," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 7263–7271.

[16] ——, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.

[17] N. Wojke, A. Bewley, and D. Paulus, "Simple online and realtime tracking with a deep association metric," in *2017 IEEE international conference on image processing (ICIP)*. IEEE, 2017, pp. 3645–3649.

[18] X. Zhou, V. Koltun, and P. Krähenbühl, "Tracking objects as points," in *European Conference on Computer Vision*. Springer, 2020, pp. 474–490.

[19] J.-S. Zhang, J. Cao, and B. Mao, "Application of deep learning and unmanned aerial vehicle technology in traffic flow monitoring," in *2017 International Conference on Machine Learning and Cybernetics (ICMLC)*, vol. 1. IEEE, 2017, pp. 189–194.

[20] Z. Bai, G. Wu, X. Qi, Y. Liu, K. Oguchi, and M. J. Barth, "Infrastructure-based object detection and tracking for cooperative driving automation: A survey," *arXiv preprint arXiv:2201.11871*, 2022.

[21] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, "Monocular 3d object detection for autonomous driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2147–2156.

[22] P. Li, X. Chen, and S. Shen, "Stereo r-cnn based 3d object detection for autonomous driving," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7644–7652.

[23] X. Zhou, D. Wang, and P. Krähenbühl, "Objects as points," *arXiv preprint arXiv:1904.07850*, 2019.

[24] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117–2125.

[25] J. Kannala and S. S. Brandt, "A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 8, pp. 1335–1340, 2006.

[26] S. Shah and J. Aggarwal, "Intrinsic parameter calibration procedure for a (high-distortion) fish-eye lens camera with distortion model and accuracy estimation," *Pattern Recognition*, vol. 29, no. 11, pp. 1775–1788, 1996.

[27] A. Bewley, Z. Ge, L. Ott, F. Ramos, and B. Upcroft, "Simple online and realtime tracking," in *2016 IEEE international conference on image processing (ICIP)*. IEEE, 2016, pp. 3464–3468.

[28] R. E. Kalman, "A new approach to linear filtering and prediction problems," 1960.

[29] H. W. Kuhn, "The hungarian method for the assignment problem," *Naval research logistics quarterly*, vol. 2, no. 1-2, pp. 83–97, 1955.

[30] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez, and V. Koltun, "CARLA: An open urban driving simulator," in *Proceedings of the 1st Annual Conference on Robot Learning*, 2017, pp. 1–16.

[31] 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, no. 2, pp. 303–338, 2010.

[32] S. Xu, H. Peng, Z. Song, K. Chen, and Y. Tang, "Accurate and smooth speed control for an autonomous vehicle," in *2018 IEEE Intelligent Vehicles Symposium (IV)*. IEEE, 2018, pp. 1976–1982.

[33] S. Xu and H. Peng, "Design, analysis, and experiments of preview path tracking control for autonomous vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 1, pp. 48–58, 2019.