2023-01-0800 Published 11 Apr 2023

Vehicle Lateral Offset Estimation Using Infrastructure Information for Reduced Compute Load

Sachin Sharma and Johan Fanas Rojas Western Michigan University

Ali Riza Ekti and Chieh (Ross) Wang Oak Ridge National Laboratory

Zachary Asher and Rick Meyer Western Michigan University

Citation: Sharma, S., Fanas Rojas, J., Ekti, A.R., Wang, C.(R.) et al., "Vehicle Lateral Offset Estimation Using Infrastructure Information for Reduced Compute Load," SAE Technical Paper 2023-01-0800, 2023, doi:10.4271/2023-01-0800.

Received: 08 Nov 2022 Revised: 09 Jan 2023 Accepted: 26 Jan 2023

Abstract

Accurate perception of the driving environment and a highly accurate position of the vehicle are paramount to safe Autonomous Vehicle (AV) operation. AVs gather data about the environment using various sensors. For a robust perception and localization system, incoming data from multiple sensors is usually fused together using advanced computational algorithms, which historically requires a high-compute load. To reduce AV compute load and its negative effects on vehicle energy efficiency, we propose a new infrastructure information source (IIS) to provide environmental data to the AV. The new energy–efficient IIS, chip–enabled raised pavement markers are mounted along road lane lines and are able to communicate a unique identifier and their global navigation satellite system position to the AV. This new IIS is incorporated into an energy efficient sensor fusion strategy that combines its information with that from traditional sensor. IIS reduce the need for camera imaging, image processing, and LIDAR use and point cloud processing. We show that IIS, when combined with traditional sensors, results in more accurate perception and localization outcomes and a reduced AV compute load.

Introduction

Typical Autonomous Vehicle (AV) systems can be divided into four main subsystems: perception, localization, path planning, and control. For perception, AVs use input from multiple sensors to extract information about the driving environment and locate current and future states of stationary and dynamic objects using cameras, radio detection and rangings (RADARs), and light detection and rangings (LIDARs). Localization is the process of locating the vehicle globally with respect to world coordinates using inputs from sensors and information sources like global navigation satellite system (GNSS), inertial measurement unit (IMU), odometry, and/or high definition (HD) maps. Incoming data from AV sensors and information sources are typically fused and interpreted using advanced sensor fusion algorithms for highly accurate perception and localization subsystems, which requires an in-vehicle computer with very high operating frequencies and multiple processors. Given the outputs from perception and localization subsystems, the path planning subsystem finds an optimal and safest trajectory for the AV to reach its desired destination. The control subsystem outputs required acceleration, torque, and steering angle values to follow the trajectory obtained through path planning. Power drawn from all the AV sensors including the in-vehicle computer, and computational load from sensor fusion algorithms for perception and localization, and path planning and control executions reduce the energy efficiency of an AV. Combining alternative sensor fusion methodologies with optimal energy management techniques [1, 2, 3], AV energy usage can be significantly reduced.

Lane detection, which can be considered as an output of perception process, is a crucial task for AVs and advanced driver assistance systems (ADAS). With advancements in deep learning in recent years, a number of deep learning based lane line detection methods have been proposed recently. Lane and drivable region detection methods based on deep convolutional neural networks (DCNNs) [4, 5] predict whether each pixel obtained through semantic segmentation belongs to a

lane line or not. Lane detection methods based on deep recurrent neural networks (DRNNs) recursively process the input image signal and build full connection layers between them for status propagation [6]. These methods demand higher computational power as a larger and deeper convolutional network is required to extract high–level semantic information. The input vector dimensions for pictures to be processed by DRNNs are particularly big since these networks require sequential data to obtain the sequence token dependencies that aid in categorization.

As an alternative, infrastructure information sources (IISs) can be used to facilitate perception and sensing by supporting vehicle–to–infrastructure (V2I) information exchange while reducing the required computational load. Stephens et al. [7] proposed safety applications capturing relevant data from roadside infrastructure sensors and in-vehicle sensors to determine potential crash hazards. Applications described here had an infrastructure- and a vehicle-based components which issued advisories to make the driver aware of the hazards in time to take action to prevent the crash.

An energy–efficient infrastructure sensor, chip–enabled raised pavement marker (CERPM) can be used to detect lane line and drivable region for AVs improving perception while reducing the compute load [8]. Integrated transmission technology fits inside commercially available Raised Pavement Markers (RPMs) as shown in Figure 1. CERPMs placed on the road have capabilities for data processing and wireless data exchange to support cooperative driving automation (CDA). CERPMs transmit GNSS coordinates of their location to the on–board vehicle receiver which transmits information to the AV control system.

CERPM application for lane and drivable region detection includes an energy savings of at least 90% when compared to a commercial solution [8]. To make the most of this application, the update frequency for the lateral offset measurements coming from the camera can be lowered as DCNNs demand higher energy use. An energy efficient sensor fusion strategy can be established if the update frequency of camera measurements are lowered. In this approach, we do not lose high level semantic information obtained from camera, and can still provide accurate positioning information by relying on IIS, like CERPM, at a reduced compute load.

FIGURE 1 Transmitter Setup for CERPM

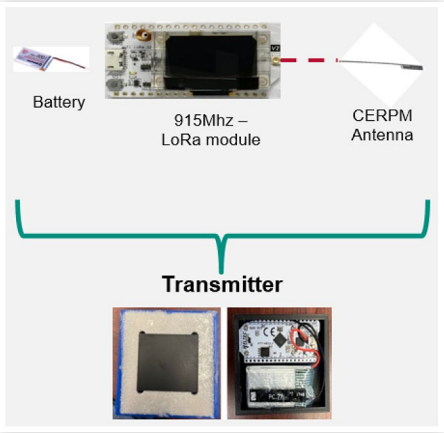
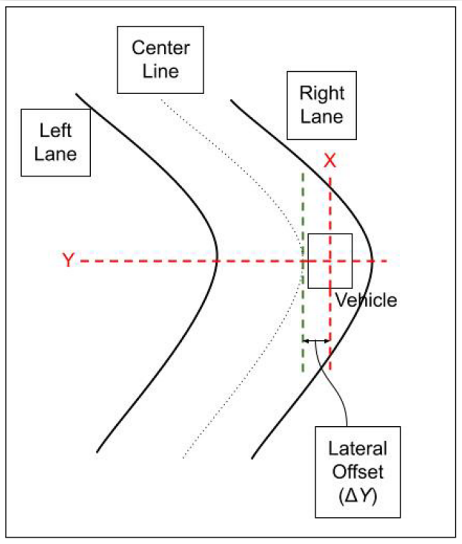


FIGURE 2 Ego-vehicle lane information



In this paper, we develop a sensor–fusion strategy for AV lane keeping using inputs from a traditional imaging sensor: camera and infrastructure sensors. We estimated the lateral offset of the ego–vehicle from the center of the lane using camera, CERPMs, and a combination of both. For an ego-vehicle in a lane, ΔY, as shown in Figure 2 is the perpendicular distance between a line that passes through the center of the lane and a line that passes through the center of gravity of the ego-vehicle. The main contribution of this paper lie in the following:

1. Real world like sensory data is generated using CARLA simulator [9]. CERPMs were simulated inside the CARLA simulator. Image messages from camera sensor and GNSS messages from CERPMs were published through Robot Operating System (ROS) nodes.
2. Kalman Filter is applied to predict ΔY using information from CERPMs and camera messages.
3. Camera–CERPM group sensor fusion is employed in harmony to predict ΔY.
4. Camera–CERPM asyncronous sensor fusion is further evaluated to predict ΔY.

The remainder of the paper is organized as follows. The methodology section reviews the lane detection methods used and data fusion methodologies. The result section reports the experiments performed and their results. The conclusion section concludes our work and briefly discusses about possible future work.

Methodology

In this section, we describe lateral lane offset estimation using inputs from camera and CERPMs seperately. Sensor data-fusion methodologies are formulated and tested in the CARLA simulator.

Lateral Offset Estimation Using Camera

Lateral offset estimation starts at lane line detection. For lane detection, an encoder-decoder DCNN, UNet [10] was used. U-Net is a fully convolutional network which has been demonstrated to be applicable to various semantic segmentation purposes. Raw images of size 512x512x3 were converted into grayscale images with a single channel and passed through the U-Net segmentation architecture as shown in Figure 3.

The output of the U-Net architecture is a masked image where lane pixels are identified after threshold processing to remove associated interference. Step-by-step explanation of a series of steps performed to get the lateral offset from the masked image are described below and illustrated in Figures 4 and 5.

1. Perspective transform to extract region of interest (ROI): The input to this stage is the masked image shown in Figure 4(b) which is converted to a top down view of the ROI using perspective transform as shown in Figure 4(d) and reduce subsequent processing. The four vertex of the region to be processed were identified and are shown in Figure 4(c) and Equation 1 was used to map the relationship between pixel (x, y) of the top down image and pixel (u, v) of the masked image.

$$\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} a_1 & a_2 & b_1 \\ a_3 & a_4 & b_2 \\ c_1 & c_2 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \quad (1)$$

where $\begin{bmatrix} a_1 & a_2 \\ a_3 & a_4 \end{bmatrix}$ is the rotation matrix, $\begin{bmatrix} b_1 \\ b_2 \end{bmatrix}$ denotes the translation vector and $[c_1 \quad c_2]$ represents the projection vector. Perspective transformation eliminates the majority of image's interference and projects the lane lines to a relatively parallel position to make subsequent processing easier [12].

2. Lane fitting: Traditional lane search algorithms are based on Hof linear transformation [13] which require input images of high resolution and does not adapt to real scenes with high interference due to poor recognition effect. A bottom-up scanning approach known as sliding window algorithm [14] was utilized to identify and track the lane lines as shown in Figure 5(a). This method preserves the coordinate of the non-zero pixels inside the windows and adjust the position of the next window accordingly. Quadratic curves are fitted along the right and left lines using the least square method.
3. Lateral offset estimation: After lane fitting, a coordinate system was established where the vertical center line of the image is considered as X-axis and horizontal line at the bottom of the image is considered as Y-axis. Assuming the center of the vehicle is the center of the image and the origin of the Y-axis at the center of the lane and considering d_1 and

FIGURE 3 U-Net Architecture for image segmentation [11]

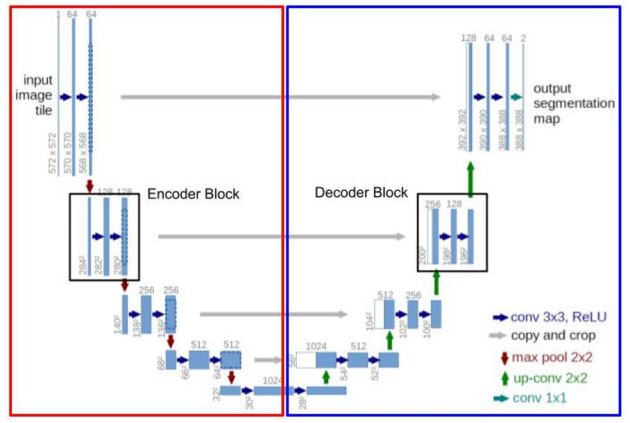
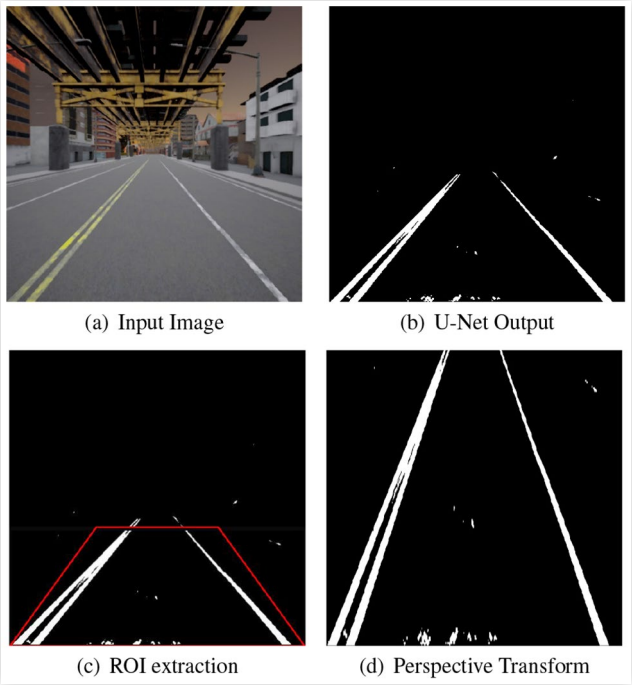


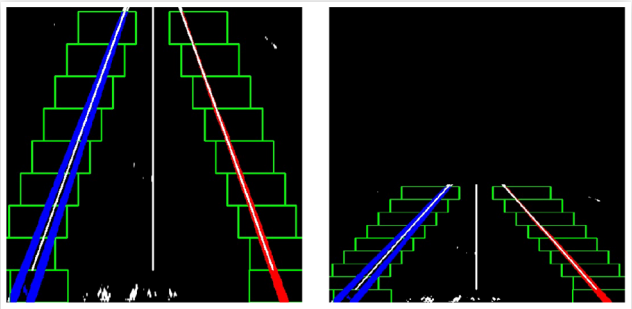
FIGURE 4 Input to the U-Net architecture, the masked output, selected region of interest, perspective transformation of selected region.



(a) Input Image (b) U-Net Output

(c) ROI extraction (d) Perspective Transform

FIGURE 5 Step 2 for lateral offset estimation showing sliding window method and inverse perspective transform of the warped image.



(a) Lane fitting in transformed image (b) Inverse perspective transform

d_2 as the Y-pixel locations of the bottom left and bottom right lines respectively, the pixel location of the lane center d_3 is calculated as the average of d_1 and d_2. The lateral offset, ΔY can be calculated using:

$$\Delta Y = d_3(MPP) \tag{2}$$

where MPP is the meters per pixel in the horizontal direction to convert the distance units from pixels to meters.

Lateral Offset Estimation Using CERPMs

CERPMs transmit GNSS location (latitude, longitude, and altitude) of their location to the vehicle computer. Global GNSS messages from CERPMs were converted into cartesian coordinates (x, y) using:

$$\begin{aligned} x &= R\cos(\lambda)\cos(\phi) \\ y &= R\cos(\lambda)\sin(\phi) \end{aligned} \tag{3}$$

where λ and ϕ are the difference in target and current latitude and longitude in radians respectively and R is the radius of the earth. Assuming the ego–vehicle current location in cartesian coordinates as the origin, the CERPMs are located in the ego-vehicle frame of reference. Polynomial curves of degree 2 are fitted along the right and left lanes using the least square method. The lateral offset, ΔY can be calculated using:

$$\Delta Y = \frac{d_4 + d_5}{2} \tag{4}$$

where d_4 and d_5 are the y-coordinate of the left and right lanes respectively in the ego-vehicle frame of reference. Figure 6 shows the detected left and right lanes from the CERPM information.

Kalman Filter Based Estimators

Estimation of lateral offset at each timestep is input to the path-planning and controls sub-system for AV lane-keeping. Kalman filter, an optimal estimator based on a recursive computational methodology for estimating the state of a discrete–data controlled process from typically noisy measurements was used to model the lateral offset over time in the y-direction. The linear time invariant system subject to random process noise $w(k)$ and measurement noise $v(k)$ and uncertain random initial condition can be modeled as Equations 5 and 6.

$$x(k+1) = Ax(k) + Bu(k) + Gw(k) \tag{5}$$

$$z(k) = Hx(k) + v(k) \tag{6}$$

where A,B, G, z, and H denote system matrix, input matrix, process noise gain matrix, sensor measurement model matrix and measurement respectively. Kalman filter involves a series of steps for state estimation which are reviewed in the Appendix section (Equations 13-17). The linear time invariant system specific to our measurements for lateral offset estimation is shown in Equations 7 and 8 where Δy is the lateral offset from the center line and V_y denotes the lateral velocity of the ego-vehicle.

$$\begin{bmatrix} \Delta y(k+1) \\ V_y(k+1) \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta y(k) \\ V_y \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{\Delta t^2}{2} \end{bmatrix} w(k) \tag{7}$$

$$z_y(k) = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} \Delta y(k) \\ V_y(k) \end{bmatrix} + v(k) \tag{8}$$

Group Sensor Method for Camera–CERPM Fusion

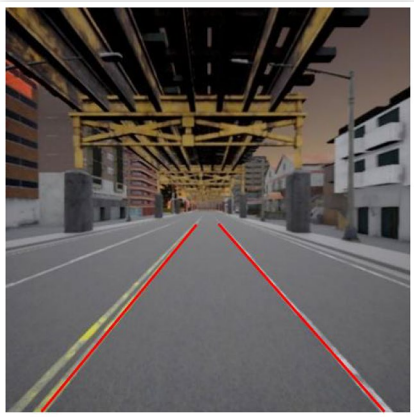
Group sensor method combines measurements for all sensors and measurement model into a single sensor and its formulation is based upon synchronized measurements. If the lateral offset measurements coming from the camera and CERPMs can be synchronized, a group sensor method can be formulated to fuse information coming from camera and CERPMs. The measurement model for linear time invariant system for group sensor method is shown in Equation 9. $z_1(k)$ and $z_2(k)$ are the offset measurements coming from camera and CERPM respectively and v_1 and v_2 are corresponding measurement noise for each sensor.

$$\begin{bmatrix} z_{1y}(k) \\ z_{2y}(k) \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \Delta y(k) \\ V_y(k) \end{bmatrix} + \begin{bmatrix} v_1(k) \\ v_2(k) \end{bmatrix} \tag{9}$$

Asynchronous Sensor Fusion for Camera–CERPM

To investigate energy efficient sensor fusion strategy, the frequency of camera sensor (frames per second) can be decreased keeping the frequency of CERPM information same as before, an asynchronous estimator can be formulated. For asynchronous measurement cases, the time between 1, 2, …., k – 1, k may not be constant. If we consider $T_{S,k}$ as the

FIGURE 6 Lane line detections from CERPM information.



current time step which is the time between $k-1$ and k, the process model over $T_{S,k}$ is shown in Equation 10.

$$\begin{bmatrix} \Delta y(k+1) \\ V_y(k+1) \end{bmatrix} = \begin{bmatrix} 1 & T_{S,k} \\ 0 & 1 \end{bmatrix} \begin{bmatrix} \Delta y(k) \\ V_y(k) \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{T_{S,k}^2}{2} \end{bmatrix} w(k) \quad (10)$$

where $A_{Ts,k}$, $B_{Ts,k}$, $G_{Ts,k}$, $H_{Ts,k}$ need to be evaluated at each k as $T_{S,k}$ is not constant.

Results

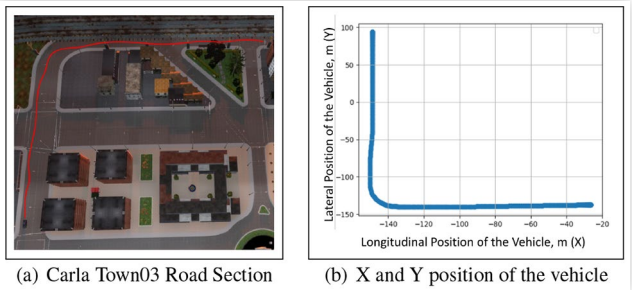
The simulation setup is discussed first, and then the results for estimates using different sensor fusion strategies are provided.

Simulation Setup

CARLA is an open source simulator we utilized for developing, validating, and testing algorithms for AV systems. Carla offers a simulation setting with a variety of sensor specifications, environmental factors, and automobiles, among other things. It gives the user the power to design environments that can be tested and validated for autonomous/ADAS driving behaviors. To facilitate two way communication with Robot Operating System (ROS), CARLA ROS bridge can be used to get information from the CARLA server. For the camera sensor, the sensor update rate, image size, field of view, and spawn points are customizable inside the CARLA simulator. CERPMs that transmit GNSS message of their location were modeled in the simulator and the sensor update rate was kept at constant 20 Hz for all of the test cases. CERPMs were spawned at both left and right lanes and the adjacent distance between the CERPMs was kept at a length of 5 m. A vehicle was spawned in Town03 of the CARLA simulator and a route was chosen to evaluate the lateral offset estimators formulated in the methodology section. Figure 7 shows the longitudinal and lateral position of the vehicle when travelling on the chosen route.

CARLA simulator has waypoints which are 3-D directed points corresponding to a lane. The location of the center of the lane was calculated as the midpoint between the right and left lanes. CARLA simulator publishes pose information to ROS as odometry messages. The true lateral location of the ego-vehicle was subtracted from the lateral location of the center of the lane to get the ground truth data.

FIGURE 7 Road section for lateral offset estimation tests in Town03 of the CARLA simulator.



(a) Carla Town03 Road Section (b) X and Y position of the vehicle

Lane Lateral Offset Estimation Setup

Lateral offset estimators were developed using a single sensory outputs from individual sensors. First, the lateral offset estimator was modeled for the camera. The measurement noise associated with lateral offset measurement can be divided as: (a) measurement noise associated with the camera sensor itself (b) measurement noise associated with the mask generated by U-Net. As the image obtained through the CARLA simulator is undistorted, the measurement noise associated with the camera sensor is 0. The mean intersection over union ($mIoU$) of the trained U-Net model for mask generation was 98%. For 512 x 512 x 3 channel image, the measurement noise in meters was calculated as:

$$\sigma_{p\Delta y} = (1 - mIoU)W \quad (11)$$

$$\sigma_{\Delta y} = \sigma_{p\Delta y} D_0 \quad (12)$$

where W is the width of the image, $\sigma_{p\Delta y}$ stands for the standard deviation of the offset in pixels, $\sigma_{\Delta y}$ denotes the standard deviation of the offset in meters, and D_0 represents the meter per pixel value for Town03 in CARLA simulator (given as 0.4 cm). $v(k)$ for camera measurements was modeled as a normal distribution with zero mean and a standard deviation of 4 cm as calculated using Equations 11 and 12.

The overall measurement noise $v(k)$ associated with CERPM and GNSS derived offset measurements can be determined by adding var(X) and var(Y) since CERPM and GNSS measurements are independent where var(X) and var(Y) are measurement noise associated with CERPMs and GNSS respectively. Modern GNSS systems incorporate real-time kinematic (RTK) positioning which uses carrier based ranging and provides position estimates up to centimeter level accuracy [15]. Likewise, preloaded GNSS points in the CERPMs are assumed to be measured using a GNSS sensor that is capable of providing RTK corrections. Measurement noise of mean 0 with 1 cm standard deviation was added to CERPM measurements and the in-vehicle GNSS sensor, hence modelling the overall $v(k)$ as a normal distribution with mean 0 and standard deviation of 2 cm.

Process noise, $w(k)$ for linear time invariant systems formulated in Equations 7 and 8 is the lateral acceleration of the ego-vehicle. A lateral acceleration value of 0 mean and 0.1g standard deviation was used to model $w(k)$ for both offset measurements.

Lane Lateral Offset Estimation Simulation Results

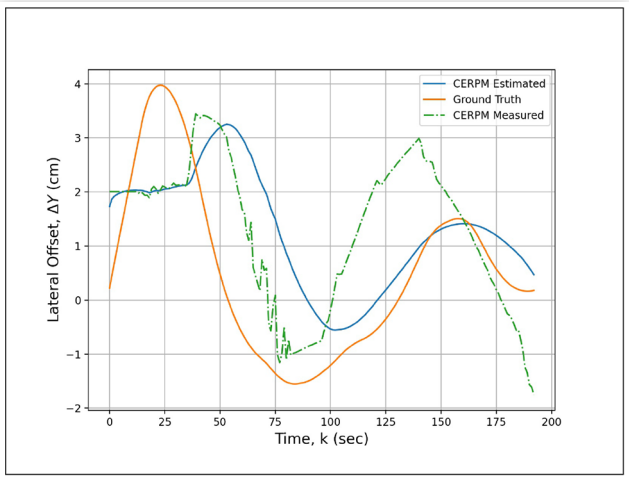
To evaluate the lateral offset estimators developed in the methodology section, ground truth lateral offset data was obtained from the simulator. Lateral offset estimators using single

sensory inputs were modeled first and are shown in Figures 8(a) and 8(b). Measured values, estimates, and ground truth offsets are plotted and the root mean square error (RMSE) for measured values and estimates were calculated. Measured values had an RMSE of 1.65 cm and estimates had an RMSE of 1.58 cm for model based on CERPM offset measurements. Likewise for the camera offset measurement model, the RMSE of measured values was 3.01 cm and the RMSE for estimates were 2.10 cm.

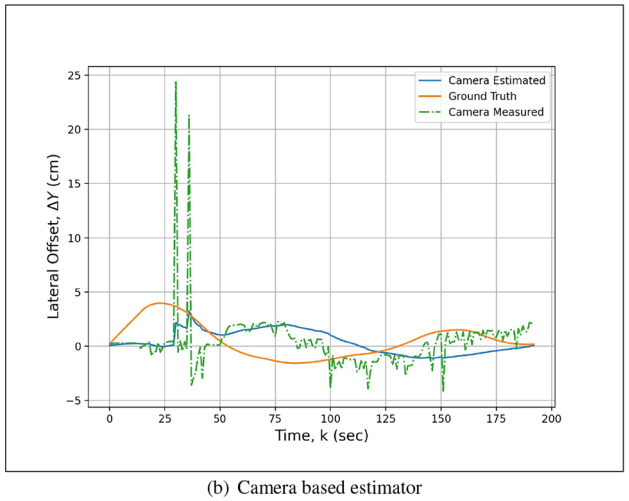
A time synchronizer function in ROS within the message filters utility library takes in messages of different types from multiple sources and outputs them only if it has received a message on each of those sources [16]. After message synchronization, the group sensor model described in Equation 9 was applied to estimate the lateral offset of the ego-vehicle. RMSE of the fused output using group sensor method was 1.52 cm.

Ego-vehicle obtains continuous information from the CERPMs with a very low compute load associated with it. On the other hand, raw image output needs to undergo a series of steps before outputting the lane offset. With added functionalities from the CERPM, the frequency of image messages per second can be decreased which will decrease the computational load associated with image processing. The camera update rate was decreased to 5 frames per second from 20 frames per second and the asynchronous model shown in Equation 10 was used to obtain lateral offset estimates. Figure 10 shows the comparison of lateral offset estimates obtained from asynchronous fusion with camera measurements, CERPM measurements, and the ground truth measurements. The RMSE of fused output using asynchronous sensor fusion was 1.55 cm. This method produced lateral offset estimations that were nearly as accurate as those produced by group sensor method while decreasing the computational need associated with image processing by 4 times as the camera update rate was decreased from 20 frames per second to 5 frames per second for the asynchronous sensor fusion method. This translates into an approximate reduction of compute load power of 4 times for image processing as shown in Table 1.

FIGURE 8 Lateral offset estimators based on single measurement input.



(a) CERPM based estimator



(b) Camera based estimator

FIGURE 9 Lateral offset estimator based on group sensor method

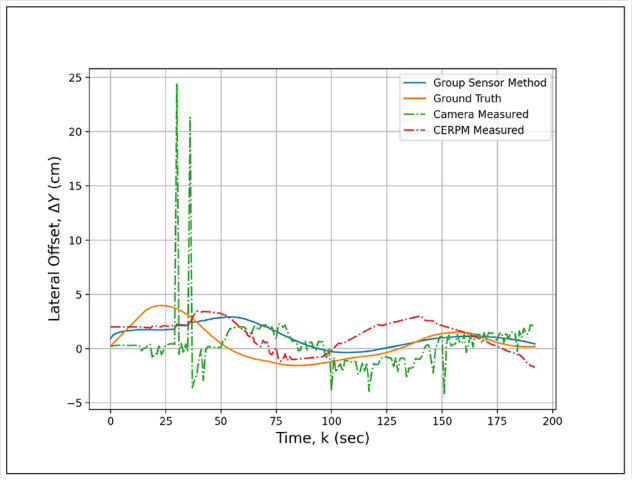


FIGURE 10 Lateral offset estimator based on asynchronous fusion of CERPM-Camera measurements

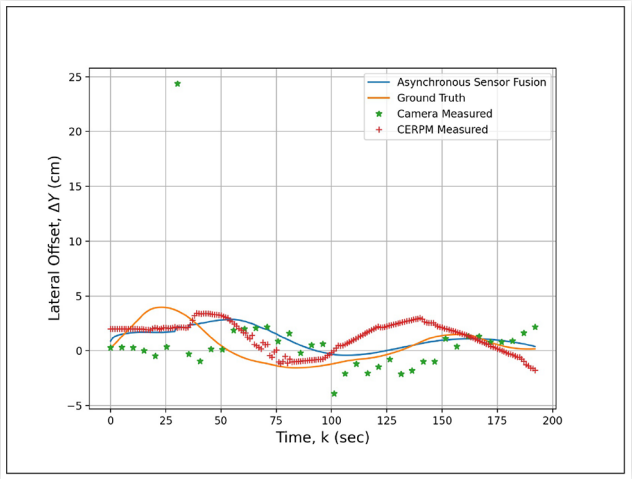


TABLE 1 Table showing required compute load for different lateral offset estimation methods. CL_{cam} and CL_{CERPM} are compute load required for image processing at 20 frames per second and CERPM measurements at 20 *Hz* respectively.

Method	Compute Load
Group Sensor Method	$CL_{cam} + CL_{CERPM}$
Asynchronous Sensor Fusion Method	$\frac{1}{4}CL_{cam} + CL_{CERPM}$

Conclusions

This work presented the development of lateral lane offset measurement estimators based on inputs from the camera sensor, CERPMs, and combinations of both. Four different estimators were modeled and tested in a road section of CARLA simulator: camera alone, CERPM alone, group sensor fusion, and asynchronous sensor fusion. The RMSE of the offset measurements when compared to ground truth were within 2.2 cm. Group sensor method had the lowest RMSE among the models tested. When compared to asynchronous sensor fusion, RMSE of group sensor method is lower by 0.03 cm but the computational load associated with image processing in group sensor method is estimated to be four times more than the asynchronous method. Asynchronous fusion allows for the fusion of highly accurate lateral offset measurements from CERPMs, which have a higher update rate, with image-based offset measurements, which have a lower update rate, for improved lane keeping while preserving high-level semantic information and conserving energy.

In the near future, lateral lane offset estimators modeled in the CARLA simulator will be tested in real-world scenarios. A control strategy based on lateral lane offset measurements will be developed. Data fusion from other IISs like HD maps and radar retroreflectors will be evaluated for more accurate offset measurements. Developed lateral offset estimators and control strategy will be tested in a variety of situations, such as on rural roads, inclement weather, and in locations with poor GNSS coverage. Additional testing will include a wider variety of measurement noises, as could be expected in the real world.

The ego-vehicle receives data from the CERPMs wirelessly using the LoRa communication protocol. The lateral lane offset would be affected by communication impairments between the ego-vehicle and the CERPMs on the road. Future studies will simulate failure case scenarios where a portion of the CERPMs are turned off while estimating the lateral lane offset to determine the impact of such communication disruptions. In these situations, estimated lane lines will be evaluated for goodness of fit before being used or not. For the purposes of this study, the on-board GNSS receiver is simulated as a contemporary GNSS system that includes RTK and offers location estimates with a precision of up to 2 cm. In the future, lateral lane offset measurements will be estimated when simulating a traditional GNSS system which typically has a higher measurement error.

References

1. Araghi, F.M., Rabinwoitz, A., Ang, C.C., Sharma, S. et al., "Identifying and Assessing Research Gaps for Energy Efficient Control of Electrified Autonomous Vehicle Ecodriving," in: , *Machine Learning and Optimization Techniques for Automotive Cyber-Physical Systems*, (Springer Nature, 2022) In review.
2. Motallebiaraghi, F., Rabinowitz, A., Holden, J. et al., *High-Fidelity Modeling of Light-Duty Vehicle Emission and Fuel Economy Using Deep Neural Networks* (SAE Technical, 2021)
3. Motallebiaraghi, F., Yao, K., Rabinowitz, A., Hoehne, C. et al., "Mobility Energy Productivity Evaluation of Prediction-Based Vehicle Powertrain Control Combined with Optimal Traffic Management," SAE Technical Paper 2022-01-0141, 2022, https://doi.org/10.4271/2022-01-0141.
4. Lee, D. and Liu, J., "End-to-End Deep Learning of Lane Detection and Path Prediction for Real-Time Autonomous Driving," *CoRR* abs/2102.04738 (2021).
5. Kadav, P., Goberville, N., Motallebiaraghi, F., Fong, A., and Asher, Z.D., "Tire Track Identification: Application of u-net Deep Learning Model for Drivable Region Detection in Snow Occluded Conditions," *in Intelligent Transportation Systems World Congress*.
6. Zou, Q., Jiang, H., Dai, Q., Yue, Y. et al., "Robust Lane Detection from Continuous Driving Scenes Using Deep Neural Networks," *IEEE Transactions on Vehicular Technology* 69 (jan 2020): 41-54.
7. Stephens, D., Schroeder, J., and Klein, R., "Vehicle-toinfrastructure (v2i) Safety Applications: Performance Requirements, vol. 1, Introduction and Common Requirements," Aug 2015. Tech Report.
8. Sharma, S., Ekti, A.R., Rojas, J.F., Brown, N.E., et al., "Development and Evaluation of Chip-Enabled Raised Pavement Markers for Lane Line Detection," in *2022 IEEE Sensors*, pp. 1–4, 2022.
9. Dosovitskiy, A., Ros, G., Codevilla, F., López, A.M. et al., "CARLA: An Open Urban Driving Simulator," *CoRR* abs/1711.03938 (2017).
10. Ronneberger, O., Fischer, P., and Brox, T., "U-Net: Convolutional Networks for Biomedical Image Segmentation," *CoRR* abs/1505.04597 (2015).
11. Kadav P., Sharma S., Araghi F.M., and Asher Z.D., "Development of Computer Vision Models for Drivable Region Detection in Snow Occluded Lane Lines," *Machine Learning and Optimization Techniques for Automotive Cyber-Physical Systems*, Springer Nature. In Review.
12. Zhu H., "An Efficient Lane Line Detection Method Based on Computer Vision," *Journal of Physics: Conference Series*, vol. 1802, p. 032006, mar 2021.
13. Cao, X., Liu, D., and Ren, X., "Detection Method for Auto Guide vehicle's Walking Deviation Based on Image Thinning and Hough Transform," *Measurement and Control* 52, no. 3-4 (2019): 252-261.
14. He, J., Sun, S., Zhang, D., Wang, G., and Zhang, C., "Lane Detection for Track-Following Based on Histogram Statistics," in *2019 IEEE International Conference on Electron Devices and Solid-State Circuits (EDSSC)*, pp. 1–2, 2019.

15. "Centimeter precision gnss explained - rtk in detail," Jul 2022.
16. Quigley, M., Conley, K., Gerkey, B., Faust, J., et al., "Ros: An Open-Source Robot Operating System," vol. 3, 01 2009.
17. Welch, G. and Bishop, G., "An Introduction to Kalman Filter," in *SIGGRAPH 2001*, 1995.

Contact Information

Richard T. Meyer, Ph.D., P.E.
richard.meyer@wmich.edu

Acknowledgments

This material is based upon work supported by the US Department of Energy (DOE)'s Office of Energy Efficiency and Renewable Energy (EERE) under the Energy Efficient Mobility Systems program under DE–EE–0009657.

Definitions, Abbreviations

AV - Autonomous Vehicle
CERPM - Chip–Enabled Raised Pavement Marker
ROS - Robot Operating System
GNSS - Global Navigation Satellite System
DCNN - Deep Convolutional Neural Networks
DRNN - Deep Recurrent Neural Networks
CDA - Co-operative Driving Automation
ROI - Region of Interest
HD - High Definition
IMU - Inertial Measurement Unit

Appendix A

Kalman filter involves five steps which are shown in Equations 13 to 17 [17].

$$M(k) = A_{k-1}P(k-1)A_{k-1}^T + G_{k-1}WG_{k-1}^T \tag{13}$$

where M is the prediction error covariance matrix.

$$\bar{x}(k|k-1) = A_{k-1}\hat{x}(k-1|k-1) + B_{k-1}u(k-1) \tag{14}$$

$$K(k) = M(k)H_k^T\left(H_kM(K)H_k^T + V\right)^{-1} \tag{15}$$

where K is the Kalman filter gain.

$$P(k) = \left(I - K(k)H_k\right)M(k) \tag{16}$$

where P is the estimate error covariance matrix.

$$\hat{x}(k|k) = \bar{x}(k|k-1) + K(k)\left[z(k) - H_k\bar{x}(k|k-1)\right] \tag{17}$$

ISSN 0148-7191