Vehicle Position and Context Detection Using V2V Communication

Paul Watta, *Member, IEEE*, Ximu Zhang, *Student Member, IEEE*, and Yi Lu Murphey, *Fellow, IEEE*

***Abstract*—A pre-crash detection and warning system in a host vehicle needs to accurately determine the position of each remote vehicle in its vicinity and the context of the driving environment. ADAS (Advanced driver-assistance systems) have extensively used camera radar and LIDAR for automatic detection of vehicles, pedestrians and other road users and their behaviors. However, these vehicle-resident sensors have short operation ranges and require objects to be within the line-of-sight. V2V communication has emerged to be a promising technology to augment vehicle-resident sensors with extended ability of an overall vehicle safety system by addressing a broader range of crash scenarios with improved warning timing. In this paper we present an intelligent system, Geo+NN, developed using the synergy of neural network and geometric modeling. We extract the key geometric features using an analytic geometric model and use them as input to a neural network that is trained on real-world V2V signals to detect and predict remote vehicles' positions. Geo+NN system is evaluated on V2V communication data recorded during real-world driving trips by vehicles installed with DSRC devices. Experimental results show that Geo+NN has the capabilities of effectively detecting and predicting remote vehicles within the context of 8 different positions.**

***Index Terms*—Intelligent vehicles systems (ITS), vehicle-to-vehicle communication (V2V), relative position, host vehicle, remote vehicle, GPS, neural networks, situational awareness.**

I. Introduction

THE ability to assess in real-time and understand the *context* of a traffic scene, also referred to as *situational awareness*, is essential for both human drivers and advanced driver assistance systems (ADAS). In this research we define the context and situational awareness with respect to the information the host vehicle needs in order to make safe maneuvers, such as a lane change, turning, passing, yielding, etc. One of the most essential tasks in achieving such situational awareness is to identify and predict the *relative position* of other nearby vehicles on the road.

Consider the schematic diagram shown in Fig. 1. Here, the vehicle of interest, or *host vehicle* **H** is driving on a 3-lane road near six other vehicles, called *remote vehicles*, and denoted $\mathbf{R}_1$, $\mathbf{R}_2$, …, $\mathbf{R}_6$. Note that the numbering or labels of the remote vehicles is arbitrary. To assess the traffic context of the host vehicle, we need to determine that there are remote vehicles driving in parallel on both the left and the right adjacent lanes: $\mathbf{R}_1$ and $\mathbf{R}_2$, there are remote vehicles in the same lane both ahead and behind: $\mathbf{R}_3$ and $\mathbf{R}_4$, and there is a remote vehicle ahead in the left adjacent lane $\mathbf{R}_5$. An interesting case in this context is the presence of $\mathbf{R}_6$ because it may not be visible at all to the host vehicle; for example suppose $\mathbf{R}_3$ is a large SUV and $\mathbf{R}_6$ is a motorcycle or small sedan. $\mathbf{R}_6$, though, is still a potential threat to the host vehicle because if $\mathbf{R}_6$ suddenly and unexpectantly slams on the brakes, with the delay caused by $\mathbf{R}_3$'s reaction time, followed by the delay from the host's reaction time, the host vehicle may not be able to avoid a three-vehicle crash. We will refer to this scenario as **H**-$\mathbf{R}_3$-$\mathbf{R}_6$.

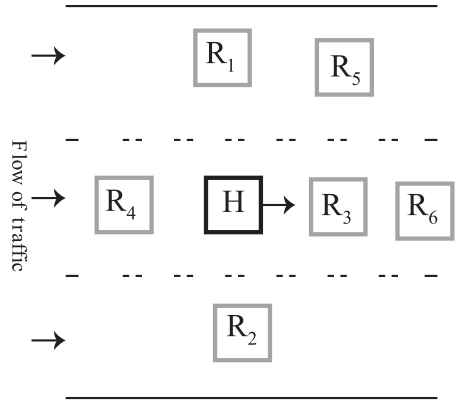


Fig. 1. A schematic diagram showing a traffic context with a host vehicle **H** and several remote vehicles, $\mathbf{R}_1 \sim \mathbf{R}_6$, which are in communication with the host.

Over the years, camera and radar-based ADAS systems have been extensively developed for crash avoidance and determining traffic context for driver assist systems and autonomous driving [1]–[6]. While these "vehicle-resident" technologies can be highly effective in many traffic scenarios, they have limitations of short seeing range and line-of-sight. For example, vision-based systems may fail in the same way that human drivers would for the **H**-$\mathbf{R}_3$-$\mathbf{R}_6$ scenario mentioned above. In addition, camera systems exhibit degraded performance in the presence of adverse weather conditions, variations in lighting, and occlusions.

Recently, there has been growing interest in using vehicle-to-vehicle (V2V) communications whereby vehicles exchange in real-time position, velocity, and heading information. Such information can be used to help determine the relative positions of other vehicles on the road, which can be used to give robust

Manuscript received February 7, 2020; revised August 30, 2020; accepted November 25, 2020. Date of publication December 11, 2020; date of current version November 23, 2021. This work was supported in part by a grant from the University of Michigan MCity, and in part by Michigan Institute for Data Science. *(Corresponding author: Paul Watta.)*

The authors are with the Department of Electrical and Computer Engineering, University of Michigan – Dearborn, Dearborn, MI 48128 USA (e-mail: watta@umich.edu; ximuz@umich.edu; yilu@umich.edu).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIV.2020.3044257.

Digital Object Identifier 10.1109/TIV.2020.3044257

and early warning to drivers about impending danger. There are several available V2V communication technologies and protocols that have the potential to be implemented in commercial vehicles, such as dedicated short-range communication (DSRC) and cellular technologies, such as fourth (4G) or fifth (5G) generation mobile systems. The major advantages of V2V communications over the vehicle-resident sensor systems are their longer ranges of operations and overcoming the line-of-sight requirement [7]–[8]. This longer detection distance and ability to *see* around corners or *through* other vehicles helps V2V-equipped vehicles perceive some potential crash scenarios sooner than sensors, cameras, or radar can, and warn drivers accordingly. According to the National Highway Traffic Safety Administration, V2V and V2I applications have the potential to address 80% of unimpaired crashes [9]. However, V2V communication technologies have challenges, such as GPS noise and network security, and limitations to address crash scenarios such as lane and road departure [7]. Many researchers believe that a robust sensor system should include a V2V communication component to augment the vehicle-resident sensor systems to extend the capabilities of the overall vehicle safety system to address a broader range of crash scenarios with improved warning timing and reducing the number of false alarms [7].

In this paper, we present our research in detecting and predicting the relative position of a remote vehicle to a host vehicle. The problem is important for a driver before making a maneuvering such as lane change, turning, passing, parking, and yielding. The problem is formulated as a classification problem where the classes represent the various relative positions that the remote vehicle can assume with respect to the host; for example ahead in the left adjacent lane (class 1), behind in the same lane (class 7), etc. The input to the classifier is a feature vector comprised of the V2V data exchanged between the host and remote vehicles. An intelligent system, which we call *Geo+NN*, has been developed based on geometric modeling and neural learning. We evaluated the Geo+NN system using the V2V data extracted from real world driving trips recorded by a fleet of over 1800 vehicles installed with DSRC communication devices.

The remainder of this paper is organized as follows. Section IIreviews related work in the area of relative position detection. In Section III, we introduce a new V2V-based method for remote vehicle position detection and prediction. Section IVpresents experimental results using naturalistic V2V driving data in an urban setting in the USA with vehicles equipped with dedicated DSRC communication devices. Finally, Section Vprovides a summary of the research results and a discussion for future work.

II. Related Work

Much of the pre-crash detection research has focused on using vehicle-resident sensors, such as cameras, radar, and/or light detection and ranging (LIDAR) devices, along with computer vision algorithms for solving problems related to vehicle safety systems, such as vehicle detection, pedestrian detection, traffic sign detection, etc. Some of these vehicle safety technologies have been deployed in existing vehicles, such as forward collision warning systems and blind spot detection systems. However, systems based on vehicle-resident sensors are not reliable in a variety of conditions, such as bad weather, poor lighting, and occlusions—conditions where the driver would most benefit from a safety system.

V2V communication technologies allow for vehicle safety systems that are capable of detecting potential vehicle collisions that are not possible using camera-based systems. In many traffic scenarios, camera-based systems are not able to detect the presence of another vehicle that can potentially cause a collision, let alone determine the other vehicle's heading, speed, or operational status. Examples of such scenarios are: a host vehicle is approaching a vehicle stopped in the roadway but not visible due to obstructions, a lane change that encroaches on the travel lane of other vehicles that are not yet in the blind spot, and intersections where a vehicle encroaches onto the lane of another vehicle, but is in a blind spot, or an intersection without a traffic signal.

A number of vehicle safety technologies have been developed based on V2V technologies, including vehicle positioning algorithms, and time to collision detection. Zhang, Cao, Bao, and Tan [2]proposed a collision warning framework for connected autonomous vehicles. Given the remote vehicle's basic safety message (BSM) data, the relative position, distance, and speed are computed (with respect to the host vehicle). A Kalman filter is used to predict the position of the remote vehicle and then used to compute a time-to-collision (TTC). If the TTC is within specified thresholds, a warning is given to the driver. Test results showed that the system works on both straight and curved roads, and that latitude error is generally larger than longitude error.

Patra, *et al.* [3] used both position data (GPS) and image data to determine the relative position of a remote vehicle with respect to the host vehicle. Both the GPS and image data were captured with a smart phone. After computing appropriate relative angles, the authors proposed a *same direction test* and a *same lane test*. Experimental results showed that the same direction test was more reliable than the same lane test. The authors concluded that the image processing approach provided better results than the GPS-based method, though at the price of requiring significantly more computational power.

Barrios and Motai [4] noted that simple Kalman filter-based prediction and estimation methods work fine for straight roads. But in the case of curves, the predictions are often erroneous and could fall outside of the road boundaries. To improve the accuracy of the prediction, the authors proposed a system which involved four different Kalman filter models. A different model was developed for each of the following different vehicle conditions:

- The vehicle is not moving,
- The vehicle is traveling at a constant velocity
- The vehicle is traveling with constant acceleration
- The vehicle traveling is traveling with a constant jerk (i.e., a constant change in acceleration).

In addition, the authors used map (GIS) data to constrain position predictions to stay on the road. Experimental results

showed the proposed system worked well, but with prediction time of 3s, twice the average reaction time of a human: 1.5s.

Bhawiyuga, Nguyen, and Jeong [5] pointed out that a key component in a V2V communication-based vehicular safety application is to find a method to accurately estimate vehicle positions and overcome the position bias and random position errors inherent in low-cost GPS devices used in the automotive industry. The authors proposed a vehicle positioning algorithm based on V2V communications and a radar sensor used in the host vehicle. The algorithm constructs two polygons of position estimates: a GPS polygon and a sensing polygon. The GPS polygon is formed by connecting the GPS measures for remote vehicles provided by V2V communications. The sensing polygon is formed by connecting the relative locations of remote vehicles provided by the radar sensor mounted on the host vehicle. The position of the host vehicle is adjusted by the difference between the mass center of the GPS polygon and that of the sensing polygon.

Cho and Kim [6] presented an algorithm to estimate the degree of risk at an intersection using a time-to-intersection (TTI) measure based on V2V communication. The algorithm first calculates the distance from the host vehicle to the intersection, as well as the distances of each remote vehicle to the same intersection. The time-to-collision for each vehicle is obtained by dividing the distance to the intersection by the vehicle speed. The degree of collision risk at an intersection can be determined through monitoring the change in the absolute value of the difference between the TTI of the host vehicle and the TTI of each remote vehicle. The algorithm was verified by applying it to a real collision detection system, and the results showed that the cooperative intersection collision detection system has better accuracy than a vehicle-to-infrastructure (V2I)-based system.

The prediction of future behavior of ego vehicles and surrounding vehicles is an important but also a challenging problem for applications in ADAS and autonomous vehicles, and in particular, when the prediction horizon is long [10]. Neural networks are popularly used in solving these prediction problems. Gao *et al.* [11] used a deep neural network to learn driver physiological patterns for the prediction of driver intent of making lane changes before the events actually occur. They modeled the driver intent prediction problem in terms of Multivariate Time Series (MTS) learning, and developed a Group-wise Convolutional Neural Network (MTS-GCNN) to learn temporal and spatial features effective for predicting lane changes.

Long short-term memory (LSTM), a particular implementation of Recurrent Neural Network (RNN), is a popular machine learning technique used in systems for vehicle behavior prediction. Zyner *et al.* [12] developed a LSTM system to predict a driver's intended path in unstructured environments, i.e., unsignalized road scenes based on the data collected with a Lidar enabled vehicle.

Xin *et al.* [13] presented an algorithm for long-horizon trajectory prediction of surrounding vehicles using a network of a dual long-short term memory (LSTM). The first LSTM in the system takes the sequential trajectory of a vehicle as input and generates driver intention recognition, which is used as an intermediate indicator by the second LSTM for future trajectory prediction. They demonstrated that the system was capable of automatically learning high-level spatial-temporal features of driver behaviors from naturalistic driving trajectory data, which includes the information such as instantaneous speed, acceleration, longitudinal and lateral positions, vehicle length, and vehicle type.

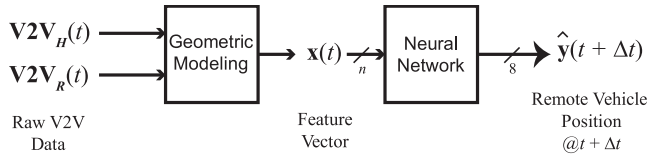


Fig. 2. An intelligent system for remote vehicle position detection and prediction.

Ma *et al.* [14] presented a LSTM-based realtime traffic prediction algorithm called TrafficPredict. TrafficPredict was constructed on a 4D Graph, which could be divided into two main layers: one was the instance layer and the other was the category layer. The instance layer was designed to capture dynamic properties and the interactions between the traffic-agents at a micro level. The category layer was designed to learn the behavior similarities of instances of the same category using a macroscopic view and guide the prediction for instances in turn. The category layer also used a self-attention mechanism to capture the historical movement patterns and highlight the category difference.

III. An Intelligent System for Detecting and Predicting Relative Positions of Remote Vehicles

To determine the driving context, that is, the relative positions of remote vehicles with respect to the host, our approach is to use geometrical modeling and neural network classification. Fig. 2 illustrates the two major components in the proposed intelligent system for detecting and predicting remote vehicle positions based on V2V communication data. The Geometric Modeling component takes as input the V2V data exchanged (at time t) between the host and remote vehicles: $\mathbf{V2V}_H(t)$ and $\mathbf{V2V}_R(t)$, and generates a feature vector, $\mathbf{x}(t)$, that characterizes the relative position of the remote vehicle. The details are presented in Section III-A. The second major component is a neural network that takes the geometric feature vector as input and detects and predicts the relative position of the remote vehicle. The details of the neural network model are described in Section III-B.

A. Geometric Modeling of Relative Positions of Host and Remote Vehicle

We formulate relative vehicle position detection as a classification problem. Fig. 3 shows a schematic diagram of eight remote vehicle position classes with respect to the host vehicle. The host vehicle **H** is shown in the center, and the surrounding squares represent the potential positions of a remote vehicle. Without loss of generality, we focus here on determining the relative position of a single remote vehicle, but the same process can be applied to any number of remote vehicles in the vicinity

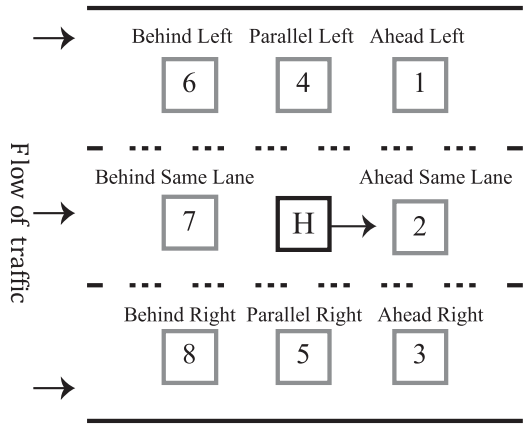


Fig. 3. Illustration of eight position classes of where the remote vehicle could be relative to the host. With respect to the host vehicle, class 1 indicates that the remote vehicle is positioned ahead and one lane to the left, class 2 indicates that the remote vehicle is ahead and in the same lane as the host vehicle, etc.

of, and in communication with, the host vehicle. In applications related to vehicle safety systems, we are interested in establishing the context of the driving environment. In particular, we want to establish whether the remote vehicle is ahead (classes 1, 2, 3), behind (classes 6, 7, 8), or parallel (classes 4 and 5) to the host vehicle, as well as in the same lane (classes 2 and 7), in the left adjacent lane (classes 1, 4, 6) or in the right adjacent lane (classes 3, 5, 8). This type of information is continually monitored and utilized by human drivers or ADAS in many important driving tasks, such as maintaining and adjusting vehicle speed, maintaining lane position, making lane changes, etc. Obviously, this same type of contextual or situational awareness is needed in autonomous vehicle systems, as well.

In real world driving, both the host and the remote vehicles are in motion and the remote vehicle is communicating with the host vehicle through a V2V device, with the following information being shared (denoted $\mathbf{V2V}_H(t)$ and $\mathbf{V2V}_R(t)$ in Fig. 2):

1) GPS Latitude
2) GPS Longitude
3) Heading (radians)
4) Vehicle speed (m/sec)
5) Time stamp (sec)

The remote vehicle positioning algorithms presented below are developed based on the above five attributes of both the host and remote vehicle.

1) Remote Vehicle Position Detection: In this section, we present a geometric model on relative positions between the host and remote vehicle.

Let $\tilde{\mathbf{H}}_{GPS}(t) = (\tilde{\varphi}_H(t), \tilde{\lambda}_H(t))$, denote the position of the host vehicle at time t and $\tilde{\mathbf{H}}_{GPS}(t-1)$ denote its position at time $t-1$, where $\tilde{\varphi}_H(t)$ is the latitude and $\tilde{\lambda}_H(t)$ is the longitude. The tilde is used to indicate raw coordinate data (before normalization). Similarly, we denote the temporal positions of the remote vehicle as $\tilde{\mathbf{R}}_{GPS}(t) = (\tilde{\varphi}_R(t), \tilde{\lambda}_R(t))$ and $\tilde{\mathbf{R}}_{GPS}(t-1)$, respectively. The heading of the host vehicle is denoted $\tilde{\theta}_H(t)$ and the heading of the remote vehicle $\tilde{\theta}_R(t)$.

Given these raw GPS coordinates, we compute some coordinate transformations so that we can work with Euclidean coordinates, as well as normalize the host motion vector so that the host is always traveling due east and the host's position at time $t = 0$ is at the origin. We first perform a UTM transformation on the GPS latitude and longitude to obtain Euclidean coordinates, resulting in: $\tilde{\mathbf{H}}(t) = (\tilde{x}_H(t), \tilde{y}_H(t))$ for the host and $\tilde{\mathbf{R}}(t) = (\tilde{x}_R(t), \tilde{y}_R(t))$ for the remote vehicle, as well as corresponding coordinates at time $(t-1)$: $\tilde{\mathbf{H}}(t-1)$ and $\tilde{\mathbf{R}}(t-1)$, as shown in Fig. 4(a). The vectors are translated so that $\tilde{\mathbf{H}}(t)$ is located at the origin, as shown in Fig. 4(b). Note that we assume that the communicating devices are positioned at the center of each vehicle, so it is the center of the host vehicle at time *t* that is at the origin. Finally, the vectors are rotated by $\tilde{\theta}_H(t)$ so that the host vehicle at time *t* is traveling due east (heading: 0 degrees), as shown in Fig. 4(c):

$$\begin{bmatrix} x_R(t) \\ y_R(t) \end{bmatrix} = \begin{bmatrix} \cos\tilde{\theta}_H(t) & -\sin\tilde{\theta}_H(t) \\ \sin\tilde{\theta}_H(t) & \cos\tilde{\theta}_H(t) \end{bmatrix} \begin{bmatrix} \tilde{x}_R(t) - \tilde{x}_H(t) \\ \tilde{y}_R(t) - \tilde{y}_H(t) \end{bmatrix} \tag{1}$$

The final normalized coordinates of the remote vehicle are denoted $\mathbf{R}(t) = (x_R(t), y_R(t))$.

The relative angle $\theta_{HR}(t)$ between the host and remote vehicle at time *t* can be computed as:

$$\theta_{HR}(t) = \operatorname{atan2}\left(\frac{y_R(t)}{x_R(t)}\right) \tag{2}$$

The distance *d* between the host and remote vehicles is then calculated as:

$$d = sqrtx_R^2(t) + y_R^2(t) \tag{3}$$

In addition, the perpendicular distance $d_\perp$ between the host and remote vehicles (see Fig. 4(c)) at time *t* is computed as:

$$d_\perp = d\ |\sin\theta_{HR}| = |y_R(t)| \tag{4}$$

Using the values of $d_\perp$ and θ_{HR}, we can determine the relative position of the remote vehicle with respect to the host. $d_\perp$ is used to determine if the host and the remote vehicles are in the same lane or an adjacent lane, and θ_{HR} is used to determine if the remote vehicle is ahead, behind, or parallel to the host vehicle.

Let the width of the lane be denoted W_L, and the width of the vehicle be denoted W_V. For simplicity, we assume the remote and host vehicles have the same width. Fig. 5 shows an analysis of $d_\perp$. Fig. 5(a) shows the case where the host and remote vehicles are fully in the same lane (SL) and the perpendicular distance is maximized:

$$d_\perp^{\text{Max, } SL} = W_L - W_V \tag{5}$$

And Fig. 5(b) shows the case where the two vehicles are in adjacent lanes (AL). Of interest here is the smallest perpendicular distance:

$$d_\perp^{\text{Min},AL} = W_V \tag{6}$$

To determine whether the host and remote vehicles are in the same or different lanes, we will choose a threshold $T_{d_\perp}$ somewhere between the above two limits:

$$d_\perp^{\text{Max},SL} \le T_{d_\perp} \le d_\perp^{\text{Min},AL} \tag{7}$$

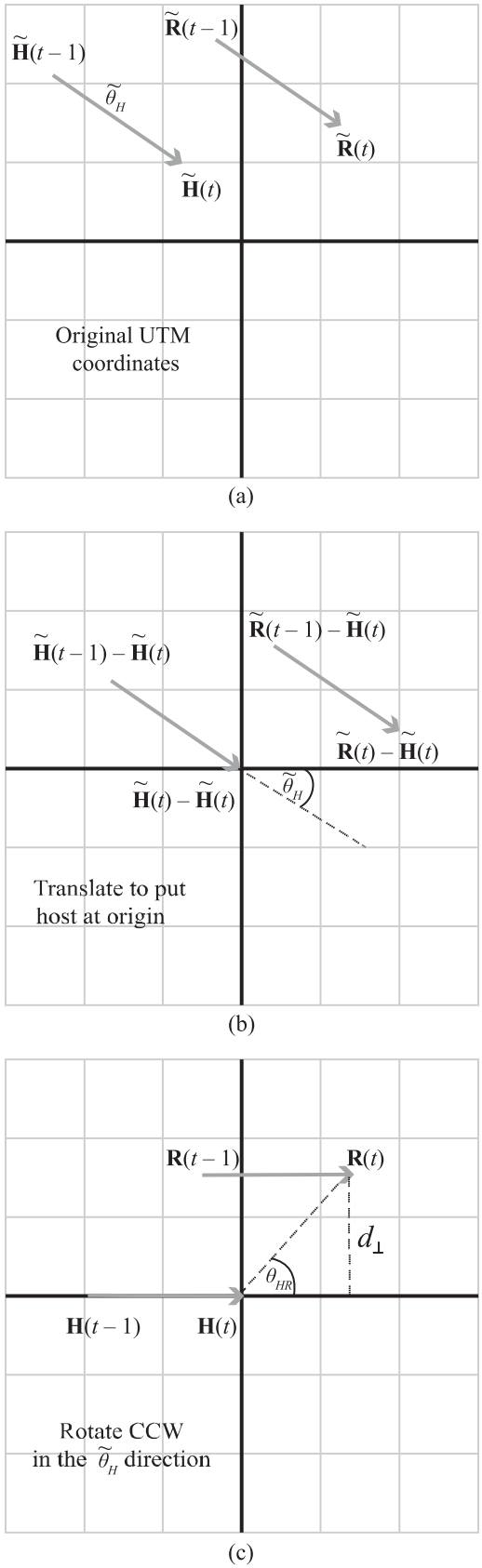


Fig. 4. Coordinate transformation host and remote vehicle positions. (a) The original vehicle coordinates (UTM). (b) Translation so that the host at time *t* is at the origin. (c) After rotation, the host is heading due east; the angle between the host and remote vehicle, as well as perpendicular distance is then calculated.

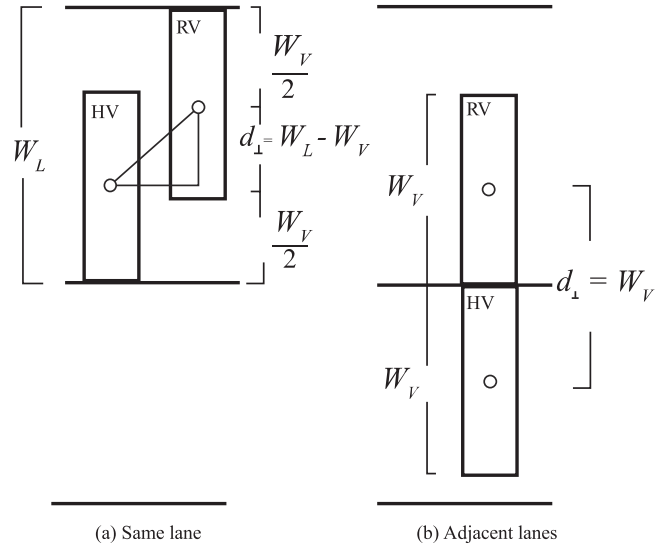


Fig. 5. Illustration of attributes used to decide if two vehicles are in the same or adjacent lanes.

So given the perpendicular distance $d_\perp$ between the host and remote vehicles, we determine whether they are in the same or different lanes as follows:

$$d_\perp \leq T_{d_\perp} : \text{Same lane}$$

$$d_\perp > T_{d_\perp} : \text{Different lanes.}$$

For example, a typical highway lane width is 3.7 m; city roads, though, are narrower: 2.7 – 3.6 m [15]. Since most of our V2V data was collected on city roads, we will assume a lane width: $W_L = 3$ m. A typical vehicle width is $W_V = 2$ m [16]. Using these values: $d_\perp^{\text{Max},SL} = 1$ m and $d_\perp^{\text{Min},AL} = 2$ m; hence, a suitable threshold would be in the range:

$$1 \leq T_{d_\perp} \leq 2 \tag{8}$$

Once we determine whether the host and remote vehicles are in the same or different lanes, the next step is to determine the actual relative position (see Fig. 3) by examining the relative angle θ_{HR} between the host and remote vehicles. If the two vehicles are in the same lane, there are two possibilities: class 2 (ahead) and class 4 (behind). The quadrant of θ_{HR} is used to distinguish between the two: class 2 (quadrants 1 or 4) and class 7 (quadrants 2 and 3). If the host and remote vehicles are in different lanes, then an angle threshold is used to determine whether the two vehicles are driving parallel (classes 4 and 5). As shown in Fig. 6, we used the following classification rules:

$$65^\circ \leq \theta_{HR} \leq 115^\circ : \text{Class 4 (parallel left)}$$

$$-115^\circ \leq \theta_{HR} \leq -65^\circ : \text{Class 5 (parallel right)}$$

The remaining classes can be detected based on the quadrant of θ_{HR}.

In summary, we present the following algorithm, *Geometric Modeling Algorithm*, for classifying the relative positions of a remote vehicle with respect to a host vehicle.

Geometric Modeling Algorithm

1) Use equations (1)–(3) to calculate $\mathbf{R}(t)$, θ_{HR}, and $d_\perp$

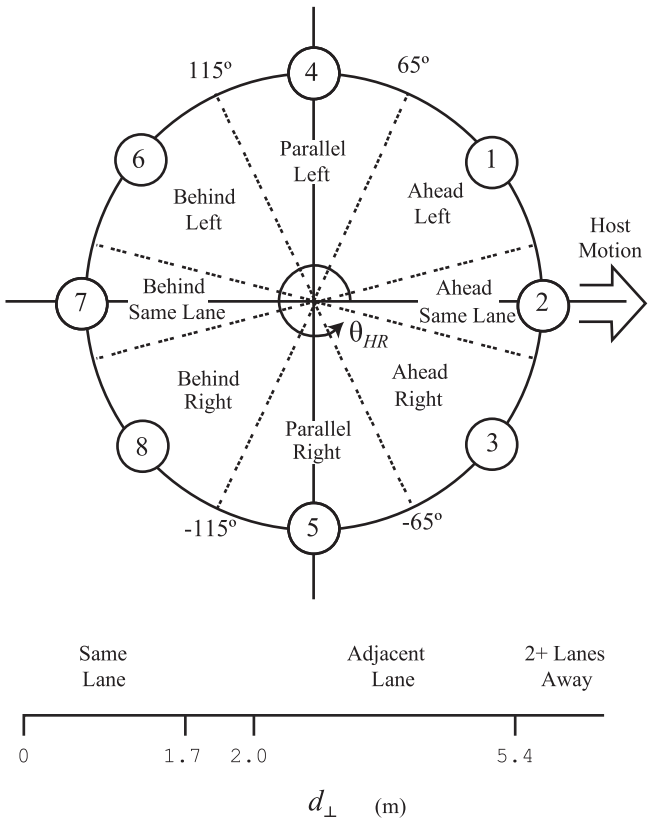


Fig. 6. An illustration of the positional information about the remote vehicle that can be obtained from two key parameters, θ_{HR} and $d_{\perp}$.

2) If $d_{\perp} \leq T_{d_{\perp}}$, the two vehicles are in the same lane. Use θ_{HR} to determine if the remote vehicle is ahead or behind the host.
3) If $d_{\perp} > T_{d_{\perp}}$, the two vehicles are in adjacent lanes. Use θ_{HR} and Fig. 6to determine the output class.

2) Remote Position Prediction: The geometric modeling features can also be used to predict the relative position of a remote vehicle at time $t + \Delta t$ using a simple dead-reckoning approach [17]. Given the host vehicle's position $(x_H(t),\ y_H(t))$, velocity v_H, and heading θ_H at time *t*, the predicted position at time $t + \Delta t$ is simply the present position plus the expected displacement:

$$x_H\left(t + \Delta t\right) = x_H\left(t\right) + \Delta t \cdot v_H \cos\left(\theta_H\right) \tag{9}$$

$$y_H\left(t + \Delta t\right) = y_H\left(t\right) + \Delta t \cdot v_H \sin\left(\theta_H\right) \tag{10}$$

And similarly, for the remote vehicle, we have:

$$x_R\left(t + \Delta t\right) = x_R\left(t\right) + \Delta t \cdot v_R \cos\left(\theta_R\right) \tag{11}$$

$$y_R\left(t + \Delta t\right) = y_R\left(t\right) + \Delta t \cdot v_R \sin\left(\theta_R\right) \tag{12}$$

To determine the predicted relative position at time $t + \Delta t$, we simply use equations 9 – 12 to compute the future positions of the host and remote vehicle along with the current heading of the host vehicle $\theta_H(t)$ and apply the above geometric modeling algorithm. However, the assumption here is that each vehicle travels in a straight line with constant velocity, which is not true for any reasonable travel distances, varying traffic conditions, varying speeds, etc.

As in the discussion above, geometric modeling of the relative positions of a remote and host vehicle in the context of 8 classes depends of a number of parameters and thresholds, such as the perpendicular distance threshold used in the classifications of the same lane or different lane, the thresholds used in mapping the relative heading angle between the host and remote vehicle to one of the eight positioning class (Fig. 6).

In order to develop a classification system that is supported by geometric modeling but also can tolerate variances in these parameters, we propose to use a neural network that takes the key geometric features that characterize the relative position between a host and remote vehicle as input, and learn the implicit relationship between the geometric feature space and the 8 relative position classes of a remote and a host vehicle. from real world V2V data. The three key geometric features we identified are *d*, $d_{\perp}$, and θ_{HR}, which represent respectively the distance, perpendicular distance, and the angle between the host and remote vehicles; see Fig. 4(c). We will use these as the primary features for the neural network.

B. Neural Network Models for Detection and Prediction of Relative Position of a Remote Vehicle

In this section we present the neural networks used in the proposed system: Geo+NN. Neural networks are non-parametric models that can learn from real-world data to implicitly detect complex nonlinear relationships between input variables and output classes, and to tolerate data noise, i.e., small changes in input do not normally cause a large change in output.

In this section we will present two types of neural networks, one for the detection of remote vehicle relative positions, and another for the prediction of remote vehicle positions in the near future.

1) Remote Position Detection: In this subsection we present neural network-based models to detect remote vehicle positions from V2V data of both the host and remote vehicles. Several feature vectors are explored.

Fig. 7 shows an architecture of an $n \times J \times K$ neural network which maps an *n*-dimensional feature vector **x** to a *K*-dimensional output **y** using a layer of *J* hidden neurons. Remote vehicle position detection is modeled as a multi-class classification problem [18]–[19]. We will use a one-hot representation for the output **y**, and since we define eight classes of remote vehicle positions, the output layer has $K = 8$ nodes.

In addition to the three primary features, we explored the effectiveness of augmenting the primary features, as well as the positions and velocities of both the remote and host vehicles. In summary, the following three different feature vectors to the neural network are investigated:

1) *NN 3-Features*: three primary features are used as input to the neural network: d, $d_{\perp}$, and θ_{HR},
2) *NN 9-Features*: three primary features plus the position vectors $\mathbf{H}(t-1)$, $\mathbf{R}(t)$, and $\mathbf{R}(t-1)$, each of which is a vector of two coordinates. Since $\boldsymbol{H}\ (t) = \ 0$, it is not included here.

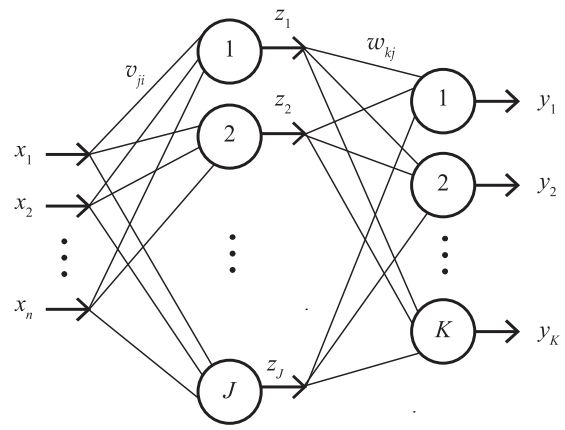


Fig. 7. Architecture of a feedforward neural network with input features extracted from V2V communication data. One-hot encoding is used for the output layer with *K* different classes.

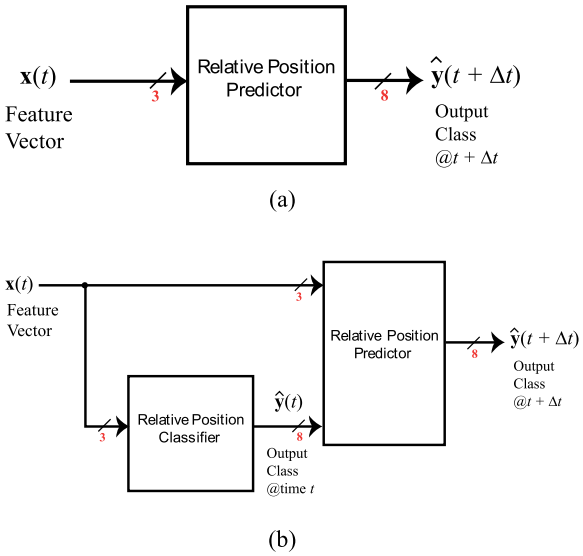


Fig. 8. Three neural network architectures for predicting the remote vehicle's relative positions in the near future. (a) A basic feedforward neural network trained to output but the remote vehicle position's relative position at time $t + \Delta t$. (b) the relative position detected at time *t* are added to the system.

3) *NN 11-Features*: includes the features used in NN 9-Features model, plus the speed of the host and remote vehicles: $v_H(t)$ and $v_R(t)$, respectvely.

2) Remote Position Prediction: The feedforward neural network structure shown in Fig. 7 can also be trained to predict remote vehicle positions at time $t + \Delta t$, where $\Delta t > 0$. The remote vehicle position prediction problem is modeled as follows. At time t, we obtain the feature vector $\mathbf{x}$, and train the neural network to output $\mathbf{y}(t + \Delta t)$, i.e., the remote vehicle's position at $t + \Delta t$. Since the V2V data are in time series, the target value at time $t + \Delta t$ is readily available at the training stage. The look-ahead time Δt is a parameter of the model that can be adjusted based on the required look-ahead time. In general, the prediction accuracy decreases as Δt increases.

Fig. 8 shows two different neural network system architectures we investigated for predicting relative positions of the remote vehicle at time $t + \Delta t$. Fig. 8(a) has the same architecture as Fig. 7, except that it is trained to generate a predicted output. The three different neural network features used in the detection neural networks, NN 3-Features, NN 9-Features, and NN 11-Features can be used with this architecture. In addition, we also investigated another set of features for predicting remote vehicle positions: *NN 5-Features*, which contains the three primary features, as well as $v_H(t)$ and $v_R(t)$.

Another neural architecture illustrated in Fig. 8(b) has the feature vector consisting of the 3 primary features and the detection results generated by the detection neural network system illustrated in Fig. 7. In this case, the prediction network has 8 additional inputs, i.e., the output generated by the relative position detection subsystem as an estimate of the relative position of the remote vehicle at time *t*. We will refer to this model as *NN 3-Features* + $\hat{\boldsymbol{y}}(\boldsymbol{t})$.

In real-world driving conditions, traffic accidents can happen in a fraction of a second; hence, we will mostly consider very short prediction times: $\Delta t \leq 1$s.

IV. Experiments

In this section, we evaluate the proposed neural network models and the geometric method used in Geo+NN for detecting and predicting nearby remote vehicle positions using a part of the V2V data collected from naturalistic driving trips by University of Michigan Transportation Research Institute (UMTRI) under the program: *Safety Pilot Model Deployment* (SPMD), sponsored by the Department of Transportation (DOT), USA. The SPMD program collected V2V data from more than 1800 vehicles installed with dedicated short-range communication (DSRC) devices, which operated at 5.9 GHz in a real-world, concentrated in Ann Arbor, Michigan, USA, during a time period of 23 months, starting August, 2012. The basic safety message (BSM) was transmitted by all DSRC devices at 10 HZ on all vehicles with the DSRC devices. The BSM contained information about the transmitting vehicle including that vehicle's GPS position, speed, and heading. All DSRC devices went through a rigorous process of selection and evaluation to make sure the data were reliably generated, transmitted, received and stored properly. The evaluation criteria used for supplier selection include test results from USDOT qualification testing of security, GPS accuracy, etc. [19]. To address the security concerns involving any network, such as malicious devices and attackers that can corrupt or disable part or all of the system, during the data collection, SPMD designed and implemented a SMOC (Security Management Operating Concept) and a SCMS (Security Credential Management System) to prevent malicious and unintentional attacks of DSRC-based devices [19].

A. Analysis of V2V Data Used in This Research

The V2V data used in our experiments were naturalistic driving trips collected by the SPMD from over 1800 unique drivers during two days: June 2-3, 2013. We extracted from BSMs the following data collected for each vehicle trip,

1) Vehicle Id
2) Trip Id
3) GPS Latitude (5 decimal digits)

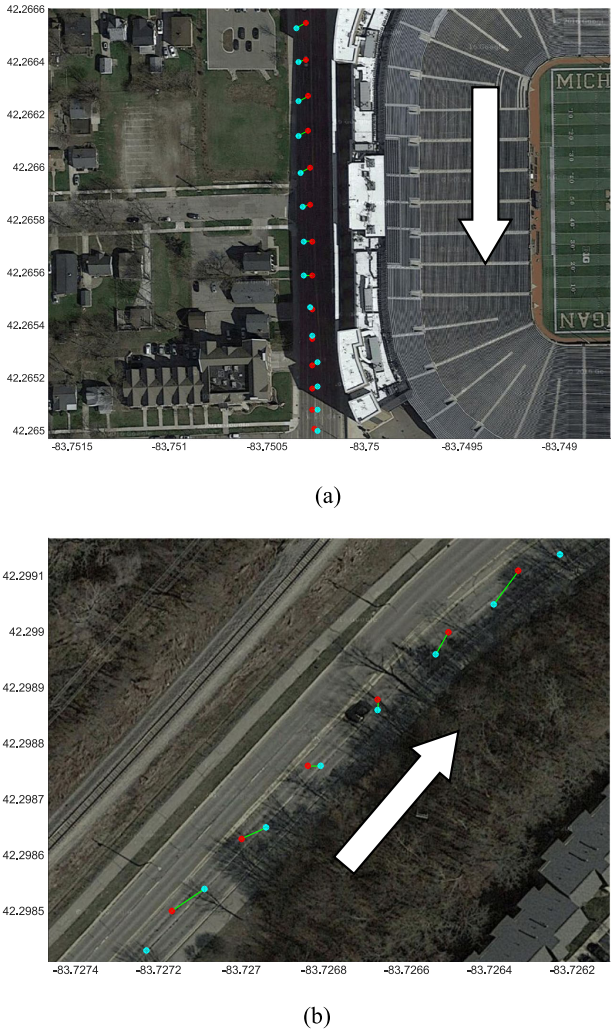


Fig. 9. Illustration of relative positions of host and remote vehicles using two sample trips. Red dots indicate the position of the host vehicle and the cyan dots the position of the remote vehicle. The direction of travel is indicated by the white arrow.

4) GPS Longitude
5) Time stamp
6) Vehicle speed (m/s)

We developed an algorithm to detect and retrieve the pairs of vehicle trips where the two vehicles equipped with V2V devices were in close proximity at the same time and were exchanging real-time V2V data. In particular, we are interested in the case where the two vehicles are within 10 m of each other. In total, we extracted 46 trip segments, in which both vehicles were within 10 m of each other. The 46 trip segments contain both city and highway driving on both straight and curved roads.

Fig. 9 shows a portion of the two such trip segments. The positions of the host vehicle are shown as red circles and the positions of the remote vehicle as cyan circles. The direction of travel is indicated by the white arrow. Note for the display purposes, we sub-sampled the trip data using a sample rate of 1 HZ.

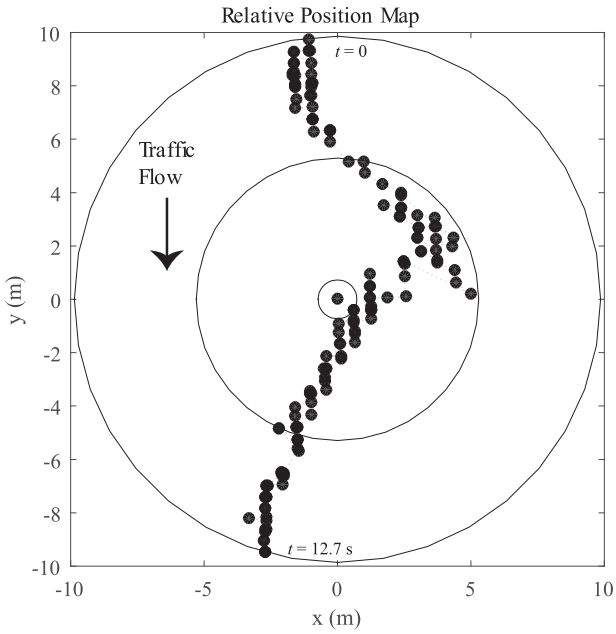
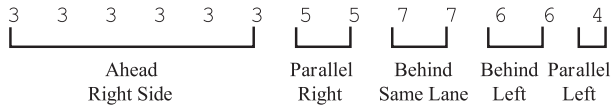
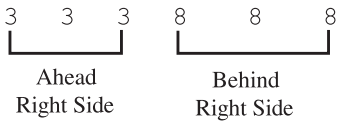


Fig. 10. A relative position map showing the position of the remote vehicle (blue dots) relative to the host vehicle (red dot). The position of the host vehicle is normalized to the origin. During this trip, the remote vehicle starts in class 1, traverses several other classes, and ends up in class 6.

In Fig. 9(a), the remote vehicle is initially ahead of the host, but then falls behind. The sequence of relative positions to the host vehicle is:



Another example is shown in 9(b) where the host vehicle passes the remote vehicle on the left with the following sequence of states:



Another way to visualize the data is to use a relative position map, where the host vehicle is fixed at the center of the map, and the remote vehicle coordinates are marked with respect to the position of the host [2]. An example is shown in Fig. 10. Here, the remote vehicle starts out in class 1 (ahead and to the left) and ends up in class 6 (behind and to the left), while traversing through classes 2, 3, 5, and 7.

Various statistics of the extracted trip segments are shown in Figs. 11–13. Fig. 11 shows a histogram of the closest distances between the host and the remote vehicles. There is a spike at a distance of 3 m and the distribution tails off quickly for both smaller and larger distances.

Fig. 12 shows a histogram of the amount of time that the host and remote vehicles are in proximity. The cases where the proximity times are quite short typically are because the host and remote vehicles are driving in opposite directions. Note that

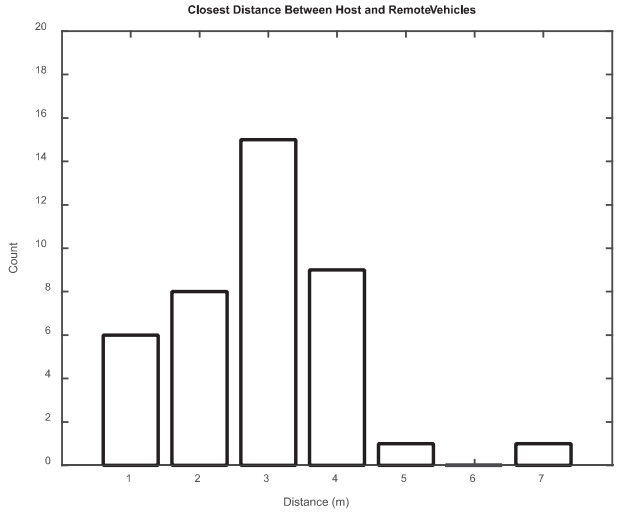


Fig. 11. A histogram of the closest distance between the host and remote vehicle among experimental data

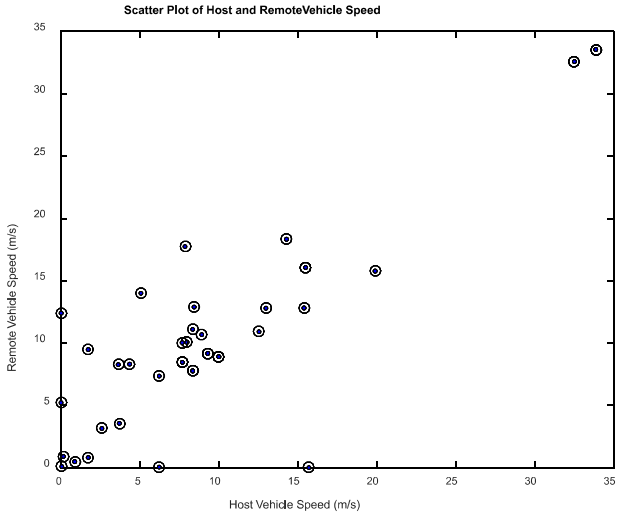


Fig. 13. A scatter plot shows vehicle speed of the host vehicle vs. the remote vehicle.

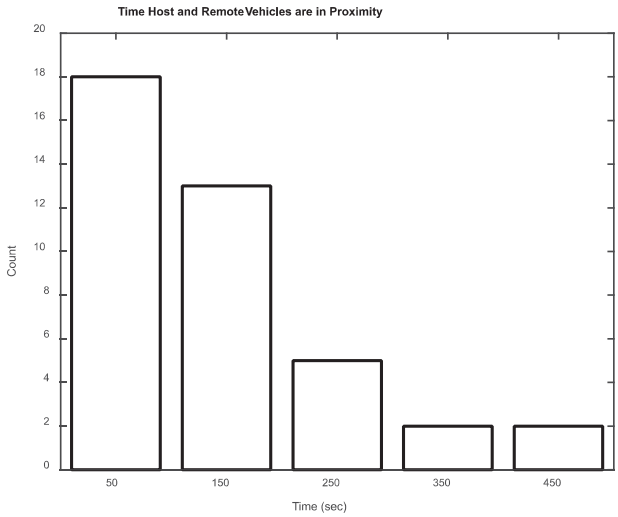


Fig. 12. A histogram of the amount of time that the host vehicle is in proximity to the remote vehicle.

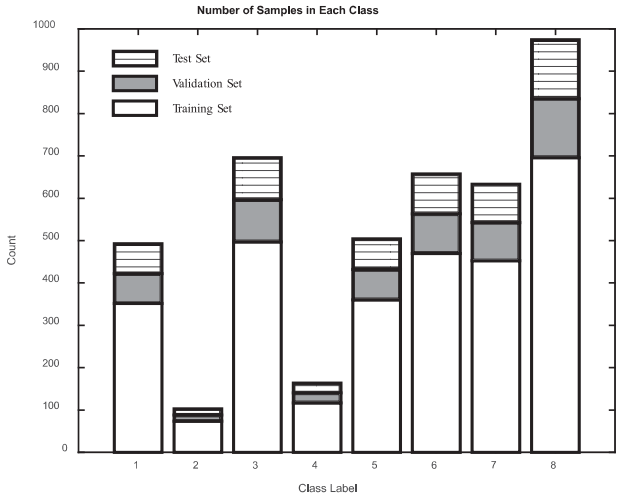


Fig. 14. Distribution of data samples in each class. For random sampling testing, the proportion of data used for the training set is shown in white; the validation set is shown in pink, and the test set is shown in blue.

opposite driving cases are observed in 11 of the 46 trip segments, roughly 24%.

Fig. 13 shows a scatter plot of the speed of the host vehicle vs. the remote vehicle. The speed is taken at the instant when the two vehicles are in closest proximity. The majority of the vehicles had speed less than 18 m/sec ($\sim$40 mph) and hence indicate driving on local city streets.

Recall that there are 8 output classes of relative positions (Fig. 3). A histogram of the amount of data (number of samples) available in each of the 8 output classes is shown in Fig. 14. Clearly, this is an unbalanced data set, with class 8 having the most samples and classes 2 and 4 having relatively few.

B. Construction of Training and Test Sets

To generate ground truth for this V2V data, we used the following manual labeling process. For trip data, $\{\mathbf{H}(t),\ \mathbf{R}(t)\}$, where $\mathbf{H}(t)$ and $\mathbf{R}(t)$ are the host and remote GPS coordinates at time t, respectively, and t varies over all the time steps of the trip. $\mathbf{H}(t)$ and $\mathbf{R}(t)$ were projected onto the Google map and were animated as simultaneous points moving on the road map as shown in Fig. 9. Well-trained human observers labeled the relative position of the two vehicles in eight different classes.

In order to properly evaluate the proposed detection and prediction systems, and in light of fact that the available data are both scarce and unbalanced, we used the following three strategies to create different training and test data sets for evaluation classification systems

1) Partition data into training, validation, and test sets through stratified random sampling,
2) Leave-one-out (complete trip) training and test method, and
3) Four selected complete trips for test data and the rest for training data.

TABLE I
STATISTICS OF THE TEST SET CONTAINING 4 COMPLETE TRIPS

Test Trip	Host Speed (m/sec)	Remote Speed (m/sec)	Closest Distance (m)	Output Classes	Traveling Direction
1	8.92	10.67	3.45	1, 4, 6	Opposite
2	4.33	8.31	4.23	1, 4, 6	Opposite
3	5.06	14.00	4.83	3, 5, 8	Same
4	14.28	18.33	3.54	1, 4, 6	Same

In the case of random sampling, the entire data set of samples was partitioned to 3 parts: a training set, validation set, and test set (chosen at random over all available test samples). The percentage of data in each set was 70%, 15%, and 15%, respectively. To handle the unbalanced nature of the data, we used stratified sampling of the available data. The color-coded regions in Fig. 14give a representation of the relative sizes of the three sets of data: white indicates training data, shaded indicates validation data, and striped indicates test data.

Ideally, the test set should consist of complete trips, rather than just random (and disjoint) samples of trip points. Since we have a small number of total trip segments (46 in total), we used a leave-one-out strategy to test the proposed systems. In this case, 45 complete trip segments are used for training, and the one remaining trip segment is used for testing. The results for these experiments are averaged over all 46 test sets. Note that the leave-one-out strategy is a special case of K-fold testing, where $K = N_T - 1$, and N_T is the number of test samples.

Finally, we selected four trip segments from the available 46 trip segments to be the test set, and all other trip segments were used for training. The four test trip segments were chosen such that there was a good representation of each output class. Statistics of the four test trip segments are shown in Table I. In total, there are 92 samples extracted from the four test trips.

Note that in evaluating the system's prediction accuracy, since our prediction system needs a window of 10 samples to make a prediction at 1 second look-ahead, trips less than 1.5s are removed. In the end, 32 trip segments were used for evaluating prediction accuracy.

C. Relative Position Detection

The performance of the neural network-based detection and prediction systems described in Section IV.A are presented next. Accuracy is measured as follows:

$$\text{Accuracy} = \frac{\#\ \text{Samples Correctly Classified}}{\text{Total}\#\text{Samples Tested}}$$

The accuracy of the relative position detection on the test data as a function of number of hidden nodes by the neural networks using random sampling is shown in Fig. 15, the leave-one-out method in Fig. 16, and the four complete test trips in Fig. 17.

The results show that the neural network systems perform well when at least 10 hidden nodes are used and that the 3-feature vector-based systems outperform the 9-feature and 11-feature systems in all three test scenarios. However, if we look at

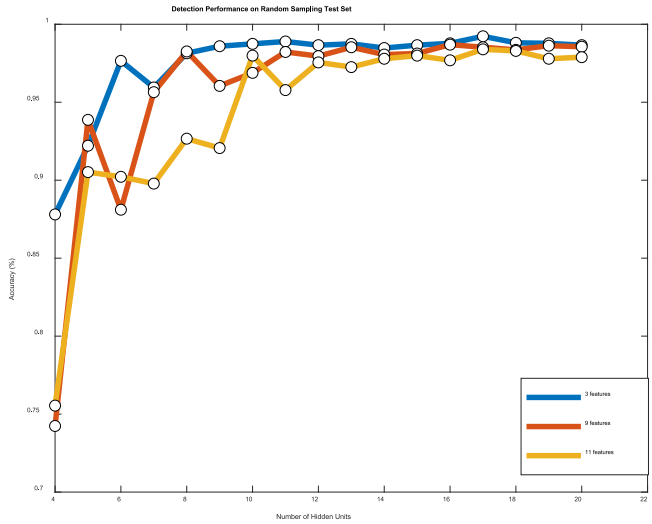


Fig. 15. Vehicle relative position detection accuracies of the neural networks using 3 features (blue), 9-features (red), and 11-features (yellow). The random sampled training and test sets were used, and the three curves represent performance on the test data sets.

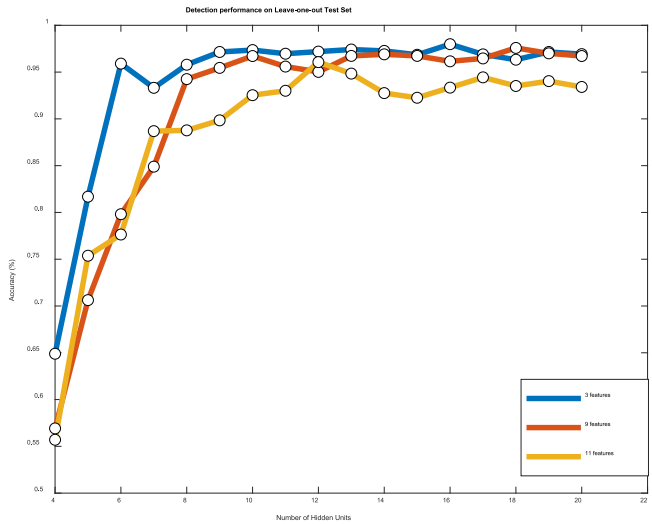


Fig. 16 Vehicle relative position detection accuracy of the neural networks using 3-features (blue), 9-features (red), and 11-features (yellow) obtained by averaging the performances on the test trips using the strategy of leave-one-out.

best performance, the 3-feature and 9-feature networks achieved similar results. These experimental results show that the three primary features work very well on this set of data.

For the neural networks of NN 3-features, in the case of random sampling test, the best result was achieved with 17 hidden units. In the case of leave-one-out test, the best result was achieved by a neural network with 16 hidden nodes, and in the case of using the four complete test trips, the best neural network was the one with 15 hidden nodes. The performances of these three systems are all above 99% in accuracy. These experimental results demonstrate that the three primary features derived from our geometric modeling work very well on this set of data.

The confusion matrices in Tables II–IV show the typical types of errors made by the neural network systems trained and tested

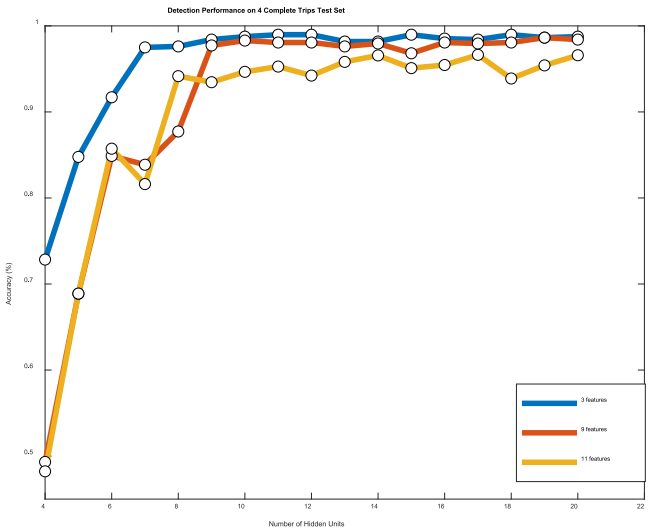


Fig. 17. Vehicle relative position detection accuracy of the neural networks using 3-features (blue), 9-features (red), and 11-features (yellow) obtained by averaging their performances on the 4 test trips.

TABLE II
CONFUSION MATRIX FOR RELATIVE POSITION DETECTION USING 3 FEATURES AND RANDOM SAMPLING TEST SET

	1	2	3	4	5	6	7	8
1	67	0	0	1	0	0	0	0
2	0	15	0	0	0	0	0	0
3	0	2	96	0	0	0	0	0
4	0	0	0	23	0	0	0	0
5	0	0	1	0	71	0	0	0
6	2	0	0	0	0	90	0	0
7	0	0	0	0	0	1	89	0
8	0	0	2	0	0	0	1	136

TABLE III
CONFUSION MATRIX FOR RELATIVE POSITION DETECTION USING 9 FEATURES AND RANDOM SAMPLING TEST SET

	1	2	3	4	5	6	7	8
1	68	0	0	0	0	0	0	0
2	0	14	0	1	0	0	0	0
3	0	0	96	0	2	0	0	0
4	1	0	0	20	1	1	0	0
5	0	0	1	0	70	0	0	1
6	1	0	0	0	0	91	0	0
7	0	0	0	0	0	0	90	0
8	0	0	3	0	0	0	0	136

using the randomly sampled training and test sets. In most cases, the errors are caused by samples in the neighboring classes positioned in the same lane, for example mistaking classes of (4 – 6), (1 – 4), (3 – 5), and (5 – 8). However, there are a few cases of confusion among non-neighboring classes: (2 - 4), (8 - 3), (6 - 1), (4 - 5). These types of misclassifications can arise due to GPS error, such as drop out, or satellite signal blockage from buildings, trees, bridges, etc. A suitable post-processing stage could be designed to filter such impossible state transitions. However, since the results in Tables II–IV show that these errors are relatively rare, and since the GPS-based system will ultimately be augmented with other in-resident sensors such as camera-based system, we did not pursue this line of research (see Section V).

TABLE IV
CONFUSION MATRIX FOR RELATIVE POSITION DETECTION USING 11 FEATURES AND RANDOM SAMPLING TEST SET

	1	2	3	4	5	6	7	8
1	64	0	0	4	0	0	0	0
2	0	15	0	0	0	0	0	0
3	0	0	96	0	1	0	0	1
4	0	0	0	23	0	0	0	0
5	0	0	8	0	61	0	0	3
6	2	0	0	5	1	82	1	1
7	0	0	0	0	0	0	90	0
8	0	0	0	0	2	0	2	135

D. Relative Position Prediction

In this section we present system performance on the problem of remote vehicle relative position prediction. Note that performance is measured in the same way as the previous subsection: the ratio of the number of correctly classified samples to the total number of samples tested. The only difference here is that correct classification is evaluated at time $t + \Delta t$ and not at time t. In the case of the geometric method with dead reckoning, we use Equations 4∼7 to determine the future position of the vehicles and then apply the usual geometric algorithm to map to one of the 8 output classes.

The prediction results generated by neural networks are shown in Figs. 18–20. Fig. 18 shows the accuracies of relative position prediction vs. the number of hidden neurons used in the prediction neural network systems shown in Fig. 8that use 3, 9, and 11 features. In the experiments used to generate the results in Fig. 19, the training and test data are generated using the random sampling rate of 70% vs 30%. The color coding (see legend) represents the amount of look-ahead time in the prediction range of 0.1 s to 0.5 s by an increment of 0.1 s and at 1 s. The results were averaged over 50 trials.

As expected, as the look-ahead time increases, accuracy decreases. When the prediction time is less than 0.4 seconds, all three systems performed similarly well, with accuracies in the range 87%∼93%. With prediction time $\Delta t \geq 0.4$, the neural networks using the 11-features system outperformed the neural networks using 3 and 9-features. At prediction time 1 s, the best performing neural network is the one with 15 hidden neurons and 11 input features, which achieved 85% accuracy, whereas the best neural networks with 3 and 9 features achieved 77% or lower.

Fig. 19 shows prediction accuracy in the case of the test set generated using the leave-one-out strategy. The neural networks

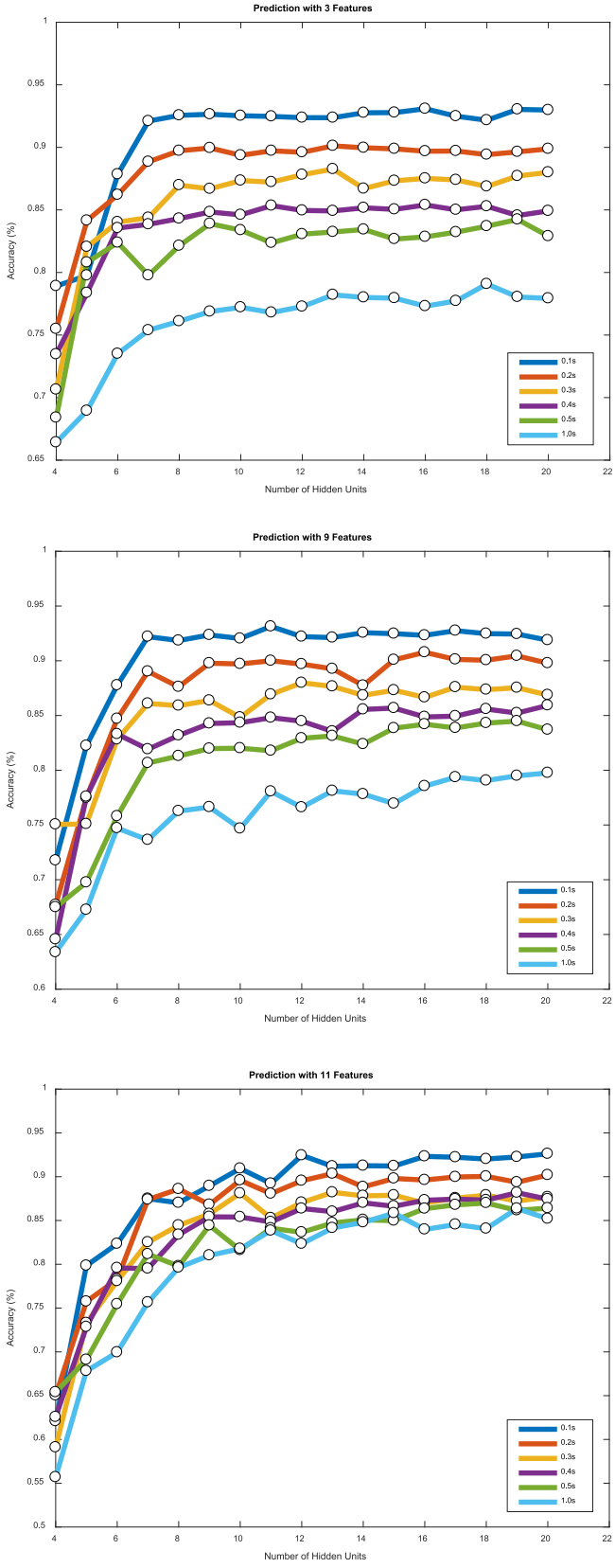


Fig. 18. Relative position prediction accuracy vs. number of hidden neurons using the prediction neural network with 3, 9, and 11 features and with the random sampling of training and test data sets. The legend indicates the amount of look-ahead time from 0.1 seconds to 0.5 seconds and at 1 second.

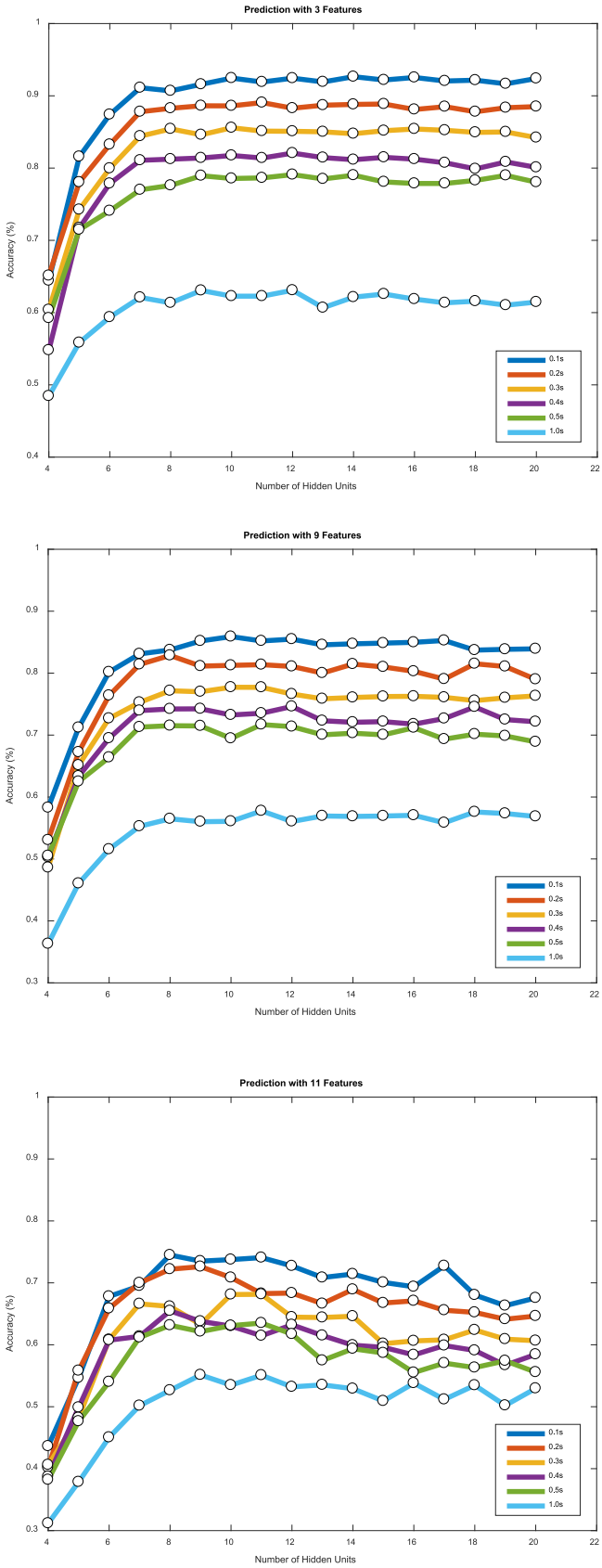


Fig. 19. Relative position prediction accuracies of the neural networks using (a) 3 features (b) 9-features and (c) 11-features with the strategy of leave-one-out. The performances shown in the figure are the average performances on the 32 test trips.

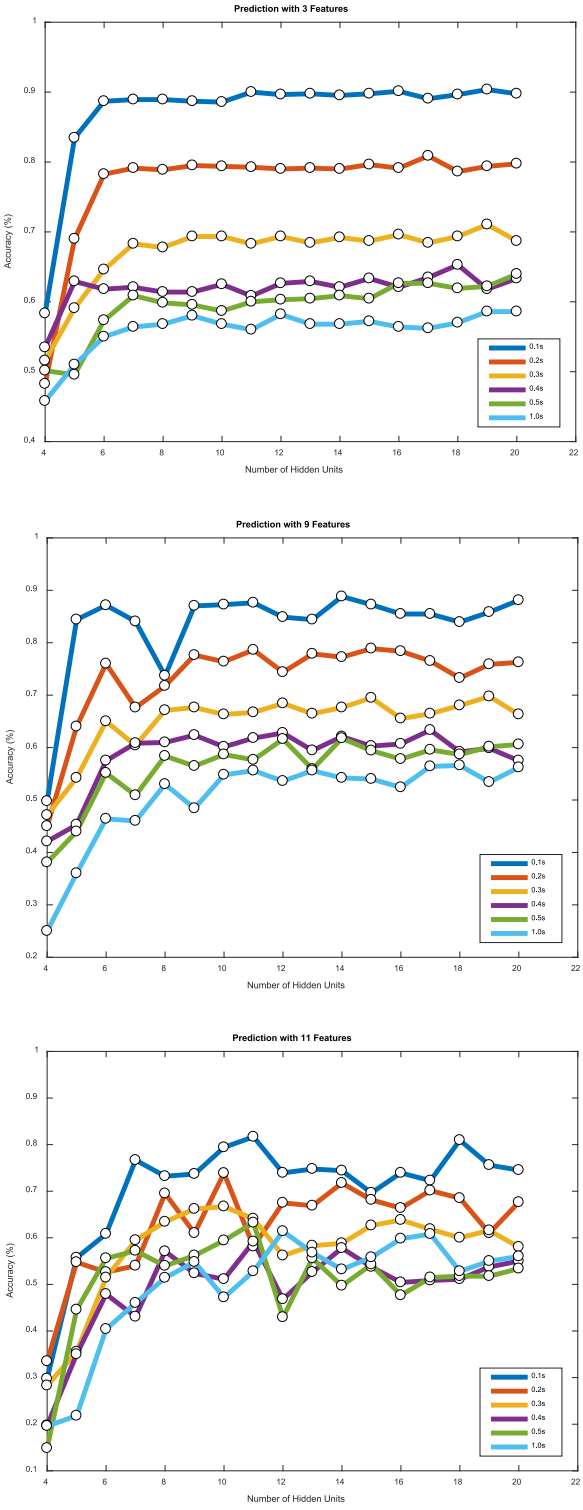


Fig. 20. Relative position prediction accuracy of the neural networks using (a) 3 features (b) 9-features and (c) 11-features on the test set consisting of 4 trips.

that used the three primary features gave the best results at all prediction times.

Fig. 20 shows prediction performances on the test set containing four complete test trips. In this case, when look-ahead of time is short, such as $\Delta t < 0.5$, the 3-feature neural networks gave the best prediction results. However, when the look-ahead of time is 1 second, the neural network of 11 features with 12

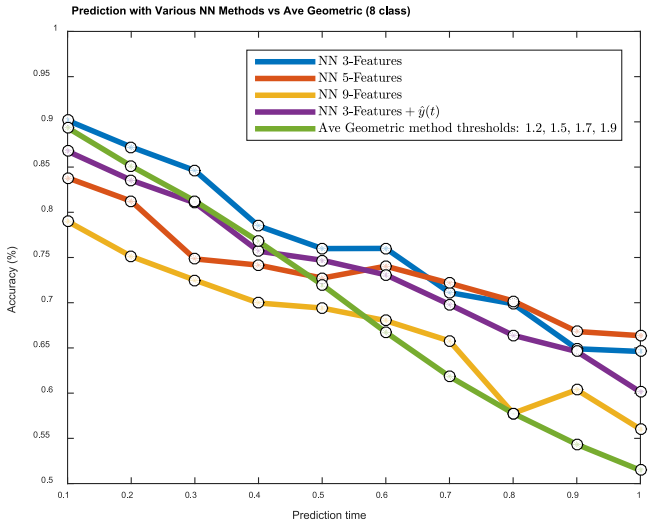


Fig. 21. Prediction results generated using 32 test trips with a leave-one-out strategy. All neural networks used 15 hidden neurons. The geometric method results were an average of the results using thresholds: 1.2, 1.5, 1.7, and 1.9.

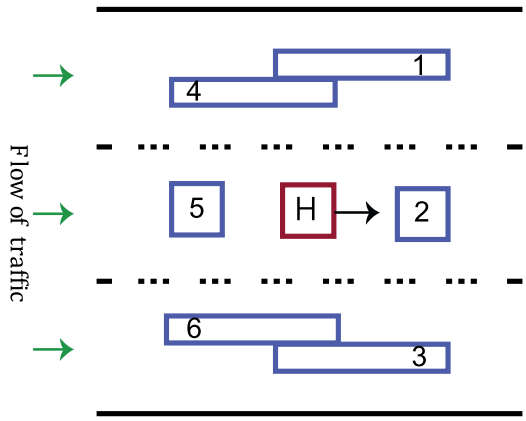


Fig. 22. A reduced set of 6 output classes are obtained by combining neighboring classes (compare with Figure 3).

hidden nodes gave the best prediction results. It is also observed that the neural networks with 11 features gave smaller variations in the prediction accuracies with the increase of look-ahead time. We believe that the 11 feature neural networks could perform better overall prediction with more training data.

Fig. 21 shows a comparison of relative position prediction accuracy vs look-ahead time between neural networks that use various features and the geometric prediction system. For the ease of comparison, all the neural networks in this figure used 15 hidden nodes. For the geometric method, in order to get a more realistic evaluation, we present the performance averaged over the perpendicular thresholds: $T_{d_\perp} = 1.2$, 1.5, 1.7, and 1.9 m. The neural network using 3 primary features gave the best prediction results when $\Delta t < 0.7$ second. When the look-ahead time increases, e.g., $\Delta t > 0.6$ the neural network prediction systems perform substantially better than the averaged results generated by the geometric method.

The discrimination between the parallel and ahead or behind classes, i.e., class pairs (1, 4), (4, 6), (3, 5) and (5, 8) is difficult even for human inspectors. In practice, it is important just to

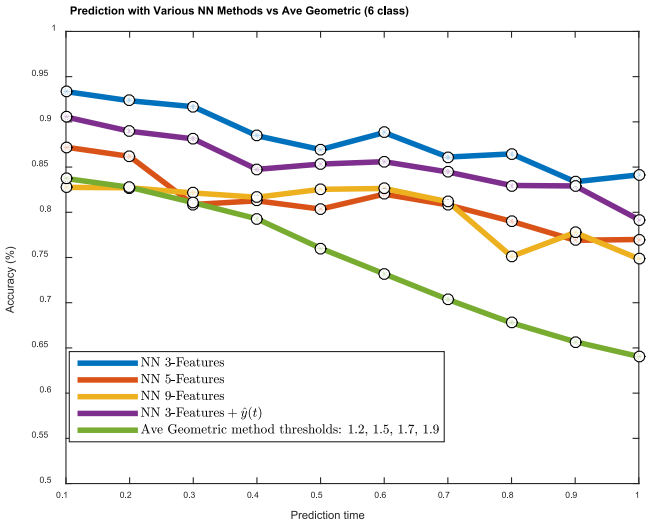


Fig. 23. Prediction of remote vehicle positions in 6 different classes.

know if there is a vehicle ahead or behind the host vehicle in the left or right lane. For this reason, we conducted another set of experiments to evaluate the prediction systems with a reduced set of output classes as shown in Fig. 22.

Fig. 23 shows performances of the prediction results on this reduced set of output classes. The neural network with 3 features gave the best performance over all prediction times. All neural networks gave significantly better prediction results than the geometric method when $\Delta t > 0.3$.

V. Conclusion

A comprehensive pre-crash detection safety system should use multiple vehicle-resident sensors as well as V2V communication devices in order to extend the capabilities of the overall safety functions applicable to a broader range of crash scenarios with improved warning timing and reducing the number of false alarms. The focus of this paper was to investigate the technologies that can be used for detecting and predicting relative positions of remote vehicles in a driving context where only V2V data was available, specifically GPS coordinates.

We presented an intelligent system, Geo+NN, that is based on geometric modeling of the relative positions of a host and a remote vehicle and neural network learning from V2V communication signals, for the detection of remote vehicle positions in the context of 8 different classes distributed in left, right and the same lane of the host vehicle. Based on the geometric modeling, we derived three meta features, i.e., primary features that are effective in detecting the relative position of a remote vehicle based on V2V data. We investigated two different neural network architectures with various additional features extracted from V2V data for remote vehicle position detection and prediction. Even with the use of only V2V, the proposed neural networks produced accurate remote vehicle detection and prediction results. For remote vehicle position detection, the proposed neural networks all reached 99% accuracy. For prediction in 8 position classes, when the look-ahead time is less than 0.7 seconds, the best prediction accuracy is above 75%, and at 1 second is above 70%. For prediction in 6 position classes, when the look-ahead of time is less than 0.7 seconds, the best prediction accuracy is above 90% and 1 second is close to 90%.

These experiments show that the neural network systems that use the three primary features derived from the proposed geometric modeling gave robust performances for relative position detection and prediction. It implies that the neural networks can generalize well on meta features derived from an analytic model.

Another important property of the proposed system is its fast speed of computation. In our experiments, the average computation time of each prediction by the proposed neural network is about 30 microseconds (using Matlab with a MacBook Pro Intel Core i5 2.6 GHz), which is fast enough for real time applications, and much faster than a computer vision based method for vehicle position detection or prediction.

It is important to note that in order to achieve the benefits that V2V communication technology has to offer, onboard V2V communication devices must be ubiquitous among vehicles on the road. At present, though, that is not the case, and there is only one US car model that has V2V capability: Cadillac CTS. However, the US National Highway Traffic Safety Administration (NHTSA) is working with the automotive industry to issue rules that such technology will be required. It is anticipated that about 60% of new vehicles (representing over 62 million vehicles) will have V2V capability by 2023 [20].

One major technical challenge in applying V2V communication to ADAS is that, as with any communication technology, there is potential for misuse, such as eavesdropping and embedded device hacking. For example, a nefarious actor in a remote vehicle could trail the host vehicle and potentially record and later access their location data. Even more concerning, a remote vehicle could send erroneous V2V information to the host and possibly cause a serious accident. These issues have been studied extensively in the cybersecurity research community, and security protocols and new technologies are under development for secured V2V communication [21].

In future work, we will perform additional studies concerning the robustness of the proposed Geo + NN system with respect to various types of GPS errors, communication time delays, and parameter settings. For example, we will study the robustness of the system to variations in the angle thresholds shown in Fig. 6, as well as the robustness to variations in lane and vehicle width used in the Geometric Algorithm. Finally, we will study the performance of a hybrid system consisting of both the Geo + NN system along with an intelligent camera-based system. It is expected that the hybrid system can provide substantial improvement in system performance. With such a hybrid system, we will be able to study error margins with respect to various sizes of remote vehicles.

References

[1] Y. Shen *et al.*, "A method of traffic travel status segmentation based on position trajectories," in *Proc. IEEE 18th Int. Conf. Intell. Transp. Syst.*, 2015, pp. 2877–2882.

[2] R. Zhang, L. Cao, S. Bao, and J. Tan, "A method for connected vehicle trajectory prediction and collision warning algorithm based on V2V communication," *Int. J. Crashworthiness*, vol. 22, no. 1, pp. 15–25, 2017.

[3] S. Patra, C. Calafate, J. Cano, C. de Vera, P. Veeleart, and W. Philips, "Determining the relative position of vehicles considering bidirectional traffic scenarios in VANETs," in *Proc. 2nd Workshop Experiences Des. Implementation Smart Objects*, Oct. 2016, pp. 6–10.
[4] C. Barrios and Y. Motai, "Improving estimation of vehicle's trajectory using the latest global positioning system with kalman filtering," *IEEE Trans. Instrum. Meas.*, vol. 60, no. 12, pp. 3747–3755, Dec. 2011.
[5] A. Bhawiyuga, H. Nguyen, and H. Jeong, "An accurate vehicle positioning algorithm based on vehicle-to-vehicle (V2V) communications," in *Proc. IEEE Int. Conf. Veh. Electron. Saf.*, Istanbul, Turkey, Jul. 2012, pp. 273–278.
[6] H. Cho and B. Kim, "Study on cooperative intersection collision detection system based on vehicle-to-vehicle communication," *Adv. Sci. Technol. Lett.*, vol. 58, pp. 121–124, 2014.
[7] J. Harding *et al.*, "Vehicle-to-vehicle communications: Readiness of V2V technology for application," Nat. Highway Traffic Safety Administration, Washington, DC, USA, Tech. Rep. DOT HS 812 014. Aug. 2014.
[8] M. El-Said, S. Mansour, and V. Bhuse, "DSRC based sensor-pooling protocol for connected vehicles in future smart cities," *Procedia Comput. Sci.*, vol. 140, pp. 70–78, 2018.
[9] "Vehicle safety communications project task 3 final report," *Nat. Highway Traffic Safety*, DOT HS 809 859, 2005. [Online]. Available: https://rosap.ntl.bts.gov/view/dot/3925
[10] S. Mozaffari, O. Y. Al-Jarrah, M. Dianati, P. Jennings, and A. Mouzakitis, "Deep learning-based vehicle behaviour prediction for autonomous driving applications: A review," *IEEE Trans. Intell. Transp. Syst.*, to be published, doi: 10.1109/TITS.2020.3012034.
[11] J. Gao, Y. L. Murphey, and H. Zhu, "Multivariate time series prediction of lane changing behavior using deep neural network," *Appl. Intell.*, vol. 48, pp. 3523–3537, 2018.
[12] A. Zyner, S. Worrall, and E. M. Nebot, "Naturalistic driver intention and path prediction using recurrent neural networks," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1584–1594, Apr. 2020.
[13] L. Xin, P. Wang, C. Chan, J. Chen, S. E. Li, and B. Cheng, "Intention aware long horizon trajectory prediction of surrounding vehicles using dual LSTM networks," in *Proc. 21st Int. Conf. Intell. Transp. Syst.*, Nov. 2018, pp. 1441–1446.
[14] Y. Ma, X. Zhu, S. Zhang, R. Yang, W. Wang, and D. Manocha, "Trafficpredict: Trajectory prediction for heterogeneous traffic-agents," in *Proc. AAAI Conf. Artif. Intell.*, 2019, pp. 6120–6127.
[15] Lane Width. [Online]. Available: https://safety.fhwa.dot.gov/geometric/pubs/mitigationstrategies/chapter3/3_lanewidth.cfm
[16] "What is the Width and Length of the Average Car?," [Online]. Available: https://www.reference.com/vehicles/wide-average-car-67a48253b70cd4c2
[17] P. Corke, *Robotics, Vision, and Control: Fundamental Algorithms in Matlab*. New York, NY, USA; Berlin: Springer Tracts in Advanced Robotics, 2011.
[18] G. Ou and Y. L. Murphey, "Multi-class pattern classification using neural networks," *J. Pattern Recognit.*, vol. 40, no. 1, pp. 4–18, 2007.
[19] Y. Liu, P. Watta, J. Bochen, and Y. L. Murphey, "Vehicle position and context detection using V2V communication with application to pre-crash detection and warning," in *IEEE Symp. Ser. Comput. Intell.,* 2016, pp. 1–7.
[20] J. Koon, "Will vehicle-to-vehicle communication ever take off?," 2019. [Online]. Available: https://www.engineering.com/IOT/ArticleID/18583/Will-Vehicle-to-Vehicle-Communication-Ever-Take-Off.aspx
[21] "Questions and Answers about DOT's Safety Pilot: Model Deployment." [Online]. Available: https://www.nhtsa.gov/sites/nhtsa.dot.gov/files/812171safetypilotmodeldeploydeltestcondrtmrep.pdf

Paul Watta (Member, IEEE) is originally from Detroit, Michigan. He received the bachelor's, master's, and the Ph.D. degrees in electrical engineering from Wayne State University. He is currently an Associate Professor of Electrical and Computer Engineering Department with the University of Michigan-Dearborn.

He teaches courses in computer methods, mobile devices, and computer music. His research interests include areas of machine learning, image processing, and autonomous vehicles.

Ximu Zhang (Student Member, IEEE) received the B.S. degree in electrical engineering and automation from Xi'an Jiaotong University, Xi'an, China, in 2018 and the M.S. degree in computer and information science from the University of Michigan – Dearborn, MI, in 2019. He is currently working toward the Ph.D. degree in electrical engineering with the University of Tennessee - Knoxville, Knoxville, TN, USA.

From 2017 to 2019, he was a Graduate Research Assistant with the College of Engineering and Computer Science, University of Michigan – Dearborn. Since 2019, he has been the Graduate Research Assistant with the CURENT in University of Tennessee – Knoxville.

Yi Lu Murphey (Fellow, IEEE) received the M.S. degree in computer science from Wayne State University, Detroit, Michigan, in 1983 and the Ph.D degree in a major in computer engineering and a minor in control engineering from the University of Michigan, Ann Arbor, Michigan, in 1989. She is currently the Interim Vice Provost for Research, an Associate Dean for Graduate Education and Research, the Associate Dean for Graduate Education and Research, a Professor of the ECE (Electrical and Computer Engineering) Department and the Director of the Intelligent Systems Lab with the University of Michigan-Dearborn. Prior to her current position, she was the Chair of the ECE Department for seven years.

She has authored more than 150 publications in refereed journals and conference proceedings in the areas of machine learning, pattern recognition, computer vision and intelligent systems with applications to intelligent vehicle systems, optimal vehicle power management, data analytics, automated and connected vehicles and robotic vision systems. She has received extensive research funding from the National Science Foundation (NSF), US Army TARDEC, State of Michigan, Ford Motor Company, Toyota Research Institute, Hyundai Motor Company, ZF-TRW and many others. She was also the recipient of HERS_DBL STEM scholarship by the LUCE foundation in 2016. Dr. Murphey is a Distinguished Lecturer for the IEEE Society of Vehicular Technologies.