

Vehicle Position and Context Detection using V2V Communication with Application to Pre-crash Detection and Warning*

Yifu Liu, Paul Watta, *Member, IEEE*, Bochen Jia and Yi Lu Murphey, *Fellow, IEEE*

Abstract— In recent years, there has been growing interest in the design of intelligent transportation systems (ITS) using vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communication technologies. These systems allow vehicles to share GPS-based information, such as latitude, longitude, speed and heading, as well as important vehicle data such as brake events, throttle position, turn signal status, etc. A pre-crash detection and warning system in a host vehicle (i.e. vehicle of interest) needs to accurately determine not only the position of each remote vehicle in its vicinity, but also the *context* of the driving environment because the context can provide important information about whether or not the remote vehicle poses a threat to the host. For example, remote vehicles in the same lane or one lane over typically pose more of a threat than remote vehicles two or more lanes over. In this paper, we propose a real-time algorithm that computes the *relative position* and *driving context* of all remote vehicles within a region of interest of the host vehicle. Experimental results on real-world V2V data show that the proposed method can effectively compute the position and context of remote vehicles in real time.

I. INTRODUCTION

Accurate detection of vehicle position plays an important role in many vehicular safety applications. The most advanced crash detection and avoidance technologies present in vehicles today include a host of on-board sensors, cameras, and radar applications. These technologies may warn drivers of impending danger so that the driver can take corrective action, or may even be able to intervene on the driver's behalf. While pre-crash detection technologies based on these on-board sensors can provide improved vehicle safety, vehicle-to-vehicle (V2V) communications represent a new technology in helping to warn drivers about impending danger. V2V communications use on-board dedicated short-range radio communication (DSRC) devices to transmit messages about a vehicle's speed, heading, brake status, and other information to other vehicles (and receive such information, as well) with range and "line-of-sight" capabilities that exceed current sensor-based vehicle safety systems—in some cases, nearly twice the range [1]. 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 [2].

Our research is focused on developing a pre-crash detection system based on V2V communication technology. In particular, we present our research in vehicle positioning technology. The vehicle that has the pre-crash detection system is referred to as the *host vehicle* (H), and a vehicle which communicates with the host is referred to as a *remote vehicle* (R). Note that, at any given time, there may be several remote vehicles communicating with the host, as shown in Figure 1.

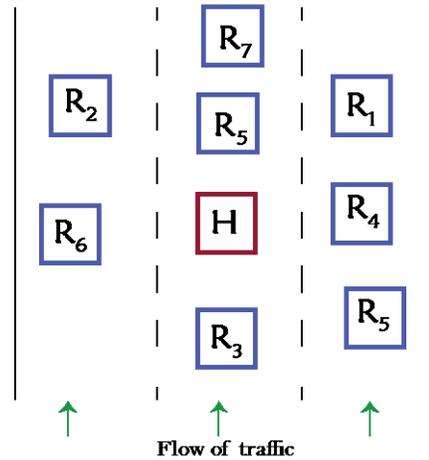


Figure 1. A schematic diagram showing the position of the host vehicle and several remote vehicles.

The design of a pre-crash detection and warning system in a host vehicle requires the determination of not only the position of each remote vehicle in its vicinity, but also the *context* of the driving environment because the context can provide important information about whether or not the remote vehicle poses a threat to the host. For example, with respect to Figure 1, the present context of the host includes the fact that remote vehicle 1 is ahead of the host and one lane over to the right, while remote vehicle 3 is behind the host in the same lane. Remote vehicles 4 and 6 are driving parallel to the host, one on either side. Of particular interest are scenarios of occlusion; for example, in Figure 1, the host vehicle closely follows both remote vehicles 5 and 7. With V2V communication, the host vehicle can be alerted if vehicle 7 comes to an abrupt stop, even though vehicle 7 may not be visible to the host and even though vehicle 5 does not slow down (as one would expect). Hence, this GPS and

*Research supported by a grant from MTC, University of Michigan.

Yifu Liu, Yi Lu Murphey, and Paul Watta are with the University of Michigan-Dearborn, Department of Electrical and Computer Engineering, Dearborn, Michigan, 48128 USA (corresponding author: Yi Lu Murphey, phone: 313-593-0541, e-mail: yilu@boulder.nist.gov).

Bochen Jia is with the University of Michigan-Dearborn, Department of Industrial and Manufacturing Systems., Dearborn, Michigan, 48128 USA, (email: bochenj@umich.edu).

communication-based technology offers the promise of expanding and enhancing the driver's field of view.

In this paper, we propose a real-time algorithm that computes the *relative position* of all remote vehicles with respect to the host, resulting in the determination of the driving context. It is important to note that in this paper, we focus on what can be achieved with the most basic of information communicated via V2V technologies; namely, the GPS coordinates of both the host and remote vehicles, as well as the speed of each vehicle. Of course, a complete pre-crash detection and warning system would combine this rudimentary information with other important technologies, such as camera and radar systems in order to achieve a better understanding of the traffic environment.

The remainder of this paper is organized as follows. Section 2 reviews related work in the area of relative position detection. All position detection systems rely on a distance measure. When it comes to computing geographical distances (geodistances), though, there is no one way to do it, and several options are available. In Section 3, we introduce and evaluate five different well known geodistance functions in terms of accuracy and computational efficiency. Section 4 presents the proposed relative position detection method. Section 5 presents experimental results using naturalistic V2V driving data in an urban setting in the USA with vehicles equipped with dedicated DSRC communication devices.

II. RELATED WORK

Much of the pre-crash detection research has focused on using vehicle-resident sensors such as cameras, radar, and/or LIDAR, 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 [3] – [8].

Bhawiyuga, Nguyen and Jeong 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 [4]. 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.

In [5], Cho and Kim presented an algorithm to estimate the degree of risk at an intersection using a time-to-intersection (TTI) value 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 V2I-based system.

III. GEODISTANCE MEASURES

Many functions have been used to calculate geographical distance, such as the Haversine formula and the great circle formula. We chose five such functions in order to compare their performance both in terms of accuracy and computational efficiency. Let (φ_1, λ_1) and (φ_2, λ_2) be two geographical coordinate points, where φ is the latitude and λ is the longitude. Both φ and λ can be measured in radians or degrees. The first distance measure assumes a spherical Earth projected to a plane and is given by:

$$d_1 = R \sqrt{(\varphi_2 - \varphi_1)^2 + [\cos\left(\frac{\varphi_1 + \varphi_2}{2}\right)(\lambda_2 - \lambda_1)]^2}$$

Here, $R = 6,371.009$ km is the radius of the earth.

The earth is actually ellipsoidal and not a perfect sphere, and the second distance measure is based on an ellipsoidal projection of the earth to a plane.

$$d_2 = R \sqrt{[K_1(\varphi_2 - \varphi_1)]^2 + [K_2(\lambda_2 - \lambda_1)]^2}$$

where φ, λ are in degrees. The two quantities K_1 and K_2 are given by:

$$K_1 = 111.13209 - 0.56605 \cos(\varphi_1 + \varphi_2) + 0.00120 \cos(2[\varphi_1 + \varphi_2])$$

$$K_2 = 111.41513 \cos\left(\frac{\varphi_1 + \varphi_2}{2}\right) - 0.09455 \cos + 0.00012 \cos\left(5\left[\frac{\varphi_1 + \varphi_2}{2}\right]\right)$$

The third distance measure is based on a polar coordinate flat-Earth formula:

$$d_3 = R\sqrt{\theta_1^2 + \theta_2^2 + 2\theta_1\theta_2\cos(\lambda_2 - \lambda_1)}$$

where the colatitude values θ_i are in radians. For a latitude measured in degrees, the colatitude in radians may be calculated as follows: $\theta_i = \frac{\pi}{180}(90^\circ - \varphi_i)$, $i = 1, 2$.

The Haversine distance formula is widely used for computing geographical distances. The Haversine formula can be written in various forms, and the expressions below for d_4 and d_5 are two such variations [10].

$$d_4 = \sqrt{R_{eq}^2(\varphi_1 - \varphi_2)^2 + R_{polar}^2(\lambda_1 - \lambda_2)^2 \cos\left(\frac{\varphi_1 + \varphi_2}{2}\right)^2}$$

Here, $R_{eq} = 6378.137$ km is the radius of the equator, $R_{polar} = 6357.752$ is the radius to the north pole, and φ, λ are in degrees. Finally, d_5 gives a more computationally efficient form:

$$d_5 = R\arccos([\cos(\varphi_1)\cos(\varphi_2) + \sin(\varphi_1)\sin(\varphi_2)])$$

where R is the radius of the earth and φ, λ are in radians.

To test and compare the performance of these geodistance measures, we used vehicle data from a 24 minute trip. The latitude and longitude were sampled at 10 Hz, resulting in a sequence of GPS coordinates:

$$(\varphi_0, \lambda_0), (\varphi_1, \lambda_1), \dots, (\varphi_{n-1}, \lambda_{n-1})$$

For each geodistance function d_i , $i = 1, 2, 3, 4, 5$ we compute the partial sums of distances over the trip data:

$$d_i(p) = \sum_{k=1}^p d_i(\varphi_k, \lambda_k, \varphi_{k-1}, \lambda_{k-1})$$

where $p = 1, 2, \dots, n - 1$ is the p th partial sum.

Figure 2 shows the resulting partial sums over the trip and how they compare to the ground truth. The ground truth was obtained by multiplying the vehicle speed by the trip time. The results for distance measures d_1, d_4, d_5 are so close that the curves overlap each other, and all three distances are close to, though slightly exceed, the ground truth. Distance measure d_3 also exceeds the ground truth, but by a larger margin. This type of distance overestimation is consistent with the result given in [11], where the authors use a statistical analysis to show why GPS distances typically overestimate the true distance. The distance measure d_2 is the outlier here, as it consistently underestimates the ground truth. As mentioned in [3], though, measuring GPS distances without considering geographical information of the road, such as curves, elevations, etc., can lead to distance underestimation. Clearly, measuring geographical distance is not a simple as measuring distances in two dimensions.

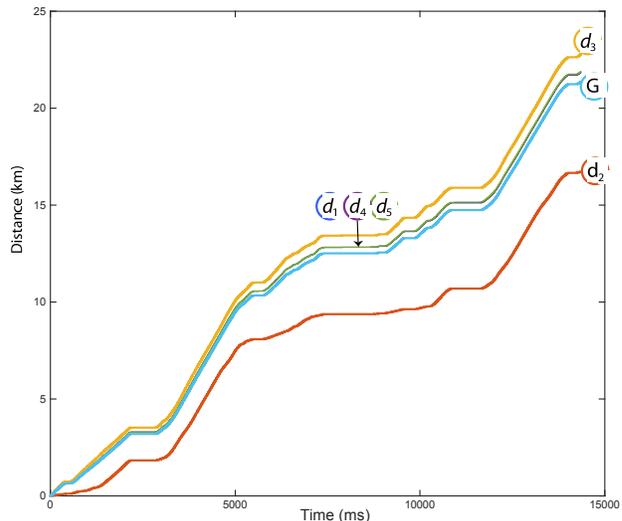


Figure 2. Comparison of five different geodistances computed over a vehicle trip of duration 24 minutes with a sampling rate of 10 Hz. The ground truth (G) is indicated in light blue.

Table 1 shows the total distance computed by each distance function, as well as the computation time required. For comparison, the total trip length indicated by the ground truth is 21.3650m. Note that these experiments were run on a MacBook Pro with an Intel Core i5 2.6 GHz processor. Here, if we choose among the most accurate distance measures: d_1, d_4, d_5 , then d_5 is preferable because it is computationally the most efficient.

Table 1. Comparison of geodistance measures.

Distance Measure	Distance (km)	Time (sec)
d_1	21.8452	0.076359
d_2	16.7218	0.076319
d_3	22.7835	0.071223
d_4	21.8580	0.084919
d_5	21.8942	0.023949

IV. RELATIVE POSITION DETECTION

The goal of this section is to localize the relative position of a remote vehicle R with respect to a host vehicle H . The relative position is then used to provide a driving context for the two vehicles. Figure 3 shows a schematic diagram of the type of classification we want to make here. The vehicle in the red center square is the host vehicle. The surrounding blue squares show potential positions of a remote vehicle. In this relative road position analysis, we are interested in whether the remote vehicle is ahead (A), behind (B), or parallel (P) to the host vehicle. In addition, we are interested in whether or not the remote vehicle is in the same lane as the host. Lanes are numbered as follows: let 0 indicate the lane that the host

is in. Lanes to the right of the host are indicated using positive integers and negative integers are used for lanes to the left of the host. So the remote vehicle in the upper left hand corner of Figure 4 would be classified as \mathbf{A}_{-2} as it is ahead of the host vehicle and two lanes over to the left.

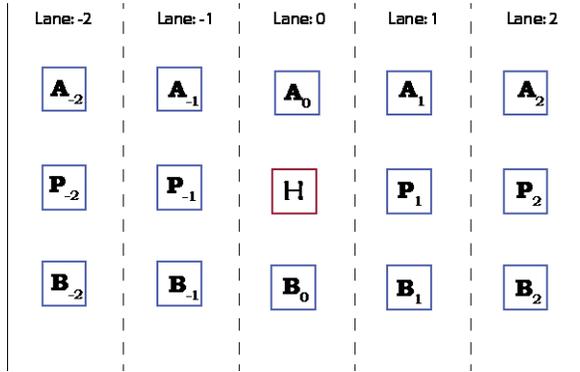


Figure 3. Categories for V2V relative position detection. With respect to the host vehicle, \mathbf{A} indicates that the remote vehicle is positioned ahead, \mathbf{B} behind, and \mathbf{P} indicates the remote vehicle is driving parallel to the host. The subscript indicates lane number. The host's lane is index 0, lanes to the right have a positive index and to the left, negative.

Both the host and the remote vehicles are in motion. As shown in Figure 4, let $\mathbf{H}(t) = (\varphi_H(t), \lambda_H(t))$, denote the position of the host vehicle at time t and $\mathbf{H}(t-1)$ denote its position at time $t-1$; similarly, let the temporal positions of the remote vehicle be denoted as $\mathbf{R}(t) = (\varphi_R(t), \lambda_R(t))$ and $\mathbf{R}(t-1)$, respectively. The variables of interest in Figure 5 are listed below. Let $\Delta\mathbf{H}$ be the vector which points in the direction that the host is traveling: from $\mathbf{H}(t-1)$ to $\mathbf{H}(t)$. The magnitude of this vector d_H is given by:

$$|\Delta\mathbf{H}| = d(\varphi_H(t-1), \lambda_H(t-1), \varphi_H(t), \lambda_H(t))$$

The adjacent (or horizontal) component of $|\Delta\mathbf{H}|$ is given by:

$$|\Delta\mathbf{H}|_{adj} = d(\varphi_H(t-1), \lambda_H(t-1), \varphi_H(t-1), \lambda_H(t))$$

The direction of travel of the host vehicle can be computed as follows:

$$\theta_H^* = \arccos(|\Delta\mathbf{H}|_{adj}/|\Delta\mathbf{H}|)$$

Note that angles are measured here using the normal convention where 0° is the direction along the positive x -axis (i.e., λ -axis) and angles increase (in the range $0^\circ - 360^\circ$) in the counter-clockwise direction.

Finally, θ_H , the moving direction of the host, can be obtained by determining the proper quadrant for $|\Delta\mathbf{H}|$ (see Table 2). Note that θ_H is measured in degrees using the standard coordinate system where 0° is east and angle increases in the counter clockwise direction.

Similar calculations can be done for $|\Delta\mathbf{R}|$, the direction of the remote vehicle, and the associated angle θ_R .

Table 2. Computing θ_H from θ_H^* .

\mathbf{d}_H Position	θ_H
1 st Quadrant	θ^*
2 nd Quadrant	$180 - \theta^*$
3 rd Quadrant	$180 + \theta^*$
4 th Quadrant	$360 - \theta^*$

Let \mathbf{d}_{RH} be the vector which points from $\mathbf{R}(t-1)$ to $\mathbf{H}(t-1)$ and β be the relative angle between the host and the remote vehicle at time $t-1$. Similar to the computation of θ_H , β can be computed by first computing the adjacent distance:

$$|\mathbf{d}_{RH}|_{adj} = d(\varphi_R(t-1), \lambda_R(t-1), \varphi_H(t-1), \lambda_H(t-1))$$

And then

$$\beta^* = \arccos(|\mathbf{d}_{RH}|_{adj}/|\mathbf{d}_{RH}|)$$

β can be determined from β^* by using a table similar to Table 2 to identify the proper quadrant for the angle

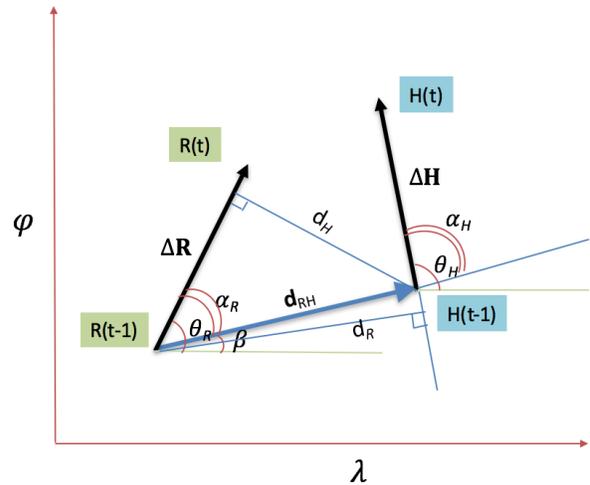


Figure 4. Relative position of the host and remote vehicles at times t and $t-1$.

α_H is the angle between \mathbf{d}_{RH} and the direction of motion $\Delta\mathbf{H}$ of the host vehicle. Similarly, α_R is the angle between \mathbf{d}_{RH} and the direction of motion of the remote vehicle. By this geometry, we have:

$$\alpha_H = \theta_H - \beta$$

$$\alpha_R = \theta_R - \beta$$

Let d_H be the perpendicular distance from the host to the remote vehicle and d_R be the perpendicular distance from the remote vehicle to the host. These are computed as:

$$d_R = d_{RH} |\sin(\alpha_H)|$$

$$d_H = d_{RH} |\sin(\alpha_R)|$$

Using the value of α_H , we can determine which vehicle is ahead of the other. There are 4 cases to consider:

1. $-90 < \alpha_H < 90$: Host is *ahead* of remote vehicle.
2. $-270 < \alpha_H < -90$: Host is *behind* remote vehicle.
3. $\alpha_H = 90$: The remote vehicle is moving *parallel* and *to the right* of the host vehicle.
4. $\alpha_H = -90$: The remote vehicle is moving *parallel* and *to the left* of the host vehicle.

Using the value of d_H , we can determine if the host and the remote vehicles are on the same lane or adjacent lanes by setting up appropriate thresholds. For example, Figure 5 shows an analysis for a typical highway lane width: 3.7m and vehicle width: 2m. These thresholds can be adjusted accordingly for residential streets, which are typically narrower. There are three cases to consider here, as shown below.

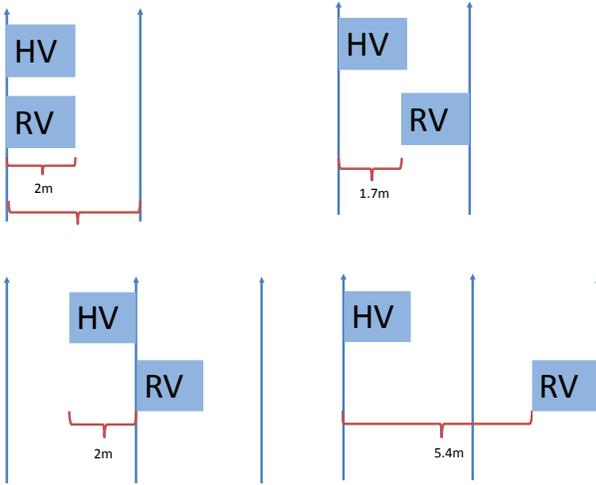


Figure 5. Typical calculations to determine the thresholds to decide if two vehicles are in the same or adjacent lanes.

1. $d_H < 1.7m$: Host and remote vehicles are in the *same* lane.
2. $2 < d_H < 5.4m$: The host and remote vehicle are on *adjacent* lanes
 - (a) $-180 < \alpha_H < 0$: The remote vehicle is on the *adjacent lane to the left* of the host.
 - (b) $0 < \alpha_H < 180$: The remote vehicle is on the *adjacent lane to the right* of the host.
3. $d_H > 5.4m$: There is one or more lane separation between the host and remote vehicle.
 - (a) $-180 < \alpha_H < 0$: The remote vehicle is on a distant *adjacent lane to the left* of the host.
 - (b) $0 < \alpha_H < 180$: The remote vehicle is on a distant *adjacent lane to the right* of the host.

Using the above observations, we formulate the following algorithm to determine the relative position of a remote vehicle with respect to the host.

Algorithm

1. Calculate β , the relative direction from R to H
2. Calculate θ_H and θ_R , moving directions of R and H
3. Calculate α_H and α_R

$$\alpha_H = \theta_H - \beta$$

$$\alpha_R = \theta_R - \beta$$

4. Calculate d_H : vertical distance between the two vehicles

$$d_H = d_{RH} |\sin(\alpha_H)|$$

5. Use Figure 6 to map the values of α_H and d_H to the output classes shown in Figure 3.

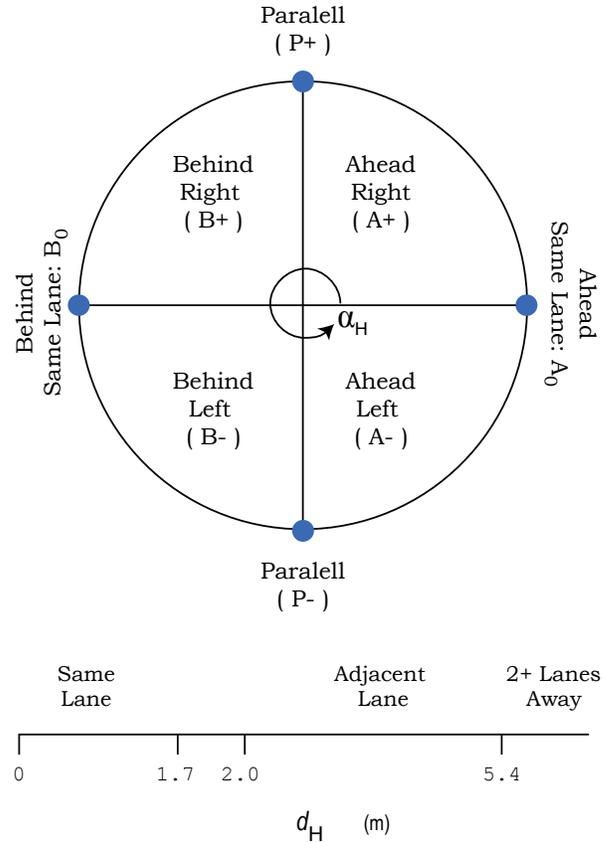


Figure 6. A summary of the positional information about the remote vehicle that can be obtained from α_H and d_H . The sign of the resulting lane index is indicated with +/-.

Note that we are tacitly assuming here that both vehicles have non-zero velocity. In fact, we assume that the vehicle moving direction gives us an indication of the lane direction. But if, for example, the host vehicle comes to a stop, then there is no motion vector between $\mathbf{H}(t)$ and $\mathbf{H}(t-1)$, and hence no angle θ_H to compute. The same applies to the remote vehicle and the calculation of θ_R . Another problem with low vehicle speed is that the GPS coordinates tend to become

more erratic. We don't want to determine travel direction on these noisy GPS measurements. Hence, we enhance step 2 of the algorithm as follows:

If the host vehicle stops or moves very slowly (less than 5 mph), go back in time to find a suitable previous position vector: $\mathbf{H}(t - \Delta) = (\varphi_H(t - \Delta), \lambda_H(t - \Delta))$ to use. The motion vector $\Delta\mathbf{H}$ is then computed between $\mathbf{H}(t - \Delta)$ and $\mathbf{H}(t)$. Similarly, when the velocity of the remote vehicle is too small, choose a suitable previous position vector $\mathbf{R}(t - \Delta)$ to compute θ_R . In the experiments below Δ was determined by going back in time until the distance between $\mathbf{H}(t)$ and $\mathbf{H}(t - \Delta)$ is at least 2.2m (which matches the 5 mph threshold).

V. EXPERIMENTS

V2V data was collected by the *Michigan Mobility Transportation Center* of drivers in and near Ann Arbor, Michigan, USA, over two days: June 2-3, 2013. There were a total of 1825 unique drivers over the two days. Using a sampling rate of 10 Hz, the following data was collected for each trip.

1. Vehicle Id.
2. Trip Id
3. GPS Latitude (5 decimal digits)
4. GPS Longitude
5. Time stamp
6. Vehicle speed (m/s)

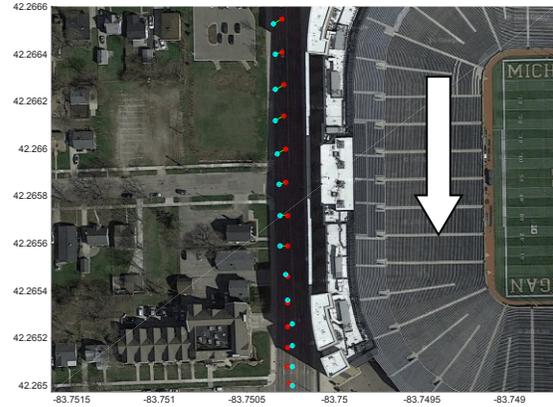
We mined this large data set to find examples of trips where two vehicles were in close proximity at the same time. In particular, the vehicles are within 10 m of each other. Figure 7 shows a portion of three such trips. Here, the position of the host vehicle is shown as a red circle and the position of the remote vehicle as a cyan circle. The direction of travel is indicated by the white arrow. Note for clarity of display purposes, we sub-sampled the trip data using a sample rate of 1 sec. In 7(a), the remote vehicle is initially ahead of the host, but then falls behind. The sequence of states was computed to be:

$\mathbf{A}_1 \mathbf{A}_1 \mathbf{A}_1 \mathbf{A}_1 \mathbf{A}_1 \mathbf{A}_1 \mathbf{P}_1 \mathbf{P}_1 \mathbf{B}_0 \mathbf{B}_0 \mathbf{B}_{-1} \mathbf{B}_{-1} \mathbf{P}_{-1} \mathbf{A}_{-1}$

Note that in this example, all of the states are correct. The accuracy of each state was determined by inspection of the trip data on the map. This inspection was done by viewing a video of the trip sequences on the map (the portion of the trip where the two vehicles interact). It is relatively easy to determine the driving context of the two vehicles when seen in real-time as a dynamic sequence of GPS points.

Another example is shown in 7(b) where the host vehicle passes the remote vehicle on the left with the following sequence of states:

$\mathbf{A}_1 \mathbf{A}_1 \mathbf{A}_1 \mathbf{B}_1 \mathbf{B}_1 \mathbf{B}_1$



(a)



(b)



(c)

Figure 7. Three sample trips. Red dots indicate the position of the host vehicle and the blue dots the remote vehicle. The direction of travel is indicated by the blue arrow.

Note that relative position classification is not always correct, as shown in Figure 7(c). Here, although the sequence of states shown are correct:

$\mathbf{B}_{-1} \mathbf{B}_{-1} \mathbf{B}_{-1} \mathbf{A}_{-1} \mathbf{P}_{-1} \mathbf{A}_{-1} \mathbf{A}_{-1}$

In another portion of the trip, though, the host vehicle comes to a stop, resulting in misclassifications.

As mentioned in Section IV, inaccuracy stems from two main sources: (1) the inherent inaccuracy of the GPS position data and (2) difficulty in determining the direction of vehicle motion at low speeds. The results of the proposed approach are better under freeway driving conditions, rather than local setting where there are many starts and stops.

Table 3 shows the accuracy of several trips. The first four rows in Table 3 show trips where the host and the remote vehicle travel in opposite directions, while the remaining rows show trips where they travel in the same direction. The first two columns show vehicle speed at the moment when the two vehicles were at their closest. The third column shows the closest distance that the host and remote vehicles achieve. The fourth column shows how many seconds the two vehicles are in close proximity.

Table 3. Accuracy of relative position detection.

Velocity Vehicle #1 (m/s)	Velocity Vehicle #2 (m/s)	Closest Dist (m)	Proximity Time (sec)	Accuracy (%)
16.66	0.180	2.37	15	86.7
1.378	0.037	2.47	14	100
33.54	33.94	2.71	13	100
0.963	1.818	2.77	12	100
17.97	17.92	2.22	12	91.6
5.440	0.040	2.37	3	100
0.076	3.400	2.37	6	83.3
9.517	9.515	2.47	9	100
2.222	0.002	2.47	9	100
32.90	32.50	2.70	15	86.6
0.020	4.044	2.71	5	100
19.95	15.78	2.71	4	100

The average computation time (using Matlab with a MacBook Pro Intel Core i5 2.6 GHz) per sample point pair is about 30 microseconds, which is fast enough for real time applications.

VI. CONCLUSION AND FUTURE WORK

A complete pre-crash detection safety system will typically make use of many on-vehicle sensors, such as cameras, LIDAR, etc. The focus of this paper, though, was to determine relative position and driving context using only GPS and vehicle velocity. Even with this limited amount of information, the proposed algorithm produced accurate results for both the vehicle position and traffic context. Although we focused our analysis here on analyzing a single host-remote vehicle pair, the same approach can be applied to determining the relative position of any number of remote vehicles that are within a region of interest to the host.

In future work, we will extend this analysis by using V2V data to infer *driver intent* based on a dynamic analysis of transitions between relative position states. For example, Figure 8 shows a common maneuver whereby the remote vehicle passes the host on the left. This maneuver is characterized by the state sequence: $\mathbf{B}_0 \mathbf{P}_{-1} \mathbf{A}_0$. To understand driver intent, we must go beyond mere localization and position information to analyze such real-time state sequences. For this purpose, a dynamic neural network can be trained on historical data.

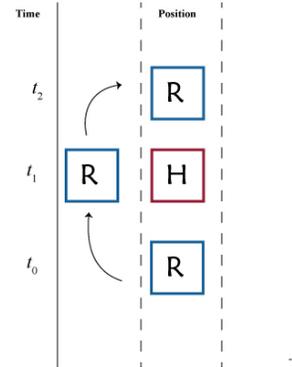


Figure 8. A sequence of states between the host and a remote vehicle. Here the remote vehicle passes the host on the left.

ACKNOWLEDGMENTS

The research presented in the paper was supported in part by a grant from the University of Michigan Mobility Transformation Center (MTC).

REFERENCES

- [1] *Vehicle-to-Vehicle Communications: Readiness of V2V Technology for Application*, DOT HS 812 014, August 2014.
- [2] *Vehicle Safety Communications Project Task 3 Final Report, National Highway Traffic Safety*, DOT HS 809 859, 2005.
- [3] B. Arijit, G. Pilla, and B. R. Tamma, "Microsegmenting: An approach for precise distance calculation for GPS based its applications." 2013 *IEEE Recent Advances in Intelligent Computational Systems (RAICS)*.
- [4] A. Bhawiyuga, H. Nguyen, H. Jeong, "An Accurate Vehicle Positioning Algorithm based on Vehicle-to-Vehicle (V2V) Communications," *IEEE International Conference on Vehicular Electronics and Safety*, July 24-27, 2012. Istanbul, Turkey.
- [5] H. Cho and B. Kim, "Study on Cooperative Intersection Collision Detection System Based on Vehicle-to-Vehicle Communication," *Advanced Science and Technology Letters*, Vol. 58, pp.121-124, 2014.
- [6] Y. Sekimoto, Y. Matsubayashi, H. Yamada, R. Imai, T. Usui, and H. Kanasuqi, "Lightweight lane positioning of vehicles using a smartphone gps by monitoring the distance from the center line." *15th Int. IEEE Conference on Intelligent Transportation Systems (ITSC2012)*, 2012.
- [7] R. Yokoyama, B. Kimura, L. Villas, E. Moreira, "Measuring Distances with RSSI from Vehicular Short-Range Communications." *IEEE International Conference on Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM)*, 2015.
- [8] Y. Shen, H. Dong, L. Jia, Y. Qin, F. Su, M. Wu, K. Liu, P. Li, Z. Tian, "A Method of Traffic Travel Status Segmentation Based on Position Trajectories." *IEEE 18th International Conference on Intelligent Transportation Systems (ITSC)*, 2015.
- [9] <http://www.mathworks.com/matlabcentral/fileexchange/5256-pos2dist>
- [10] P. Ranacher, R. Brunauer, W. Trutschnig, S. C. Van der Spek, S. Reich, "Why GPS Makes Distances Bigger Than They Are," *Int. Journal of Geographical Information Science*. Vol. 30(2), 2016.