RoadMap: Mapping Vehicles to IP Addresses using Motion Signatures

Gopi Krishna Tummala
Department of Computer Science and Engineering
The Ohio State University
tummala.10@osu.edu

Dong Li *
Baidu, Inc.
lidong.main@gmail.com

Prasun Sinha
Department of Computer Science and Engineering
The Ohio State University
sinha.43@osu.edu

ABSTRACT

Inter-vehicular communication (IVC) can be used to enhance the sensing region of vehicles for improved safety on the roads. For many applications based on IVC, the relative locations and communication identities (e.g., IP addresses) of other collaborating vehicles are important for accurate identification. This is particularly challenging to achieve in the presence of legacy vehicles which may not have any sensing or IVC capabilities. We present a system called RoadMap, that matches IP addresses with respective vehicles observed through a camera. It assumes a smartphone or a dashboard camera deployed in vehicles, to identify the vehicles in field of view (FoV), and IVC capability. It runs in the adopted vehicles and accurately matches information obtained through multiple sensing modalities (e.g., visual and electronic). RoadMap matches the motion-trajectories of vehicles observed from the dash-board camera with the motion-trajectories transmitted by other vehicles. To the best of our knowledge, RoadMap is the first work to explore motion-trajectories of vehicles observed from a camera to create a map of vehicles by smartly fusing electronic and visual information. It has low hardware requirement and is designed to work in low adoption rate scenarios. Through real-world experiments and simulations, RoadMap matches IP-Addresses with camera observed vehicles with a median matching precision of 80%, which is 20% improvement compared to existing schemes.

CCS Concepts

•**Hardware** → **Wireless devices;** *Sensor applications and deployments;* •**Computer systems organization** → *Real-time systems;*

*This work is done when the author was at The Ohio State University.

CarSys'16, October 03-07, 2016, New York City, NY, USA

DOI: http://dx.doi.org/10.1145/2980100.2980108

Keywords

Connected Vehicles, Vehicular Localization, Motion trajectories, Design, Experimentation.

1. INTRODUCTION

The popularity of in-vehicle cameras and smartphones provides an opportunity to implement cooperative vehicular applications. The US Department of Transportation issued a new rule requiring car manufacturers to include rearview cameras in all cars manufactured after May 1, 2018 [19]. Meanwhile, smartphones, which are typically equipped with cameras, GPS and radio interfaces, are available to more than 62.5% of the U.S. population [7]. Worldwide smartphone sales accounted for 55% of overall mobile phone sales in the third quarter of 2013 [10]. In vehicles, smartphones can be mounted on the dashboard to provide services such as navigation, over speed warning, and traffic alert. Also, these smartphones can be leveraged to communicate with neighboring vehicles (IVC) and are equipped with cellular connectivity.

Emerging cooperative vehicular applications, such as vehicle platooning [3], adaptive cruise control and autonomous vehicle, can potentially benefit from the information of the vehicles multiple hops away, as well as the information of the immediate neighboring vehicles. The adaptive cruise control system can adjust a vehicle's speed if it knows the acceleration and speed of the vehicles in front. However, in these applications, collaboration can only benefit a vehicle if the relative locations and communication identities (e.g. IP addresses) of the other vehicles are known. For example, vehicles typically use RADAR and LIDAR to scan neighboring vehicles in Line-of-Sight (LoS). However these sensors cannot identify the IP-Address of neighboring vehicles. Communicating with vehicles that have known relative locations can further expand the scanned region. But it is difficult to utilize information provided by vehicles with inaccurate relative location.

The identities of neighboring vehicles can be obtained by leveraging QR-Codes, Ultrasonic communication, Visual Light communications, Wi-Fi MIMO based Angle of Arrival (AoA), or video captured by cameras. Employing these modes of communications can potentially give the identity of vehicle in FoV along with its relative location. However, these schemes require neighboring vehicles to have additional *hardware upgrades*. With minimal assumption of a dashboard camera, computer vision techniques which identify a vehicle based on visual features (color, aspect-ratio, SIFT features etc.) can be employed. However, there can be multiple such vehicles with the same identical visual features. Additionally, when legacy vehicles co-exist, detecting the relative locations and communication identities of the collaborating vehicles is a challenging problem. In this paper, we attempt to solve this problem by assuming minimal hardware requirement (such as a simple smartphone or a deployed dashboard camera). Essentially, RoadMap matches the motion-traces of the vehicles observed from a camera with the motion-traces received from IVC. By doing so, RoadMap solves two major problems involved in cooperative vehicular applications. First, *relative localization problem:* Can the neighboring vehicles be localized with respect to a given vehicle? Second, *targeted communication problem:* Can a vehicle communicate with a vehicle at a given relative location (e.g., vehicle in front)?.

Existing schemes that focus on addressing only one of the problems cannot satisfy the requirement of cooperative vehicular applications. Many schemes have been proposed for vehicle localization such as GPS based localization, map matching, and dead reckoning [5]. These systems cannot determine the relative locations of legacy vehicles. Devices such as camera and RADAR do not require cooperation from other vehicles. But, they do not know which vehicle they are localizing. Schemes based on radio RSSI [16, 21] can potentially localize vehicles not in LoS. The problem is that such schemes do not work for legacy vehicles. By using the emerging IVC techniques, vehicles can collaborate to extend the capability of their sensing devices. To take advantage of information provided by neighboring vehicles, the design must address the following challenges:

- **Lack of observable identities**: A vehicle observes other vehicles through its camera or RADAR, without knowing their global identities (such as MAC addresses or IP addresses) of the detected vehicles. Observable features such as color, aspect ratio of vehicle and radar-signature can correspond to multiple vehicles. Thus, a vehicle cannot use its radio to directly communicate and collaborate with a particular vehicle detected through the camera.

- **Errors in GPS measurements**: Errors in GPS readings make it challenging to associate unique and unambiguous positions to vehicles observed using IVC. Commodity GPS receivers (Standard Positioning Service (SPS)-GPS) have error of 4 meters standard deviation [11] and this error can go beyond several meters in downtown areas due to multi-path effect. Li et al.[16] conducted an experiment in which two GPS devices placed in the same car reported that they are in different lanes in 46% of the cases.

- **Lack of distinguishability in a camera frame**: Vehicles might not be distinguishable in a camera frame due to identical visual features, close spacing or, partial occlusion by another vehicle. Vehicle tracking errors can also lead to discontinuous or erroneous views of a vehicle.

- **Low adoption rate**: A vehicle can only cooperate with other vehicles that have adopted the same or compatible systems. Schemes that require additional software or hardware will not be adopted by all vehicles instantly. So a practical scheme needs to consider the presence of legacy vehicles.

To address these challenges and support cooperative vehicular applications, we seek to provide a global view of the vehicles on road. Global view is represented by a graph-like rigid *structure* with nodes representing vehicles and their corresponding positions. Each node is bundled with associated information such as, IP-Address, GPS, color, or possibly destination of the trip (can be used by platooning applications). For building the global-view, the map of vehicles around a vehicle is the building block. Let us refer to this map of vehicles around a vehicle as a local-map. For building the local-map, a vehicle must perform *relative vehicle localization* and should be able to associate the localized vehicles with their identifiers such as IP-Address, or MAC-Address. Let us refer to the later problem as *IP-Address vehicle matching* problem which is defined as follows: *In a heterogeneous system adoption environment, given a vehicle which can detect the relative location of its neighboring vehicles by sensors (e.g., camera, RADAR) and can communicate with other vehicles with IVC, how to determine the communication identity of the vehicles detected by the sensors?*

Addressing this problem is significantly important for cooperative vehicular applications. Knowing the relative location of the vehicles that are not in the sensing region expands the sensing region. To address this problem, we designed a system called RoadMap. The key contributions are as follows:

- We have designed a novel algorithm that determines the identities of the neighboring vehicles by exploring the movement pattern of the vehicles along with their visual features.

- We conducted a proof-of-concept experiment for the RoadMap system and observed median matching precision of 80% which is 20% higher than existing schemes.

- We simulated RoadMap with high-fidelity configurations. RoadMap simulated in different traffic scenarios and different system adoption rates outperformed existing schemes.

2. SYSTEM DESIGN

System Requirements and assumptions: RoadMap assumes minimal hardware which comprises of a camera, a radio and a GPS receiver. Since a typical smartphone has all these components, RoadMap can be implemented in a smartphone. The low hardware requirement will help to increase the adoption rate of the RoadMap system. RoadMap also accounts for legacy vehicles in its design.

RoadMap uses the camera to detect vehicles (described in §4.1). We call a camera-detected vehicle as a visual neighbor, and assign a unique VID (Visual Identity) to it. Note that a VID is only defined locally within the vehicle that detected this vehicle. If two vehicles detected the same vehicle, these two vehicles might assign a different VID in each of their systems. In the following, we assume that each vehicle only has one camera. In fact, multiple cameras facing different directions can be used in one vehicle to expand the viewing area of the vehicle. The radio is used to communicate with neighboring vehicles. The WiFi module of a smartphone and the DSRC [26] technology can be used as the wireless radio. To be discovered by other vehicles, a vehicle will broadcast its identity, GPS location and visual features to other vehicles. The radio can detect other vehicles by receiving the broadcast information. We call a vehicle received over the radio as an electronic neighbor or an EID (Electronic Identity). Unlike the VIDs, an EID is globally unique. Therefore, different vehicles can easily check if they have common EIDs[1]. The GPS receiver can be used to estimate the GPS coordinates of a vehicle.

RoadMap novel Local Matching (LM) component: The LM component works as follows: a vehicle in RoadMap periodically uses its camera to detect other vehicles, and uses its radio to broadcast its own ID and related information to allow itself to be discovered by other vehicles. At the same time, the vehicle receives the broadcast information from other vehicles over the radio. LM needs to match the vehicles observed through the camera with the vehicles identified through radio communication. Besides using visible features such as color and shape of the vehicle, LM also detects and tracks the movement history of the vehicles in camera. Each vehicle also broadcasts its own visible features and movement trace. After receiving such information from a vehicle over the radio, LM employs matching algorithms to find similarities between the detected visible information and the information received over the radio. The similarity value is calculated to indicate whether the vehicle received over the radio is one of the vehicles in the visual field. In reality, legacy vehicles can be detected in the camera, and the vehicle received over the radio may not be in the view of the camera. In addition, the camera may have low detection accuracy. LM is designed to work in these scenarios.

3. LOCAL VEHICLE MATCHING

This section presents the LM algorithm which matches the VIDs and EIDs.

3.1 Background

Assume a vehicle C has electronic neighbors $E(C)$ and visual neighbors $V(C)$. To identify the IP addresses of the vehicles in $V(C)$, and estimate the relative location of vehicles in $E(C)$, we have to create a match between the two sets of vehicles based on the features of the vehicles. Examples of such features include GPS coordinates and color. In fact, any feature that can be observed or measured by the vehicle itself, and can be observed by its neighboring vehicles can be used in RoadMap. The features of vehicles in $V(C)$ are called visual features, and the features of vehicles in $E(C)$ are called electronic features. The accuracy of these features are limited by the observational variance of the respective sensors. These features are used to find the similarity of an electronic neighbor $e \in E(C)$ with a visual neighbor $v \in V(C)$. Prior works such as Zhang et al. [27] uses camera to help improve the accuracy of wireless localization by comparing the visual distance with electronic distance. ForeSight [17] presents a thorough study of different algorithms to combine visual feature vectors with electronic feature vectors at a given time stamp. It proposes the AdaptiveWeight algorithm to match $E(C)$ and $V(C)$ for vehicle C based on their similarities at a given time. The AdaptiveWeight algorithm will adaptively calculate the weighted mean of the feature similarities to combine different types of features and derive the similarity between a VID and an EID. The weight of each type of feature is calculated based on the distribution of the feature values. For example, if the GPS coordinates of the VIDs are similar, but the colors are distinct, then the color feature is given a higher weight than the GPS feature. Additionally, several works such as [6, 15] have identified the vehicles based on movements and orientation on the road. Nevertheless, these works are limited to visual domain identification of vehicles.

The performance of ForeSight is limited by the lack of temporal information in the system design. The historical matching results are not exploited for the further vehicle matching. Further when the traffic density is high ($|E(C)|/|V(C)|$ are large), ForeSight performs poorly as there are more number of vehicles matching same feature description.

This paper exploits the uniqueness of vehicle movement traces when associated with time. All the features from camera like color, position etc., may not uniquely identify a particular vehicle in visual domain. But the recent movement trace of a vehicle is more likely to be unique and can be measured. This paper takes advantage of the past observations in the visual and electronic domains, and proposes a novel matching algorithm (LM Algorithm) which matches the VIDs and EIDs based on historical movements of vehicles along with the visual features.

3.2 The Local Matching Algorithm

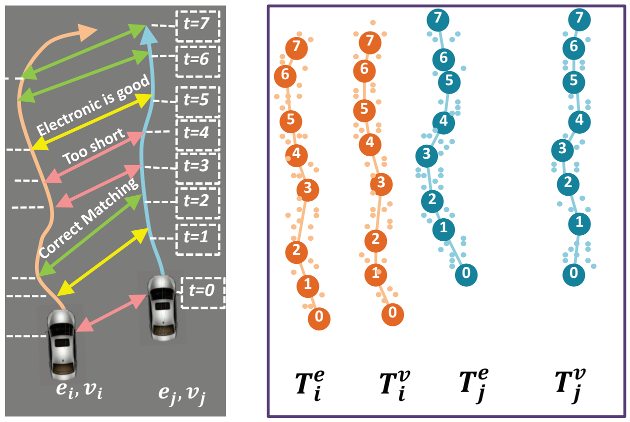
This section presents the importance of trace similarity compared to other features followed by a description of challenges to extract trace similarities between EIDs and VIDs are presented. Finally, this section presents the LM algorithm that takes the movement traces of the VIDs and EIDs, and computes the trace similarity. Then LM algorithm combines trace-similarity with additional features using *AdaptiveWeight Algorithm* from [17] to get matching result.

Motivation: Assume $e_i, e_j \in E(C)$, and $v_i, v_j \in V(C)$ corresponds to vehicle-i and vehicle-j. The movement traces associated with time will provide unique identity of the vehicle, which is observed in electronic and visual domains, by vehicle C. Figure 1(a) shows the movement traces of vehicle-i and vehicle-j and their relative distances over 8 time-slots. Based on the distance between the vehicles, the vehicle pair (i, j) can be classified into one of the following states:

- **State-1**: The vehicles are far enough and are distinguishable in both visual and electronic domains. It gives the correct matching result. It corresponds to the green line in the Figure 1(a).
- **State-2**: The vehicles are very close and cannot be differentiated in both visual and electronic domains. It corresponds to the pink line in the Figure 1(a).
- **State-3**: The vehicles are distinguishable in one domain but not in the other domain. It corresponds to the yellow line in the Figure 1(a).

The distances between vehicle pairs (i, j) will increase or decrease with time, depending on the relative velocity between the vehicles. With time, the vehicle pair (i, j) moves from one state to another. If the vehicles are in **State-2**

[1]In the remaining paper, the terms visual neighbor and VID and electronic neighbor and EID are used interchangeably.



(a) Ground Truth **(b) Movement traces of e_i, v_i , e_j, v_j**

Movement traces of vehicle-*i* and vehicle-*j*

Figure 1: (a) Traces of vehicle-i and vehicle-j associated with time-stamps to give unique identities in visual and electronic domain. The vehicle pair (i, j) are in different states over a span of 8 time-slots. (b) These traces are extracted as movement traces T_i^e and T_j^e in electronic domain and T_i^v and T_j^v in visual domains. Short-term noise and long-term trend can be observed in each time-series. The trace T_i^v is more similar to T_j^e at time slot-3, but longterm trend of T_i^v is more similar to T_i^e which leads to correct match.

and there is relative velocity between them, then after considerable amount of time, they will be in **State-1**. When the vehicles are in **State-1** they can be differentiated. Similarly, the past information can be used to differentiate the vehicles at current time.

Matching based only on current state will be inaccurate, when the vehicle pairs are in **State-2** since there is no conclusive matching result. But past state (**State-1**) information can be used to differentiate the vehicles. By matching the movement traces which have past positions, one can compute the average distance between these observations over a time interval. Matching two movement traces $\mathbf{T}_i^e$ and $\mathbf{T}_j^e$ in electronic domain with two-time series in visual domain $\mathbf{T}_i^v$ and $\mathbf{T}_j^v$ over a time interval t will give the average distances between observations. This average distance is more accurate, and so can be used for matching. Figure 1(b) shows the movement traces in visual and electronic domains. These movement traces have two notable characteristics: one is short-term fluctuations introduced by measurement errors, and the other one is long term trends which capture the movement of the vehicles. Matching the movement traces in visual domain with electronic domain over an interval can remove the effect of short term fluctuations due to *errors in GPS measurements* and events of *indistinguishable vehicles in camera frame*.

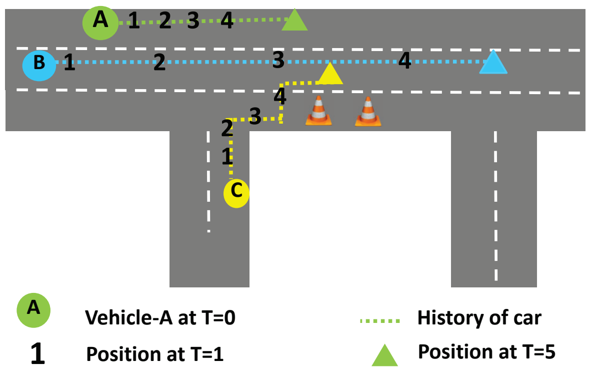
The LM Algorithm: The LM Algorithm (see Algorithm 1) addresses the challenges mentioned in section 1 and matches EIDs and VIDs. The information from different inertial sensors can be merged to smooth the traces of EIDs. These sensor data received from broadcast is smoothed using the Kalman filter [12]. Line 1 of Algorithm 1 takes the electronic traces like position, velocity, acceleration and smoothens out these electronic traces. Later, multiple hypothesis tracking (MHT) for multiple target tracking as described in [4] is used by Line 2 to trace multiple vehicles in the visual domain. This step uses the history information to resolve the conflicts of tracking (*errors in tracking*) a vehicle. Subsequently, line 5 uses exponential moving average to compute the average distance between visual trace and electronic trace. Moving averages are frequently used with time series data to smooth out short-term fluctuations and highlight longer-term trends. The *GPS errors* and *camera errors* are short-term fluctuations whereas actual movement trace of the vehicle is a long term trend. The effect of short term fluctuations can be smoothed out by using moving average. *Matching error* propagation problem is avoided as the erroneous event weight decreases with time, and the current event is given a higher weight. Also in the scenarios of exchanged VID due to *tracking errors*, LM corrects the matching since it considers the history information for performing matching. The similarity of the two traces based on the average distance is computed in line 8. The similarity in other features are computed by line 11. This trace similarity is combined with the similarities in the other domains using AdaptiveWeight algorithm mentioned in [17] to give matching result at line 12. Finally, this step helps in identifying two vehicles when they are clearly distinguishable using additional features compared to their motion-traces by giving them more weight. The similarity matrix which is output of the LM algorithm is used to determine the matching by using threshold

Algorithm 1: LM Algorithm

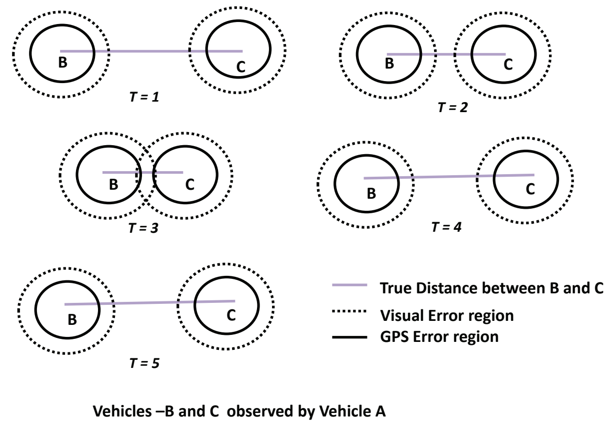
// $\mathbf{P}^v(t)$ Positions of VIDs at time t.
// $\mathbf{P}^e(t)$ Positions of EIDs at time t.
// $\mathbf{P}(t-1)$ Average distance between EIDs and VIDs until time $t-1$
// $\mathbf{f}^v(t)$ Visual Feature vector at time t.
// $\mathbf{f}^e(t)$ Electronic Feature vector at time t.
Input : $\mathbf{P}^v(t)$,$\mathbf{P}(t)^e$, $\mathbf{P}(t-1)$, α, w, $\mathbf{f}^v(t)$, $\mathbf{f}^e(t)$
Output: Similarly Matrix $\mathbf{S}$
// Smoothing the electronic trace
1 $\mathbf{P}^e(t) \leftarrow$Kalman $(\mathbf{P}^e(t), \mathbf{V}^e(t), \mathbf{A}^e(t))$;
// MHT Multi-Hypothesis tracking [4]
2 $\mathbf{P}^v(t) \leftarrow$ MHT $(\mathbf{P}^v(t))$;
3 **for** $P_i^v(t) \in \mathbf{P}^v(t)$ *and* $P_j^e(t) \in \mathbf{P}^e(t)$ **do**
4 **if** $t \neq 0$ **then**
// α is the weight given to the past event.
5 $P_{ij}(t) \leftarrow \alpha P_{ij}(t-1) + (1-\alpha)|P_i^v(t) - P_j^e(t)|$;
6 **else**
// Default weight for new EID-VID match
7 $P_{ij}(t) \leftarrow w$;
// $(P_{ij}(t))$ is the average distance between i^{th} visual trace and j^{th} electronic trace
8 **for** $i = 1$:Size $(\mathbf{P}^v(t))$ *and* $j = 1$:Size $(\mathbf{P}^e(t))$ **do**
// $S_T(i,j,t)$ is the trace similarity between v_i and e_j at time t.
9 $S_T(i,j,t) \leftarrow$ GetSimilarity $(P_{ij}(t))$;
10 **for** *Featuref* **do**
// Similarity in feature f at time t.
11 S_f =GetSimilarity $(\mathbf{f}^v(t), \mathbf{f}^e(t))$;
// Combining similarities across different features
12 $\mathbf{S}$ =AdaptiveWeight $(\mathbf{S}_f, \mathbf{S}_T)$;

The following example illustrates the matching performed by vehicle A using the moving average algorithm that avoids the effect of short term fluctuation.

Example: Figure 2 shows the matching of vehicles B



(a) Movement History of 3 vehicles A, B, C observed in 6-time slots.



(b) Based on the movement history of vehicles B and C; Vehicle-A uses LM algorithm to improve accuracy at time slot-3.

Figure 2: VID-EID matching using movement traces. When all the vehicles are identical their random movements give unique identity based on history.

and C done by vehicle A. Vehicle A can hear from both the B[2] and C and also can see them in visual domain. Figure 2(a) shows the true movements of A, B and C on the road with time. A which is moving on the left lane observes passing-by vehicles, B and C over 5 time-slots. Figure 2(b) shows the visual information and electronic information errors bounds in the system deployed in A. The system captures the movement history from the broadcast messages to match their respective VIDs with EIDs. The following observations happen at

- Time $T = 1, 2$: Both B and C are far apart as shown in Figure 2(b), are in **State-1**. A can match EIDs with VIDs correctly.

- Time $T = 3$: The vehicles are very close and cannot be differentiated in visual domain. Based on the GPS, A can deduce relative position of B and C, but it cannot map to corresponding visual neighbors. The vehicle pair (B, C) are in **State-3**, By taking the mean distance across the trace, and based on the mean the vehicle pair is converted to **State-1**, thereby the VIDs are correctly matched to VIDs.

- Time $T = 4, 5$: Similar to $T = 1, 2$, vehicles are far apart leading to correct match and so are in **State-1**. If there is wrong match in time $T = 3$, due to the current event weight, it will not be propagated in time. This way *error propagation* can be avoided.

4. EXPERIMENTS

In this section, we first introduce our implementation of the vision based vehicle detection and tracking system. The reason for implementing our own vision based vehicle detection system is the lack of a working open-source vehicle detection software. Based on the vehicle detection system, we performed proof-of-concept experiments to evaluate RoadMap .

[2] A, B and C refers to vehicle A, vehicle B and vehicle C respectively.

4.1 Vision Based Vehicle Detection

In our vehicle detection system, we mounted a smartphone camera on the dashboard of a vehicle to detect vehicles visible in the camera's field of view as well as the lane markers on the road. Currently, the system is only designed for daylight condition. Techniques such as rear light based vehicle detection [20] can be implemented in the future to detect vehicles at night. The system has three major components: lane detection, vehicle detection and vehicle tracking, which are briefly described below.

Lane detection: Lane detection is an effective way to remove many of the false positives that would have normally been detected when searching for vehicles. Lane detection searches for all straight lines in an area in front of the user's vehicle. We prune the lines that could not indicate a lane marker due to length, location, or angle. The pruned lines are sorted and combined into lane markers based on proximity to each other. This algorithm is able to determine if the lane detected is a solid line or a dashed line. This information allows the vehicle detection algorithm to ignore anything that is outside of solid lines (meaning it is off the road) but still identify vehicles in neighboring lanes.

Vehicle detection: Vehicle detection relies on a tiered system of key characteristics to identify vehicles in each frame of the video. The first identifier is used to contrast between the bottom of the vehicle and the road. During preliminary work it was discovered that the region directly beneath the car was significantly darker than the rest of the road even during cloudy conditions. These dark areas provide a region in which more computationally intensive algorithms can be run more efficiently. The region is determined based on the size of the dark area identified. The next identifier determines if the dark areas are symmetric enough to be considered vehicles. This is a successful identifier because rear views of cars are symmetric while tree lines (the most common false positive identified in the previous tier) are not. Next, each region of interest is scanned to determine the edges of objects in the region. From this scan, we try to identify the top of the vehicle. This is accomplished by searching for horizontal lines in a certain portion of the region. This location is predetermined based on the size of the region. The length of each horizontal line is scaled based on its distance from the expected location of the top of the car. This technique prevents the algorithm from incorrectly identifying the horizon or a bridge as the top of a vehicle simply because it is a long straight line in the region. If no sufficiently long line is found it is determined that the region does not contain a vehicle and is removed. Finally, any remaining regions are returned as containing one vehicle.

Relative localization: The distance to the car in front of the user is estimated using lane markers as a measurement. Lane markers on freeways have fixed length (3 foot) and have a 9-foot gap between the lane markers [25]. This standard creates a relationship between vertical position in image and distance in real world. The position of the vehicle in image is estimated based on the relative size of the lane markers.

Color Estimation: To estimate the color of a vehicle, we used the k-means algorithm implemented in [8] to cluster the colors on the detected vehicle to find the largest color cluster. The mean of the largest cluster is used as the color of the vehicle.

Vehicle tracking: Vehicle tracking provides an added level of confidence that the regions detected are, in fact, vehicles. Tracking is done by comparing the detection results of the neighboring frames. We assign a score for each region that has been identified to be a vehicle. The score increases if the region is identified as a vehicle in the next frame, and decrease otherwise. Only the regions in a frame that have scores above a threshold are reported as detected vehicles.

Vehicle detection result: To evaluate our vehicle detection system, we used videos pre-recorded by a Samsung GS4 phone mounted on a car's dashboard. For each video, we sample 200 frames and examine the detection result by counting the number of correctly detected vehicle, the false positives and the false negatives. Then we compute the precision and recall of the detection result. Based on analysis, the algorithms have achieved a precision of 82% and a recall of 76% for the video recorded under favorable conditions (sunny).

4.2 The Proof-of-concept Implementation

Experiment details: Evaluating RoadMap using real-world driving requires multiple cars and devices, and the cooperation of the drivers. All such factors make the experiment extremely difficult to conduct. Instead, we performed a small scale driving experiment with three different cars in real-world traffic. The goal of the experiment is to show the performance of RoadMap in real world scenarios. One car is used as the observer car, and it is always behind the other two cars. We mounted a smartphone in each of the car, to record the GPS positions. The smartphone in the observer car is also used to record videos (at the default resolution of the smartphone) and GPS coordinates during the driving experiment. The ground truth color of each car is known beforehand. In summary, during the experiment, we recorded the driving videos from the observer car and the GPS coordinates of the three cars. In this experiment, the observer car has two EIDs. The electronic features of the EIDs are the recorded GPS coordinates and the ground truth color. The VIDs are the vehicles detected by our vehicle detection system. We extract the visual features from the video by estimating the color and the relative location of the vehicles.

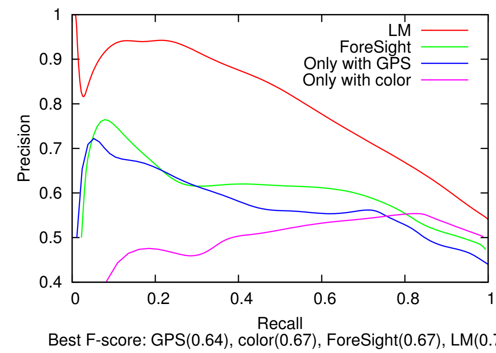


Figure 3: Experiment result. LM outperforms Foresight with median precision improvement of 20% (7% higher F-Score than ForeSight).

Processing: The obtained traces are processed offline, where the communication across different vehicles is emulated using NS3 simulation. Along with RoadMap, ForeSight is implemented for sake of comparison. ForeSight and RoadMap will match the two EIDs onto some of the detected vehicles in the video. The other two vehicles have no VIDs and their structures will only have themselves. To evaluate the matching result, we randomly selected 200 images from the video and manually labeled the correct matches. Next, we used a varying threshold to eliminate low similarity matching results and obtain different recall rates. In addition, we also tried to match the vehicles by only using GPS coordinates and only using color. Figure 3 shows the precision for matching vehicles using LM and ForeSight, while considering GPS and color. This demonstrates that the LM component of RoadMap can improve the accuracy of vehicle matching significantly.

5. SIMULATIONS

Simulation set-up: Evaluating RoadMap with large scale real-world driving requires multiple drivers and vehicles, which makes it difficult to conduct in practice. Instead, we implemented high fidelity simulations using SUMO [14] and NS-3 [24]. SUMO is an open source simulator that can create customized road network and vehicle traffic on demand. NS-3 is a network simulator commonly used to simulate communications between wireless devices. We record the driving traces of the vehicles in SUMO, and simulate each vehicle as a node that moves following the SUMO traces in NS-3. The nodes use 802.11b IBSS mode for communication. Since we mainly compare the performance of our work with ForeSight, we use the same simulation parameters as ForeSight. Colors and GPS coordinates are selected as the two types of features used in LM for vehicle matching. The same configuration is used for the color detection error model and GPS receiver's error model.

Simulation details: We first use SUMO to generate a road map that has a square shape. The length of each edge in the square is 2 kilometers, and the total length of the road in the map is 8 kilometers. There are five one-way lanes on each edge, and the speed limit is 50 km/h. Based on this map, SUMO simulates the traffic and logs the position of each vehicle at each time instant (every second). We used three representative traffic scenarios in the simulation: light

Table 1: Different traffic scenarios in the simulation.

Traffic Condition	Light	Medium	Heavy
Avg. # of vehicles at each time instance	238.13	349.97	749.33
Avg. # of EIDs (100% adoption rate)	8.59	12.74	28.01
Avg. # of VIDs	2.39	3.50	6.28

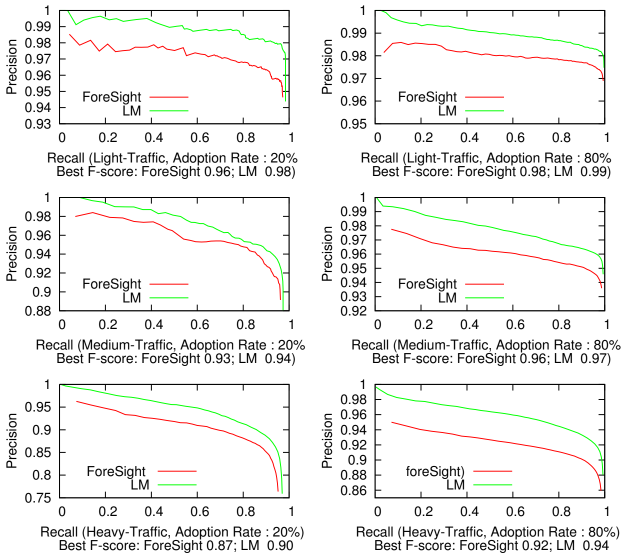


Figure 4: The simulation results of the LM algorithm.

traffic, medium traffic and heavy traffic. The simulation period is 500 seconds. We skipped the first 200 seconds of the traces because the traffic condition is unstable in the beginning of the traces. Table 1 summarizes the basic information for different traffic conditions. These traces are used as input to the NS-3 simulator to simulate the mobility of the vehicles in NS-3.

In NS-3, we install RoadMap on a randomly chosen set of vehicles to simulate different adoption rates. Vehicles that have installed RoadMap will periodically estimate their own GPS coordinates and detect vehicles in LoS. In the simulation, we temporarily set the period to 5 seconds. We modeled the geometric shapes of each vehicle to simulate the visual blockage. Each vehicle is modeled as a rectangle ($3.8m \times 1.75m$). Cameras are installed in the front center of the vehicle. A vehicle C can only see a vehicle in front of the camera if vehicles rectangle has at least one edge visible from C's position. The vehicle will broadcast their own GPS coordinates and color to its neighboring vehicles. After receiving new EIDs, vehicles with RoadMap will match the VIDs and EIDs using the LM algorithm. We randomly select a fraction of the adopted vehicle as the vehicles that has access to a global server. Such vehicles will send their detection results to the global server after executing the LM algorithm.

Results: We have simulated the LM algorithm in different traffic situations and measured for recall and precision. Since not all vehicles on road can have a system deployed, we study the LM for adoption rates of 20% and 80%. Figure 4 shows the simulation results with different traffic densities. The F-score is calculated for each pair of recall and precision, and we show the best F-score for each case in the Figure. By exploring the movement signatures, LM performs better than ForeSight. In the light traffic and medium traffic case, the F-score improvements are between 1% and 2%. In the heavy traffic case, the F-score is improved by 3%. One observation is that the F-score decreases as the traffic density increases. From Table 1 we can observe that as traffic increases, the the average number of EIDs per VID increases. For the heavy traffic case the number is 4.5, whereas for medium traffic situations is 3.6. As the number of EIDs per VID increases, there will be more VIDs matching the color and position for each VID.

6. RELATED WORK

RoadMap allows vehicles to find the IP addresses of their neighboring vehicles which is related to works in *matching information in different domains* and *vehicle localization*.

6.1 Matching Information in Different Domains

The LM component has the same objective as ForeSight [17]. ForeSight is the first work that implemented the unicast communication by vehicle matching. However, ForeSight *only considers the features available in each time instance.* We observed that the movement histories of the vehicles provides a rich set of information that can be leveraged. There are other works that matches information obtained in different domains [23, 27, 18]. Zhang et al. [27] use camera to help improve the accuracy of wireless localization. The proposed EV-Loc system compares the moving traces obtained by wireless AP with the people's moving tracing in camera to improve the localization. These works only consider the location features in the matching. RoadMap can automatically obtain the electronic features given the model and color of the vehicle.

6.2 Vehicle Localization

Many schemes have been proposed for vehicle localization, such as GPS, map matching and dead reckoning. These systems cannot determine the relative location of the legacy vehicles. Devices such as camera and RADAR can be used for relative localization and do not require cooperation of other vehicles. These devices have limited angle-of-view, and can only detect objects in LoS. Schemes based on radio RSSI [16, 21] can potentially localize vehicles not in LoS. The problem is that such schemes do not work for legacy vehicles. Recent vehicular relative localization techniques based on vehicle collaborations include [9, 13, 22]. Fenwick et al. [9] introduced a scheme to allow autonomous vehicles to collaboratively create the road map and localize themselves. Karam et al. [13] and Richter et al. [22] presented schemes for relative localization by exchanging GPS and motion estimations. Their systems assume the vehicles have 100% adoption ratio.

7. FUTURE WORK

We presented RoadMap, a system that can match observed vehicles with their respective unique IDs. The following aspects of the solution need further exploration:

(i) Global Matching: A global matching component can be designed which can fuse the LM results from different vehicles. However, this problem is challenging because of two reasons. First, the LM matching result can have matching errors leading to conflicts. Second, due to the presence of legacy vehicles, there may be limited information about some parts of the road

(ii) Real-time sensor data platform: During the design and implementation of RoadMap, we noticed that one challenge in implementing vehicular applications is the lack of a platform that has access to real-time sensor data. We want to create a portable vehicular application development and testing platform. Currently, vehicles made after 1996 only export limited data though the On-Board Diagnostics (OBD) interface, which is designed for diagnosing the mechanical problems of the vehicles. Some vehicle manufactures, such as GM [1] and Ford (OpenXC [2]), provide a platform for infotainment applications. However, the platform is mainly designed for in-vehicle entertainment within a single vehicle, and only exposes limited sensor data. Our platform includes a set of uniform APIs, which provide data and services. The data-APIs can expose the real-time sensor data to the authorized services and third-party applications. The service-APIs will provide summarized information and services based on the raw sensor data. RoadMap proposed in this work can be one examples of the service APIs.

8. ACKNOWLEDGMENTS

This work was partially supported by NSF grant CNS-1618520.

References

[1] GM Developer APIs. https://developer.gm.com/index.php.

[2] OpenXC. https://developer.ford.com/pages/openxc.

[3] The SARTRE project. http://www.sartre-project.eu/, 2012.

[4] Blackman, S. S. Multiple hypothesis tracking for multiple target tracking. *Aerospace and Electronic Systems Magazine, IEEE 19*, 1 (2004), 5–18.

[5] Boukerche, A., Oliveira, H. A., Nakamura, E. F., and Loureiro, A. A. Vehicular ad hoc networks: A new challenge for localization-based systems. *Computer communications 31*, 12 (2008), 2838–2849.

[6] Coifman, B., Beymer, D., McLauchlan, P., and Malik, J. A real-time computer vision system for vehicle tracking and traffic surveillance. *Transportation Research Part C: Emerging Technologies 6*, 4 (1998), 271–288.

[7] comScore. comScore reports october 2013 U.S. smartphone subscriber market share. https://www.comscore.com/Insights/Press_Releases/2013/12/comScore_Reports_October_2013_US_Smartphone_Subscriber_Market_Share, December 2013.

[8] de Hoon, M. J., Imoto, S., Nolan, J., and Miyano, S. Open source clustering software. *Bioinformatics 20*, 9 (2004), 1453–1454.

[9] Fenwick, J. W., Newman, P. M., and Leonard, J. J. Cooperative concurrent mapping and localization. In *Proc. of Robotics and Automation* (2002), vol. 2, IEEE, pp. 1810–1817.

[10] Gartner. Gartner Says Smartphone Sales Accounted for 55 Percent of Overall Mobile Phone Sales in Third Quarter of 2013. Press Release http://www.gartner.com/newsroom/id/2623415, 2013.

[11] GPS.gov. Global positioning system standard positioning service performance standard, 2008.

[12] Kalman, R. E. A new approach to linear filtering and prediction problems. *Journal of basic Engineering 82*, 1 (1960), 35–45.

[13] Karam, N., Chausse, F., Aufrere, R., and Chapuis, R. Cooperative multi-vehicle localization. In *Proc. of Intelligent Vehicles Symposium* (2006), IEEE, pp. 564–570.

[14] Krajzewicz, D., Erdmann, J., Behrisch, M., and Bieker, L. Recent development and applications of SUMO - Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements 5*, 3&4 (December 2012), 128–138.

[15] Lalonde, M., Foucher, S., Gagnon, L., Pronovost, E., Derenne, M., and Janelle, A. A system to automatically track humans and vehicles with a ptz camera. In *Proc. of Defense and Security Symposium* (2007), International Society for Optics and Photonics, pp. 657502–657502.

[16] Li, D., Bansal, T., Lu, Z., and Sinha, P. MARVEL: Multiple antenna based relative vehicle localizer. In *Proc. of ACM MOBIMCOM* (New York, NY, USA, 2012), Mobicom, ACM, pp. 245–256.

[17] Li, D., Lu, Z., Bansal, T., Schilling, E., and Sinha, P. ForeSight: Mapping vehicles in visual domain and electronic domain. In *Proc. of IEEE INFOCOM* (2014).

[18] Li, X., Teng, J., Zhai, Q., Zhu, J., Xuan, D., Zheng, Y. F., and Zhao, W. Ev-human: Human localization via visual estimation of body electronic interference. In *Proc. of IEEE INFOCOM 2013* (2013), pp. 500–504.

[19] National Highway Traffic Safety Administration. Federal Motor Vehicle Safety Standards; Rear Visibility. https://www.federalregister.gov/articles/2014/04/07/2014-07469/federal-motor-vehicle-safety-standards-rear-visibility, 2014.

[20] O'Malley, R., Jones, E., and Glavin, M. Rear-lamp vehicle detection and tracking in low-exposure color video for night conditions. *In Proc. of Intelligent Transportation Systems 11*, 2 (2010), 453–462.

[21] Parker, R., and Valaee, S. Vehicle localization in vehicular networks. In *Vehicular Technology Conference* (2006), IEEE, pp. 1–5.

[22] Richter, E., Obst, M., Schubert, R., and Wanielik, G. Cooperative relative localization using vehicle-to-vehicle communications. In *Proc. of Information Fusion* (2009), IEEE, pp. 126–131.

[23] Teng, J., Zhu, J., Zhang, B., Xuan, D., and Zheng, Y. F. EV: Efficient visual surveillance with electronic footprints. In *Proc. of IEEE INFOCOM* (2013), pp. 109–117.

[24] The NS-3 Network Simulator. http://www.nsnam.org (June 3, 2012).

[25] US Department of Transportation, Federal Highway Administration. *Manual on Uniform Traffic Control Devices.* 2009.

[26] US. Department of Transportation Research and Innovative Technology Administration. DSRC: The future of safer driving. http://www.its.dot.gov/factsheets/dsrc_factsheet.htm (Feburary 27, 2013), 2012.

[27] Zhang, B., Teng, J., Zhu, J., Li, X., Xuan, D., and Zheng, Y. F. EV-Loc: Integrating electronic and visual signals for accurate localization. In *Proc. of ACM MOBIHOC* (2012), pp. 25–34.