Roadview: Live View of On-Road Vehicular Information

Gopi Krishna Tummala*, Dong Li† and Prasun Sinha‡
*‡Department of Computer Science and Engineering, The Ohio State University, USA
† Baidu, Inc., USA
Email: *tummalag@cse.ohio-state.edu, †lidong.main@gmail.com, ‡prasun@cse.ohio-state.edu,

***Abstract*—Inter-vehicular communication (IVC) can be explored for enhancing collaborative vehicular applications related to traffic statistics, safety by accident prediction and prevention, and energy efficient route planning. For enhancing these applications, a live map of vehicles associated with their communication identities (e.g., IP/MAC addresses) is needed. This is particularly challenging to achieve in the presence of legacy vehicles which might not have any sensing or IVC capabilities. Additionally, vehicles might have diverse sensing capabilities and can have conflicting estimates of parameters of surrounding vehicles. We present RoadView, a system that builds the live map of surrounding vehicles by intelligently fusing the local maps created by individual vehicles. RoadView runs on top of existing local vehicular matching systems (LM) such as Foresight [10] or RoadMap [20]. RoadView is the first work that provides a live map of vehicles by leveraging collaboration across vehicles. Our simulations show that for different adoption rates and traffic densities, RoadView can robustly fuse information from a collection of local maps and enhance vehicles to sense 1.8x (average) number of immediate neighboring vehicles compared to state of art LM algorithms.**

I. Introduction

Each year worldwide road accidents lead to USD 518 billion in losses and 1.3 million deaths. An additional 20-50 million are injured or disabled. Intelligent road transportation offers the promise to sharply cut down these numbers and revolutionize how we travel on the roads. More specifically, intelligent navigational and driving control decisions automatically made by vehicles can lead to reduced chance of accidents, stress-free driving, increased passenger comfort, increased fuel-efficiency and reduced travel time. Some of the high-end cars on our roadways are already equipped with various semi-autonomous features. Tesla S is one such car which supports an autopilot mode with features such as driving within a lane, changing lanes, and managing speed by using active cruise control. Recent works such as Foresight [10] and RoadMap [20] have shown the potential of using various sensing modalities such as RADAR, LIDAR, and cameras to build a local-map of neighboring vehicles. In this paper, we attempt to produce a global information view of vehicles by fusing multiple such local-maps. RoadView is the first work which leverages the sensing capabilities of multiple vehicles to build a collaborative map which also includes legacy vehicles.

The following classes of applications can benefit from such a global map:

- *Traffic statistics based applications:* Existing route planning applications such as Google/Apple maps can benefit from this global map for purposes of road traffic analytics. Additionally, with live traffic count, traffic deadlocks which are prominent in many major cities can be predicted and the traffic can be efficiently routed to alleviate such situations. Applications such as Automatic Traffic Control (ATC) can benefit from information of incoming traffic. The count of vehicles moving from one location to another is vital for planning enhancements to roadways and public transportation facilities.
- *Enhancing safety:* Imminent accidents can be predicted and prevented. Applications such as Adaptive cruise control can benefit from data observed by surrounding vehicles.
- *Energy efficient route planning:* Different vehicles traveling to the same destination can be grouped together to form a fuel efficient formations such as a vehicle platoon [3].

For the above applications, collaboration can only benefit a vehicle if the relative locations and communication identities of the neighboring vehicles are known. For identifying the communication identities of vehicles which are in the Field-of-View (FoV) of a vehicle's sensors, different Local Matching Algorithms (LMs [10, 20]) can be explored. The identities of neighboring vehicles can be obtained by exploring QR-Codes, Ultrasonic communication, Visual Light communication, MIMO with Wi-Fi, radio RSSI [9, 14], and visual features (color, aspect ratio, Scale-invariant feature transform (SIFT) features [15]). By employing such techniques the identity of vehicles in FoV along with their relative locations can be obtained. Techniques such as Foresight [10] or RoadMap [20] can be applied to improve the accuracy in localizing and identifying neighboring vehicles. However such techniques can only provide the map of vehicles in the FoV of the sensors in the vehicle. The sensing region of a vehicle can be enhanced by fusing local maps created by individual vehicles.

However, solutions for fusing the observations from multiple such vehicles have the following challenges:

- *Incomplete local maps due to limited FoV:* The maps produced by individual vehicles are limited by the vehicles they observe using the onboard sensors. Additionally, techniques based on radio RSSI [9, 14] cannot localize non-Line-of-Sight (NLoS) vehicles. Consequently, the local maps may be incomplete which makes it non-trivial to fuse them.

- *The presence of legacy vehicles:* The legacy vehicles may be observed by sensors such as a camera, but they will not be observed in the electronic domain (no messages from such a vehicle).
- *Conflicting observations or errors in LM:* The local maps created at individual vehicles may be inconsistent due to errors in matching vehicles or due to the presence of legacy vehicles.

Thus the problem of Global Matching (GM) is defined as follows: *Given legacy vehicles, time-varying traffic densities, incomplete local maps, and inconsistent local maps, how can the local maps created at the participating vehicles be fused to produce an accurate global map of vehicles?*

We propose RoadView, a system that can provide a global view of the vehicles on road. RoadView works on top of LM algorithms (Foresight [10], RoadMap [20] etc.) and uses novel Global Matching (GM) algorithm to generate a global view of vehicles on road. We call a vehicle, that reports its LM outcome to the server, a reporter. The outcome of the LM component contains the visual neighbors, electronic neighbors and the matching between these two sets of vehicles. GM maintains a graph-like global structure in which each node represents a physical vehicle, and the edge between two nodes represents the relative location of the two nodes. An edge exists between a pair of nodes only if the relative location between the vehicles has been reported by at least one of the reporters. For each received LM outcome, GM first creates a star-like structure, where the center node is the reporter itself, and the satellite nodes are the visual neighbors of the reporter. There are edges between all pairs of nodes in the structure. GM will merge the created structure with the global structure using a modified solution of the maximum common subgraph problem. The idea is to join the two structures based on their overlaps. After merging the structures, the global structure can have more nodes and edges added. The global structure contains the relative locations between vehicles, and the global identity of each vehicle. It can also be used to correct the errors in the LM's outcome. GM is an incremental algorithm. By this design, we do not require all reporters to submit their LM outcome at the same time, and can provide real-time response to the reporters. The contributions of this work are as follows:

- First work to study the challenges involved in building a global information view of the road.
- Proposes a novel GM algorithm that enhances the capability of vehicles to sense 1.8x (average) more number of neighboring vehicles compared to state of the art LM algorithms. Note these neighboring vehicles may not be in FoV of vehicular sensors.
- Evaluates the system with extensive trace-driven simulations and different LMs.

II. SYSTEM DESIGN

RoadView divides the map into road segments, a concept commonly used in digital maps. A road segment represents a portion of a road with uniform characteristics. A road segment has no intersections and contains one or more one-way lanes.

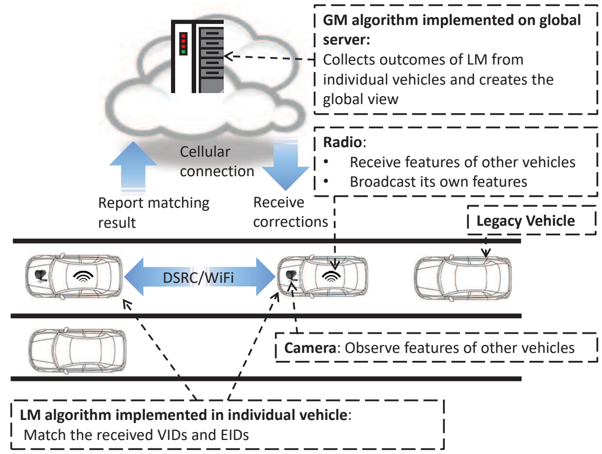

Fig. 1: The system architecture.

We do not assume that all vehicles have adopted the RoadView system. For easy adoption, RoadView has minimal hardware requirements that consists of a camera, a radio, and a GPS receiver. Since a typical smartphone has all these components, RoadView can be implemented in a smartphone.

Let us refer to a camera-detected neighboring vehicle as a visual neighbor, and assign a unique vehicle-specific VID (Visual Identity) to it. Note that a VID is only defined locally by a reporter. If two reporters detected the same vehicle, they will each assign a different VID in each of their systems. Each vehicle advertises its globally unique electronic identity (EID). e.g., its MAC address, along with some visual [10] and kinematic signatures [20].

A. Solution Overview of RoadView

RoadView has two components: the local matching (LM) component and global merging (GM) component. The objective of the LM component is to *match the vehicles detected by camera, with the vehicles that are learnt from the messages received over the radio*. The objective of the GM component is to *collaboratively create a unique view of the vehicles on the road based on the reported detection results from LM*. The LM component is distributed while the GM component is centralized. GM depends on the outcome of LM. Since GM requires having access to a global server, GM does not assume all vehicles will report their LM outcome to GM. This increases robustness and flexibility. The system architecture is depicted in Figure 1.

We have evaluated the GM algorithm with LM algorithms presented in Foresight [10] and RoadMap [20]. After receiving such information from a vehicle over the radio, LM employs matching algorithms to find the similarities between the detected visible information and the information received over the radio. The similarity value is calculated to indicate whether the vehicle learned over the radio is one of the vehicles in the camera's view. In practice, it is possible that a legacy vehicle is detected by the camera and the vehicle learnt over the radio is not in view of the camera.

RoadView creates a global view represented by a graph-like *structure* (say, $\mathcal{G}$). The concept of a structure is commonly used in the computer vision field. Like a graph, there are nodes and edges in a structure, but the edges have fixed orientations in a n-dimensional space (here, we consider n=2) and lengths. In $\mathcal{G}$, each node represents a physical vehicle, and each edge represents the relative orientation and the distance between two vehicles. RoadView builds this global view by employing a novel Global Matching algorithm which incrementally combines the new LM result from a reporter (say, vehicle A) with the Global map ($\mathcal{G}$) as follows:

(1) `CreateStructure`: It creates a structure $\mathcal{M}$ from the output of LM. RoadView fuses $\mathcal{M}$ with $\mathcal{G}$ by creating an association graph between $\mathcal{M}$ and $\mathcal{G}$. (2) `CreateAssociationGraph`: It creates a graph which has nodes representing all potential associations between the visual neighbors of a vehicle and the nodes in $\mathcal{G}$. Each node in the new association graph represents a pair of nodes, where one is a visual neighbor of the reporting node and the other is an existing node in $\mathcal{G}$. (3) `FindMaximumWeightedClique`: It finds a maximum weighted clique in the association graph by defining the weight based on the following two notions of similarity: (i) `NodeSimilarity:` quantifies similarity between two nodes in an association graph based on adaptive weight algorithm [10, 20]. (ii) `EdgeSimilarity`: is a metric quantifying the association between pairs of vertices based on the rigidity of $\mathcal{G}$. These two similarities are combined adaptively to find the maximum weighted clique. This step leverages feature similarity matrices of vehicle A and Global view $\mathcal{G}$. Thus this step resolves any *conflicting observations* by giving more weight to more accurate matching.

Essentially, steps (2) & (3) leverage similarity with a modified version of the *maximum common subgraph problem* for obtaining maximum overlap based on underlying LM results to resolve *conflicting observations*. Finally, the maximum clique is added to the Global map $\mathcal{G}$ and this process is repeated whenever a vehicle adds its LM result to $\mathcal{G}$. Note that a vehicle that has not adopted the RoadView system can also appear in $\mathcal{G}$ if it is detected by other vehicles. Based on $\mathcal{G}$, a vehicle can identify the relative location of another vehicle, and find its identity (IP address) if it has adopted the system.

III. Global Vehicle Merging

A. Motivation

The LM algorithms [10, 20] focus on exploring the features associated with each vehicle to perform vehicle matching. The detection result of a vehicle C from LM algorithms [10, 20] contains visual neighbors and electronic neighbors represented as $V(C)$ and $E(C)$ respectively, and the matching between vehicles in $V(C)$ and $E(C)$. We observed that merging the detection results of neighboring vehicles can help each vehicle to identify and localize more vehicles. In addition, it can potentially correct the matching results of individual vehicles. Here we present two examples. In the first example, a vehicle C is not able to match a VID D because D is far away from C. If C has a correctly matched neighbor E that is located between C and D, then E could help C to match D. The second example is illustrated in Figure 2, where each vehicle can observe the other two vehicles, and vehicle C has incorrect matching result. If vehicles A and B forward their matching results to C, then C can find that there is a conflict between C's matching result and the other matching results. The two examples show that if we have access to the detection results of multiple vehicles, we will have more opportunities to discover neighboring vehicles and increase the accuracy of identified vehicles.

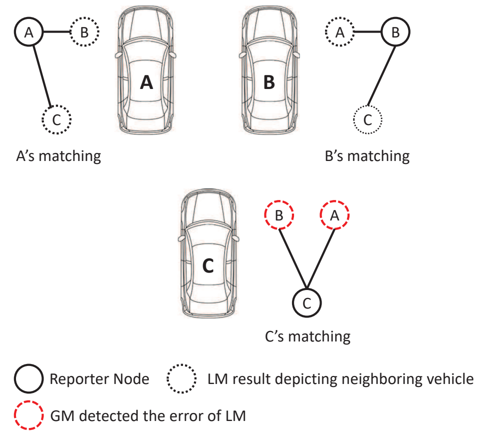

Fig. 2: An example of correcting matching errors with collaboration. The circles in the dotted rectangle represent the relative locations and matching result of the three vehicles. Merging their matching results can correct C's incorrect matching result.

The GM algorithm does not assume that it has the detection results from all vehicles in the road segment. There are several reasons that a vehicle may not be able to report its matching results to the global server: the vehicle does not have the RoadView system; or the vehicle has the system but does not have network connectivity, or the vehicle does not have matching result to report. Here we summarize the challenges in merging the detection results:

- Vehicles cannot directly compare whether they have common VIDs. In the first example, C and E cannot guarantee that they are matching the same VID D. E's matching can be incorrect if D has not adopted the RoadView system.
- The ad hoc approaches introduced in the two examples only apply to scenarios when specific conditions are satisfied. It is challenging, to enumerate all scenarios in which conflicts can happen, especially when detection results from multiple vehicles are used.
- When comparing the detection results of multiple vehicles, the conflicts could be correlated. Correcting one conflict could introduce other conflicts.

B. The Structures Used in GM

A structure $\mathcal{S}$ has a set of nodes $N(\mathcal{S})$ and edges $E(\mathcal{S})$. Each node $n \in \mathcal{S}$ has a set of VIDs and EIDs, denoted by V_n and E_n, respectively.

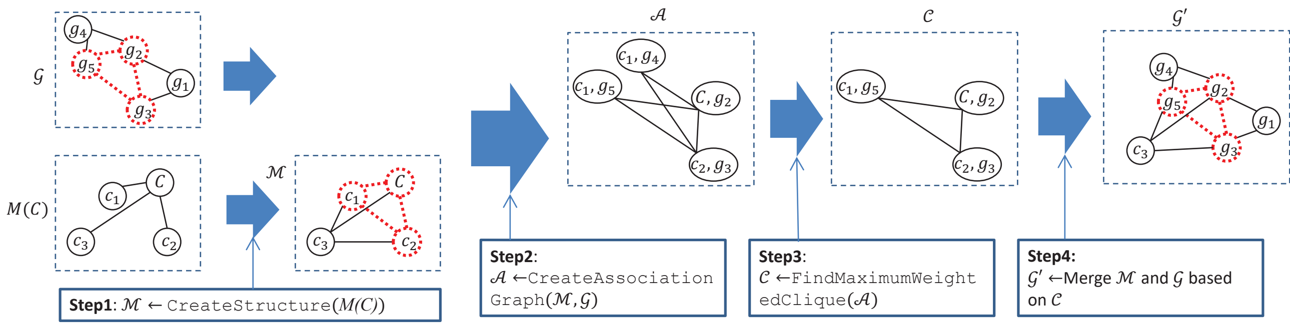

Fig. 3: An example of the GM algorithm. Initially vehicle C has three VIDs $\{c_1, c_2, c_3\}$, and $\mathcal{G}$ has five nodes. The red doted nodes and edges in Structure $\mathcal{M}$ and $\mathcal{G}$ indicate the same sub-structure shared by $\mathcal{M}$ and $\mathcal{G}$. In **Step2**, we assume only vertex-pairs (C, g_2), (c_1, g_4), (c_1, g_5) and (c_2, g_3) have similarities that satisfy the constraints in Algorithm 1. Therefore, only four nodes exist in the association graph $\mathcal{A}$.

As introduced before, the Local Matching (LM) result of C contains $V(C)$, $E(C)$ and a set of matching pairs $M(C) = \{(e, v)\}$, where pair (e, v) indicates that $e \in E(C)$ is matched to $v \in V(C)$. If we draw C and C's visual neighbors $V(C)$ on a 2D plane using the GPS coordinate system, we can get a star-like structure, where the center node is C, and the satellite nodes are $V(C)$. We create structure $\mathcal{M}$ based on the detection result of C. Then we mark node C as the reporter node in $\mathcal{M}$, because C reports matching ($M(C)$) to the global server. Note that in $\mathcal{M}$ each node has only one VID and at most one EID except node C. The EIDs of node C which are not matched to any EID by LM are excluded from creating Global view. Creating the structure $\mathcal{M}$ from matching result $M(C)$ is implemented in `CreateStructure` method. This structure $\mathcal{M}$ is used to update the global view $\mathcal{G}$.

C. Creating the Association Graph

In this section, we introduce the key techniques used in merging the structures. We create an association graph $\mathcal{A}$ based on two structures $\mathcal{M}$ and $\mathcal{G}$, then find the maximum weighted clique $\mathcal{C}$ in $\mathcal{A}$. The maximum weighted clique $\mathcal{C}$ indicates the overlapping structure between $\mathcal{M}$ and $\mathcal{G}$.

Maximum common subgraph background: In the graph theory literature, given two graphs G and H, the association graph S is created as follows [13, 4]. The vertices of S correspond to the vertex-pairs (u, v), where $u \in G$ and $v \in H$. Vertex $(u, v) \in S$ represents the option of matching $u \in G$ and $v \in H$. Therefore the number of vertices in S is $|G| \times |H|$. The edges in S are defined based on the connectivity of G and H. Assume $E(G)$ and $E(H)$ are the edges of G and H, correspondingly, and (u_1, v_1) and (u_2, v_2) are two vertices in S. There is an edge between the vertex (u_1, v_1) and (u_2, v_2) if and only if one of the two conditions is satisfied: i) $(u_1, u_2) \in E(G)$ and $(v_1, v_2) \in E(H)$; or ii): $(u_1, u_2) \notin E(G)$ and $(v_1, v_2) \notin E(H)$. The way of creating an association graph S captures the topology constraints when searching for the common subgraph between G and H. The maximum common subgraph between G and H can be found by finding the maximum clique in association graph S.

In our case, given two structures $\mathcal{M}$ and $\mathcal{G}$, we need to enforce the constraints of the structures when creating the association graph. Algorithm 1 creates a weighted association graph, in which each edge and node has a weight represented by a real number in $[0, 1]$. These weights are created based on two functions `NodeSimilarity` and `EdgeSimilarity`.

`NodeSimilarity`(u, v) signifies the similarity between two nodes. Note the association graph is created between $\mathcal{M}$ and $\mathcal{G}$, therefore $u \in N(\mathcal{M})$ and $v \in N(\mathcal{G})$. First RoadView creates two centroid nodes u' and v' for u and v, respectively in n-dimensional feature space where n is number of features used by LM. Some example features are color of the vehicles, aspect ratio, kinematic signatures. The centroid node u' is created by using the mean of the feature values of the VIDs in V_u and the EIDs in E_u. v' is created in the similar way based on V_v and E_v. If $\mathcal{F}$ is the set of features used by RoadView, for i^{th} feature $f_i \in \mathcal{F}$, u''s value of feature f_i is $u'[i] = \texttt{mean}(f_i, f_i \in \{V_u \cup E_u\})$, and v''s value of feature f_i is $v'[i] = \texttt{mean}(f_i, f_i \in \{V_v \cup E_v\})$. Then `NodeSimilarity` uses the AdaptiveWeight (AW) algorithm [10] to compute the similarity between u' and v'. Note AW algorithm fuses different features by allocating weights based on distinguishability of the feature. For example, if color feature is more distinguishable, AW allocates more weight to the color feature for computing similarity between nodes.

`EdgeSimilarity`$((u_1, u_2), (v_1, v_2))$ is implemented by computing the difference of the feature values of the mean nodes. We first create the centroid nodes u_1', u_2', v_1' and v_2' based on u_1, u_2, v_1 and v_2, respectively. Then we create feature difference vector $w = (u_1' - u_2') - (v_1' - v_2')$, where the minus sign means subtracting corresponding feature values of the nodes. Then `EdgeSimilarity` $= 1.0 - \min\left(1.0, \sqrt{\frac{\sum_{f \in \mathcal{F}} w_f^2}{\sum_{f \in \mathcal{F}} \sigma_f^2}}\right)$, where σ_f is the standard deviation of feature f. This heuristic captures the similarity between two edges in the association graph. If the distance between features is greater than the variance of standard deviation of the feature then the `EdgeSimilarity` is 0, signifying different edges. Our simulation shows that the similarities between the nodes and edges in the structures are well-captured by this heuristic approach.

Constrains on association graph: Different from the commonly used method of creating the association graph, Algorithm 1 imposes the following three extra constraints:

- *Reporters are not merged*: Different reporters represent different vehicles. We do not create an association node when both the corresponding pair of nodes are reporter nodes (Line 3). In this way, we exclude the case that two different reporter nodes are merged into the same node.
- *Threshold on* `NodeSimilarity`: We create an association node only if the `NodeSimilarity` is larger than a threshold τ_1 (Line 4). It excludes matching nodes that are completely different.
- *Threshold on* `EdgeSimilarity`: We do not create an association node only if the `EdgeSimilarity` is larger than a threshold τ_2 (Line 11). It indicates that to merge u_1 with v_1 and u_2 with v_2, edge (u_1, u_2) and edge (v_1, v_2) should have similar orientation and length.

Algorithm 1: Create the Association Graph

Input : $\mathcal{G}$, $\mathcal{M}$
Output: Association Graph $\mathcal{A}$
1 $\mathcal{A} \leftarrow \Phi$;
2 **for** *each vertex-pair* (u, v)*, where* $u \in N(\mathcal{G})$*,* $v \in N(\mathcal{M})$ **do**
3   **if** *not (u and v are reporters, and they are different)* **then**
4     $s \leftarrow$ `NodeSimilarity(`(u, v)`)` ;
5   **if** $s > \tau_1$ **then**
6       $N(\mathcal{A}) \leftarrow N(\mathcal{A}) \cup \{(u, v)\}$;
7       $weight\,(Node(u, v)) \leftarrow s$;
8 **for** $(u_1, v_1) \in N(\mathcal{A}), (u_2, v_2) \in N(\mathcal{A})$ **do**
9   **if** $u_1 \neq u_2$ *and* $v_1 \neq v_2$ **then**
10   **if** $\{(u_1, u_2) \in E(\mathcal{G})$ *and* $(v_1, v_2) \in E(\mathcal{G})\}$ *or* $\{(u_1, u_2) \notin E(\mathcal{G})$ *and* $(v_1, v_2) \notin E(\mathcal{G})\}$ **then**
11       $s \leftarrow$ `EdgeSimilarity(`(u_1, u_2), (v_1, v_2)`)` ;
12   **if** $s > \tau_2$ **then**
13         $E(\mathcal{A}) \leftarrow E(\mathcal{A}) \cup \{((u_1, v_1,), (u_2, v_2))\}$; $weight\,(Edge((u_1, v_1), (u_2, v_2))) \leftarrow s$;

These constraints significantly reduce the number of nodes and edges in the created association graph, which directly reduces the computational complexity of finding the maximum weighted clique. Therefore, `NodeSimilarity` and `EdgeSimilarity` can affect the computing time of the algorithm. Figure 3 shows one example of creating the association graph based on $\mathcal{M}$ and $\mathcal{G}$. Assuming Line 6 in Algorithm 1 allows matching C with g_2, c_1 with g_4 or g_5 and c_2 with g_3, the association graph $\mathcal{A}$ will only have four nodes and two three-node cliques. Note that the node pairs in each vertex of $\mathcal{A}$ indicate the matching options.

D. The Global Merging (GM) Algorithm

In this section, we show how GM merges the structures based on the concept of association graph. The GM algorithm is an incremental algorithm. Initially, $\mathcal{G}$ is empty. When a detection result $M(C)$ from vehicle C is received, GM will convert $M(C)$ into a structure $\mathcal{M}$, and merge the structure with $\mathcal{G}$ based on the overlaps between them. We denote the merged structure as $\mathcal{G}'$. C can request to receive $\mathcal{G}'$ or part of $\mathcal{G}'$ based on C's interest.

Matching $\mathcal{G}$ and $\mathcal{M}$: Algorithm 2 shows the detailed procedure of the GM algorithm. In Algorithm 2, we first convert the detection result $M(C)$ into a structure $\mathcal{M}$ (Line 1 in Algorithm 2). If $\mathcal{G}$ is not empty, we use Algorithm 1 to create association graph $\mathcal{A}$ based on $\mathcal{M}$ and $\mathcal{G}$ (Line 5). After creating the association graph, the problem is reduced

Algorithm 2: Merge $\mathcal{G}$ with detection result $M(C)$

Input : $\mathcal{G}$, $M(C)$
Output: $\mathcal{G}'$
1 $\mathcal{M} \leftarrow$ `CreateStructure(`$M(C)$`)` ;
2 **if** $\mathcal{G} = \Phi$ **then**
3   $\mathcal{G}' = \mathcal{M}$;
4 **else**
5   $\mathcal{A} \leftarrow$ `CreateAssociationGraph(`$\mathcal{G}$, $\mathcal{M}$`)`;
  `// Each node in` $\mathcal{A}$ `is a pair` (u, v)`, where` $u \in N(\mathcal{G})$ `and` $v \in N(\mathcal{M})$
6   $\mathcal{C} \leftarrow$ `FindMaximumWeightedClique(`$\mathcal{A}$`)`;
7   matchedNodes $\leftarrow \{\ \}$;
8   **for** $(u, v) \in \mathcal{C}$*, where* $u \in N(\mathcal{G})$*,* $v \in N(\mathcal{M})$ **do**
    `// Add the VID and EID associated with` v `to the VID set and EID set of` u`.`
9     $V_u \leftarrow V_u \cup V_v$;
10     $E_u \leftarrow E_u \cup E_v$;
    `// Update the EID set of node` v `in` $\mathcal{M}$
11     $E_v \leftarrow$ `VoteForEID(`u`)`;
12   matchedNodes $\leftarrow$ matchedNodes $\cup \{v\}$;
  `// Add` $N(\mathcal{G})$ `and the un-matched nodes to` $\mathcal{G}'$
13   $N(\mathcal{G}') \leftarrow N(\mathcal{G}) \cup \{N(\mathcal{M}) \setminus$ matchedNodes$\}$;
  `// Add corresponding edges in` $\mathcal{M}$ `to` $\mathcal{G}'$
14   $E(\mathcal{G}') \leftarrow E(\mathcal{G}) \cup E(\mathcal{M})$;

to finding the maximum weighted clique in graph $\mathcal{A}$. In GM, the maximum weighted clique is defined as the clique in $\mathcal{A}$ that maximizes the total weight of the nodes and the edges. Finding the maximum weighted clique in an arbitrary graph is an NP-hard problem [4, 13]. Any maximum weighted clique detection algorithm can be applied in Line 6. In our simulation, we implemented the pivoting version of the Bron-Kerbosch Algorithm [5] due to the simplicity in implementing it. The time complexity of this algorithm is $O(3^{n/3})$ [6].

Fusing matching results of $\mathcal{G}$ and $\mathcal{M}$: After finding the matched node pairs based on the clique, we save the VIDs and EIDs associated with the nodes in $\mathcal{M}$ to the corresponding nodes in $\mathcal{G}$ (Line 9- 10). In this way, we combine the matching result in $M(C)$ with the matching results merged into $\mathcal{G}$ previously. For each node in $\mathcal{M}$ that has a matched node in $\mathcal{G}$, we find the representative EID of the corresponding node in $\mathcal{G}$ by combining all the related matching results reported to $\mathcal{G}$. This representative EID is assigned to match with the only VID in $\mathcal{M}$'s node (remember that there is at most one VID in each of $\mathcal{M}$'s node). In the algorithm, we use the `VoteForEID` procedure to detect the representative EID of a node in $\mathcal{G}$ (Line 11). A node $u \in \mathcal{G}$ could contain multiple associated VIDs in V_u and multiple associated EIDs in E_u. `VoteForEID` will find the EID for VIDs in V_u based on the following four rules.

1) If u has a reporter, then the VIDs in V_u are matched with u. `VoteForEID` returns the reporter of u. We will correct the EIDs $e \in E_u$ if e is not the same as the reporter of u.
2) If $E_u = \Phi$, it means that there is no EID can match with the VIDs in V_u. `VoteForEID` will return nothing. It indicates that u represents a legacy vehicle.
3) If $|E_u| = 1$, all the VIDs in V_u are matched to the only EID in E_u, which is the output of `VoteForEID`.
4) If $|E_u| > 1$, we create a VID v' that is the centroid of the VIDs in V_u. `VoteForEID` returns the EID that has the maximum similarity with v'.

Improving LM matching result by GM: In Items 1, 3 and 4, if any reporter or EID have been selected, we match the VID of the node in $\mathcal{M}$ with the selected EID. It can potentially improve the matching recall and precision of $\mathcal{M}$. The un-matched nodes in $\mathcal{M}$ are also added to $\mathcal{G}'$ (Line 13). In the future matching process, these un-matched nodes can potentially be matched with the nodes in the new structure. Line 14 adds the edges in structure $\mathcal{M}$ to $\mathcal{G}$. This is an important step as it creates connections between the nodes in $\mathcal{G}$ and the node newly added by $M(C)$, which can let the existing nodes learn the relative location of vehicles that does not exist in their list of VIDs. Therefore, after the matching, structure $\mathcal{G}'$ is also valuable for vehicles who have previously submitted their detection results before C. In Figure 3, we assume the maximum weighted clique $\mathcal{C}$ is the clique with nodes $\{(c_1, g_5), (C, g_2), (c_2, g_3)\}$. $\mathcal{C}$ indicates matching c_1 with g_5, C with g_2 and c_2 with g_3. We merge these node pairs, and finally add the un-matched node c_3 into $\mathcal{G}$ to create the merged structure $\mathcal{G}'$. Note that the edge between c_3 and g_2 is one of the edges that do not exist in $\mathcal{G}$. It indicates that by merging the detection result $M(C)$, vehicles associated with node g_2 can discover the relative location with the vehicle associated with c_3. In our simulation, we examine the degrees of the nodes in $\mathcal{G}$ to indicate how GM helps the vehicles to discover extra neighbors.

The feature values of the VIDs and EIDs will continually keep on changing. The existing values saved in $\mathcal{G}$ need to be updated. To address this problem, the GM algorithm records the time-stamp when the VIDs and EIDs are merged into $\mathcal{G}$. GM uses time alignment techniques [11] to update the state of the vehicles, based on the speed and the map of the road. Upon invocation, GM removes the VIDs and EIDs that are merged into $\mathcal{G}$ more than τ seconds ago.

IV. Simulations

Simulation set-up: Evaluating RoadView 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 [8] and NS-3 [18]. SUMO is an open source simulator that can create customized 2D 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 [10] and RoadMap [20] we use the same simulation parameters as mentioned in [10, 20]. 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.

TABLE I: 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

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 on 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 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 at the beginning of the traces. Table I summarizes the basic information at 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 RoadView on a randomly chosen set of vehicles to simulate different adoption rates. Vehicles that have installed RoadView 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 its rectangle has at least one complete edge visible from C's camera position. The vehicle will broadcast its own GPS coordinates and color to its neighboring vehicles. After receiving new EIDs, vehicles with RoadView will match the VIDs and EIDs using the RoadMap algorithm. We randomly select a fraction of the adopted vehicle as the vehicles that have access to a global server. Such vehicles will send their detection results to the global server after executing the RoadMap algorithm.

A. Evaluation of the GM Component

In this section, we focus on evaluating the performance of the GM component and examine the properties of the global structure $\mathcal{G}$. Although GM is implemented on top of RoadMap, we only label GM in the following figures because the legend space is limited. $\mathcal{G}$ contains the relative locations and the IP addresses of vehicles that have adopted the RoadView system.

The percentage of the reporter vehicles among the adopted vehicles is denoted by r. We select $r = 20\%$ and $r = 80\%$ as two representative cases in the simulation to show how it affects the performance of the GM algorithm.

Precision, Recall, and F-Score in contrast to LMs: First, we simulate the performance of vehicle matching in $\mathcal{G}$. We only consider the matching accuracy and recall for the reporter vehicles because only their vehicle matching result could be changed by $\mathcal{G}$. Different adoption rates and different traffic conditions are simulated. For each adoption rate and traffic condition, Figure 4 plots the best F-score, and the corresponding precision and recall of ForeSight, Roadmap, GM with $r = 20\%$ and GM with $r = 80\%$. Note that when $r = 20\%$ and the adoption rate is 20%, there are only 4% $(20\% \times 20\%)$ of vehicles that have access to the global server. When $r = 20\%$, the density of the reporters is low, and there will be less collaborations between the reporters. Therefore, the performance of GM with $r = 20\%$ only has slight improvement compared with the performance of Roadmap. When $r = 80\%$, the average F-score improved 0.8% compared with Roadmap in different traffic conditions. Although seems small, the improvement is still significant because the F-score, precision, and recall of Roadmap is already high, and the improvement space is very limited. For example, when $r = 80\%$ and traffic condition is medium, the recall of RoadMap is 0.974, and the recall of GM is 0.986. The improvement is 1.2%. However, it also means that 46.2% (which is calculated by $(0.986 - 0.974)/(1.0 - 0.974)$) of the un-matched neighboring vehicles in RoadMap found a match in GM.

Enhanced sensing by GM: Unlike LM, GM allows a reporter to discover the relative locations of vehicles that it may not able to detect through its camera. The degree of a reporter node C in $\mathcal{G}$ indicates the number of visual neighbors of C, plus the number of vehicles that are added by the vehicles matched with C. The degree of a node represents the number of immediate neighboring vehicles that have known relative locations. Figure 5(a), 5(b) and 5(c) show the average degree of the reporter nodes in $\mathcal{G}$ for different adoption rates. Note that the average number of VIDs only depends on the traffic condition, and does not change as the adoption rate increases. On the other hand, the average number of EIDs increases linearly with the adoption rate. As we have expected, as r increases from 20% to 80%, the degree of the nodes increases. One interesting observation is that when the adoption rate is larger than 50%, the average degree stops increasing and stays close to $2\times$ of the average number of VIDs. Figure 5(d) depicts enhancement to the sensing capability achieved by the GM algorithm for different traffic densities compared to LM algorithms. Figure 5(d) depicts this enhanced sensing of 1.8x times for light traffic, 1.6x for medium traffic 1.3x for heavy traffic scenarios. In the simulation, each adopted vehicle only has one camera facing front. $2\times$ the average number of VIDs is roughly the average number of VIDs a vehicle could observe if it has one front-facing camera and one camera facing back. By collaboration, GM discovers neighboring vehicles not in the view of the cameras and significantly increases the number of neighboring vehicles with known relative locations. This is extremely useful for applications such as blind-spot detection. At the same time, GM maintains high matching precision, recall and F-score.

Computational Intensity of GM: Finally, we study the size of the association graph $\mathcal{A}$ and the clique $\mathcal{C}$ in different traffic conditions. We assume the adoption rate is 100%, and all vehicles are reporters, which is the most compute-intensive setting. We use $|N(\mathcal{A})|$ and $|E(\mathcal{A})|$ to denote the number of nodes and edges in association graph $\mathcal{A}$, and use $|N(\mathcal{C})|$ to denote the number of nodes in clique $\mathcal{C}$. We plot the CDF (Cumulative Distribution Function) of $|N(\mathcal{A})|$, $|E(\mathcal{A})|$ and $|N(\mathcal{C})|$ in Figure 6. Although the clique detection problem is an NP-hard problem, GM can significantly reduce the size of the problem and work efficiently. Heavy-traffic scenarios are the most compute-intensive. The average value of $|N(\mathcal{A})|$ is 21.7 and the average value of $|E(\mathcal{A})|$ is 9.3. In medium traffic and light traffic scenarios, the size of the association graph is even smaller.

V. Related Work

RoadView enables vehicles to find the IP addresses of their neighboring vehicles, and it can combine the matching results into a global view of the vehicles on road. There are related works in matching information in other domains and graph matching.

A. Matching Information in Different Domains

RoadView uses the Adaptive weight algorithm for fusing the `NodeSimilarity` which signifies the similarity between two nodes used by GM in fusing a reporter node with the existing GM structure. The adaptive weighted algorithm is employed by on-vehicular matching systems such as Foresight [10] and RoadMap [20]. Similarly, adaptive weighted algorithms are employed by [19, 21] for vehicular to infrastructure (V2I) pairing the vehicles observed over camera (VIDs) with their respective EIDs. In contrast, RoadView also uses novel metric `Edgesimilarity` which signifies the similarity between edges in Global-map $\mathcal{G}$ and new detection result $M(C)$. The metric explores the rigidity of the vehicular map structure to improve matching results and minimizing the errors in combining new detection results.

B. Graph Matching

The GM algorithm merges the matching results of individual vehicles. Related works include graph matching and jigsaw puzzle matching problem. The graph matching problem can be reduced to the maximum clique problem by creating the association graph, which is an NP-hard problem [4]. Pardalos and Xue [13] and Bomze et al. [4] have detailed survey of the maximum clique problem. In 2001, Robson [16] designed an algorithm that can find the maximum clique with time complexity $O(2^{n/4})$. This is currently the best known result. RoadView has the freedom to employ any maximum weighted clique detection algorithm. We enforce three restrictions when

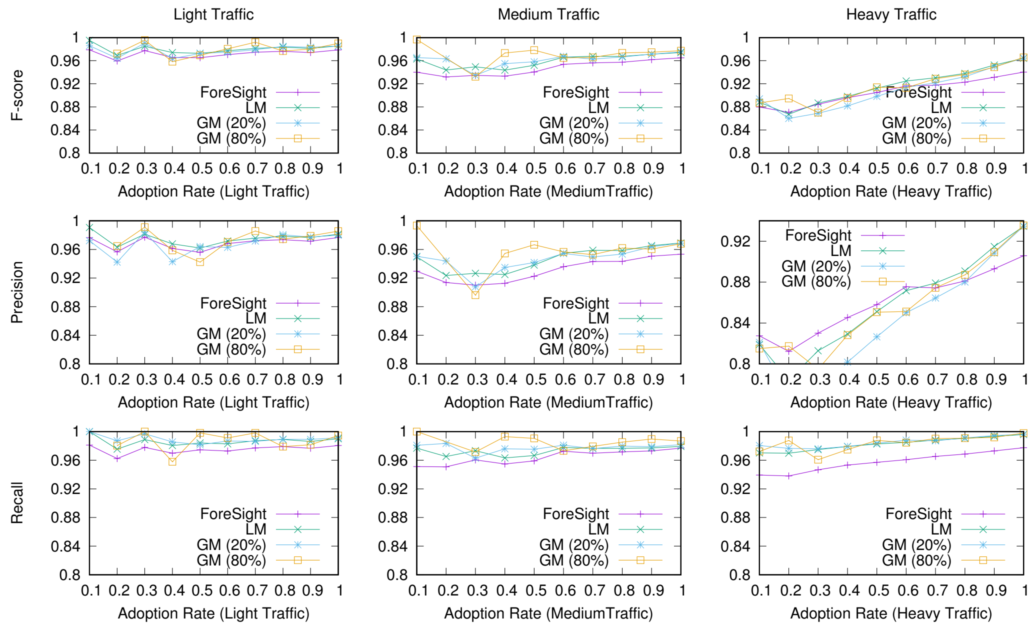

Fig. 4: GM preserves the precision and recall in fusing the different LM results with almost zero LM result fusing error.

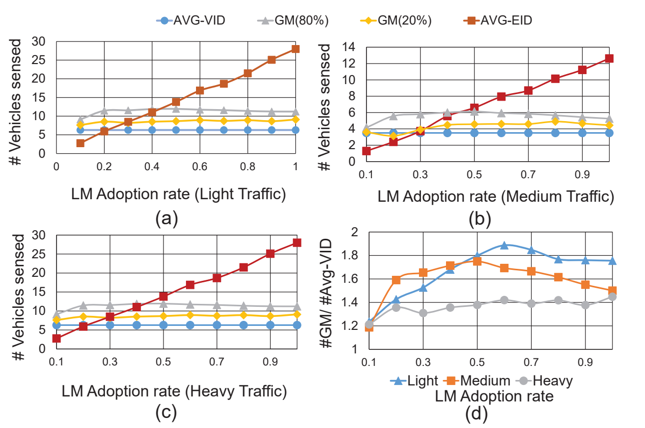

Fig. 5: The number of vehicles sensed by different algorithms (average degree of the reporter nodes in global structure $\mathcal{G}$). GM improves sensing by a factor of 2.

creating the association graph, which significantly reduces the size of the association graph and computational complexity.

GM combines different pieces of information to create the global structure, therefore, our problem has similarity with the image stitching problem and the jigsaw puzzle problem. The image stitching problem [12] needs to discover the correspondence relationships among images with varying degrees of overlap. It is used in video stabilization and the creation of panoramic images. In our problem, we need to identify the EID of the vehicles. Besides the jigsaw puzzle games, the jigsaw puzzle problem is also applied in document and archaeological artifact reconstruction [7]. Solutions of the jigsaw puzzle includes matching the share, edges, patterns or colors of the non-overlapping pieces to reconstruct the global picture. In our case, the detection results cannot be represented by non-overlapping pieces. In GM, we created a star-like structure for the detection results.

VI. Future Work

The following directions of research are left as future work.

(i) Adaptive Broadcast based on the map: During the simulation, we noticed that in heavy traffic condition, the packet collision rate is high if multiple vehicles broadcast their EIDs at the same time. As one of our future work, we plan to design schemes to reduce packet collision in RoadView. One potential solution is to explore the movement correlation to reduce the number of broadcasts. The movements of the vehicles in single lane with inter-vehicle spacing from zero to about 125 meters are correlated [17]. If two consecutive vehicle pairs on a single lane are maximum separated at one location, (e.g., when one is accelerating and the other is drifting), then there is a higher chance the succeeding vehicles will be separated similarly. Therefore, as our future work we want to adaptively select the broadcast time-window and broadcast-rate by using the correlations between the vehicles. This way the matching of vehicles can be very power efficient and cause less number of collisions.

(ii) Real-time sensor data platform: During the design and implementation of RoadView, we noticed that one challenge in implementing vehicular applications is the lack of a platform that has access to real-time sensor data. Therefore, as our future work, 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 [2] and Ford (OpenXC [1]), provide

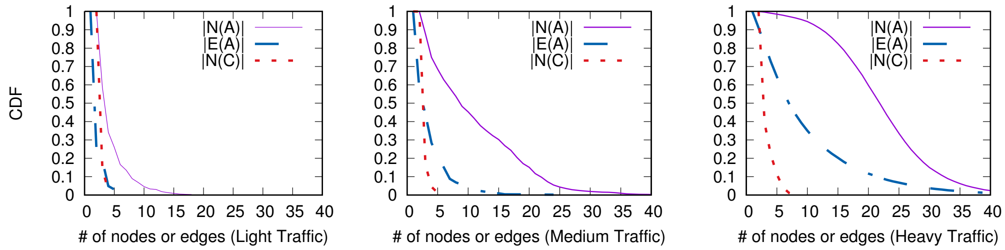

Fig. 6: The CDF of the number of nodes and edges in association graph $\mathcal{A}$ and the number of nodes in the detected maximum weighted clique $\mathcal{C}$ when $r = 100\%$ and the adoption ratio is 100%.

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. RoadView proposed in this work can be one example of the service APIs.

VII. Conclusion

RoadView is a system that builds a live map of the surrounding vehicles by collaboratively fusing local maps created by vehicles. RoadView layers on top of local matching algorithms such as Foresight [10] or RoadMap [20] and improves sensing capabilities of vehicles by a factor of 1.8x. RoadView can work even at low adoption rates and can also map the legacy vehicles. The extended sensing range can benefit collaborative vehicular applications related to traffic statistics, safety by accident prediction and prevention, and energy efficient route planning.

VIII. Acknowledgments

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

References

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

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

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

[4] Immanuel M Bomze, Marco Budinich, Panos M Pardalos, and Marcello Pelillo. The maximum clique problem. In *Handbook of combinatorial optimization*, pages 1–74. Springer, 1999.

[5] Coen Bron and Joep Kerbosch. Algorithm 457: finding all cliques of an undirected graph. *Communications of the ACM*, 16(9):575–577, 1973.

[6] Frédéric Cazals and Chinmay Karande. A note on the problem of reporting maximal cliques. *Theoretical Computer Science*, 407(1):564–568, 2008.

[7] Florian Kleber and Robert Sablatnig. A Survey of Techniques for Document and Archaeology Artefact Reconstruction. In *Proc. of Document Analysis and Recognition*, pages 1061–1065. IEEE, 2009.

[8] Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch, and Laura Bieker. Recent development and applications of SUMO - Simulation of Urban MObility. *International Journal On Advances in Systems and Measurements*, 5(3&4):128–138, December 2012.

[9] Dong Li, Tarun Bansal, Zhixue Lu, and Prasun Sinha. MARVEL: Multiple Antenna based Relative Vehicle Localizer. In *Proc. of ACM MOBIMCOM*, Mobicom, pages 245–256, New York, NY, USA, 2012. ACM.

[10] Dong Li, Zhixue Lu, Tarun Bansal, Erik Schilling, and Prasun Sinha. ForeSight: Mapping Vehicles in Visual Domain and Electronic Domain. In *Proc. of IEEE INFOCOM*, 2014.

[11] Hassnaa Moustafa and Yan Zhang. *Vehicular networks: techniques, standards, and applications*. Auerbach publications, 2009.

[12] Nikos Paragios, Yunmei Chen, and Olivier D Faugeras. *Handbook of mathematical models in computer vision*. Springer, 2006.

[13] Panos M Pardalos and Jue Xue. The maximum clique problem. *Journal of global Optimization*, 4(3):301–328, 1994.

[14] Ryan Parker and Shahrokh Valaee. Vehicle Localization in Vehicular Networks. In *Vehicular Technology Conference*, pages 1–5. IEEE, 2006.

[15] Apostolos P Psyllos, Christos-Nikolaos E Anagnostopoulos, and Eleftherios Kayafas. Vehicle Logo Recognition Using a SIFT-Based Enhanced Matching Scheme. *IEEE Transactions on Intelligent Transportation Systems*, 11(2):322–328, 2010.

[16] John M Robson. Finding a maximum independent set in time $O(2^n/4)$. Technical report, LaBRI, Université de Bordeaux I, 2001.

[17] Richard W Rothery. Car following models. *Trac Flow Theory*, 1992.

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

[19] Gopi Krishna Tummala, Derrick Cobb, Prasun Sinha, and Rajiv Ramnath. Poster: Soft-Swipe: Enabling High-Accuracy Pairing of Vehicles to Lanes using COTS Technology. In *Proc of ACM CarSys 2016*, pages 62–63, .

[20] Gopi Krishna Tummala, Dong Li, and Prasun Sinha. RoadMap: Mapping Vehicles to IP Addresses using Motion Signatures. In *Proc of ACM CarSys 2016*, pages 30–37, .

[21] Gopi Krishna Tummala, Derrick Ian Cobb, Prasun Sinha, and Rajiv Ramnath. Methods and apparatus for enabling mobile communication device based secure interaction from vehicles through motion signatures, 2016. US Patent App. 15/060,494.