2015 International Conference on Computing, Networking and Communications, Mobile Computing & Vehicle Communications Symposium

OmniView: A Mobile Collaborative System for Assisting Drivers with a Map of Surrounding Traffic

Rufeng Meng[†], Srihari Nelakuditi[†], Song Wang[†], Romit Roy Choudhury[‡]
[†]University of South Carolina [‡]University of Illinois at Urbana-Champaign
{mengr, srihari, songwang}@cse.sc.edu, croy@illinois.edu

***Abstract*—To assist drivers and prevent collisions, we propose a system called *OmniView* that extends driver's vision in all directions, using cameras of multiple collaborating smartphones in the surrounding vehicles. OmniView provides a driver with a traffic map about the relative positions of surrounding vehicles. Under OmniView, each vehicle detects other vehicles in its view, estimates their relative positions, and broadcasts its local map. Upon reception of a map from another vehicle, a vehicle updates its own map by fusing it with the received map. A key issue faced by OmniView is, how does a vehicle address another vehicle in its map? We propose that a vehicle's image itself could be treated as its address. However, if we include images in each map message, the communication overhead will be high. Towards that end, OmniView resolves a vehicle's image to a small unique ID. With this approach, we demonstrate that it is feasible to develop the OmniView system that produces a traffic map in real-time. Besides, through computer vision techniques and the collaboration between vehicles, OmniView could show the positions of surrounding vehicles with reasonable accuracy on the map. Such a traffic map, even without being displayed to the drivers, can act as the common substrate based on which various alerts can be triggered to avoid accidents.**

I. Introduction

A common cause behind most accidents is that drivers fail to notice the presence of surrounding vehicles and maintain a safe distance from them. Therefore, it will be beneficial to drivers if the vehicles have a map of traffic around them. Such a traffic map, even without being made visible to the drivers, can help trigger acoustic alerts to prevent collisions. Indeed, some advanced driver assistance systems are being actively developed [1]. But, they tend to be pricey and available only in the latest luxury vehicles such as Mercedes Benz GL SUV. Our goal is to bring similar safety features to drivers of legacy and economy vehicles. Towards that end, we propose a system called *OmniView* that extends the vision of drivers in all directions using cameras of multiple collaborating smartphones.

OmniView provides a vehicle with a traffic map about the relative positions of its neighboring vehicles (see Fig. 1). An obvious way to obtain such a map is to use GPS and have each vehicle report its position. Then, a vehicle can determine its distance to other vehicles. Unfortunately, GPS error for civilian usage could be up to 30 meters [2], which is too high to distinguish lane-level position. Although Differential GPS [3] could help reduce the error, it requires deploying reference stations all along the road. Furthermore, GPS based approach requires every vehicle's participation. If a vehicle does not report its GPS position, other vehicles would not even be aware of its presence. Therefore, we propose to utilize the cameras of smartphones and computer vision techniques to gather the traffic map, without requiring full participation from all surrounding vehicles. For instance, in Fig. 1, even if B does not participate, other vehicles like D or F can help E in making it aware of B's presence.

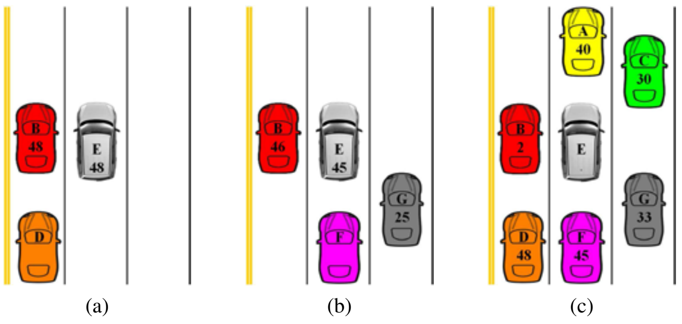


Fig. 1. Local maps at (a) vehicle D and (b) vehicle F. (c) The map at E after merging its local map with the received maps from D and F. When E broadcasts its map, D and F also learn about the surrounding traffic. The numbers shown on the vehicles tell the distance in meters.

The operation of OmniView can be briefly described as follows. A smartphone, running OmniView app, mounted on the dashboard or windshield of a vehicle, with its camera facing forward, detects vehicles in front of it and estimates their relative positions. Next, it sends a broadcast so that smartphones in nearby vehicles can receive this information[1]. For example, in Fig. 1(a), vehicle D computes positions of B and E in its view and broadcasts this map. Upon receiving a similar map from F (Fig. 1(b)), E can fuse the received information together to form a map of surrounding vehicles, as in Fig. 1(c). It can then broadcast this map so that D and F can learn about the traffic around them. The traffic map, thus obtained, can help trigger alerts to prevent collisions.

To construct such a traffic map and make it practically useful, three issues need to be addressed. First, OmniView-enabled vehicle should be able to address the neighboring vehicles. When two vehicles meet on the road, they have no knowledge about each other's IP or MAC address. License number shown on the license plate might be an option, but a vehicle's license number is not legible from another's view beyond a certain limited distance and angle. In OmniView, we use vehicle's image as its address. Every vehicle knows its own appearance (here we call it *self-image*[2]). When a vehicle, say F in Fig. 1(b), sees E and broadcasts a message with E's image, E could recognize that F is talking to it.

[1]For convenience, we say that vehicles are computing and communicating, though in reality smartphones are running the OmniView app.

[2]We assume that the OmniView app has the host vehicle's images (referred to as self-images), taken from multiple directions/distances by the driver.

Second, during map exchange (which happens frequently), if OmniView always uses images to represent every vehicle in the map, the communication overhead will be huge and the system will be very inefficient. To solve this problem, we construct a mapping between each vehicle's image and a short unique ID (which could be a hash of its license number or VIN). Every vehicle knows its self-image and ID; when another vehicle inquires its ID with its image, it announces its ID. Later, other vehicles use the ID to represent this vehicle, which reduces communication overhead significantly. Last, as a traffic map, OmniView needs to locate the vehicles accurately. We use computer vision techniques to estimate the relative distances and directions between vehicles, and form complete and accurate map through map exchange and fusion.

In the following, we first describe the design of OmniView. Then, we present our evaluation, which shows that producing a traffic map in real-time with OmniView is practical.

II. Design of OmniView

The OmniView system, as shown in Fig. 2, can be divided into five functional parts: Vehicle Detection, Vehicular Communication, Image Matching, Position Calculation, and Map Computation. Below, we describe each of these parts.

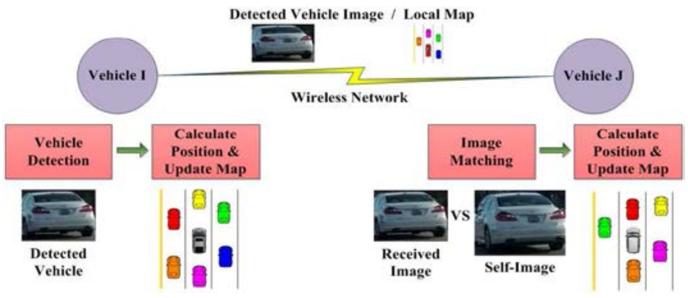


Fig. 2. OmniView system workflow: vehicle I detects vehicle J. It calculates the position of vehicle J and updates its local map to show vehicle J. Upon receiving the detected vehicle image from vehicle I, vehicle J estimates the relative position of vehicle I and shows vehicle I on its local map. Vehicles also regularly exchange their local maps to help each other form more complete and accurate map about neighboring vehicles.

A. Vehicle Detection

When a smartphone is mounted with its camera facing forward, it could see the whole scene in front of the ego vehicle. In OmniView, the first thing the smartphone needs to do is to detect vehicles in its camera view. For that, we train a Haar Cascade classifier [4] [5] to detect vehicles. It works as shown in Fig. 3. Once OmniView detects a vehicle, it extracts the part of the image containing the detected vehicle (in the green rectangle). The extracted image becomes the visual identifier for the detected vehicle, which will be used in position calculation, map computation, and communication.

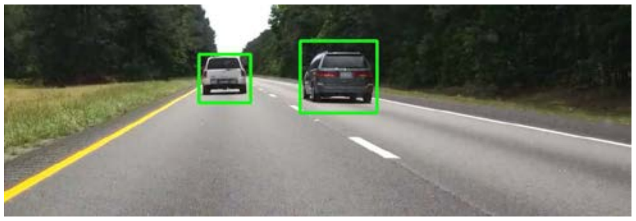


Fig. 3. OmniView detects vehicles moving in front of the ego-vehicle and extracts the images about the detected vehicles.

B. Vehicular Communication

In OmniView, vehicles collaborate with each other to exchange surrounding traffic maps over DSRC [6]. But, when two vehicles meet, they don't have any knowledge about each other's conventional address (such as MAC address or IP address). So, OmniView uses the vehicle's image itself as its address. However, if we include an image in each message, the communication overhead will be high. Therefore, OmniView constructs a mapping from the image of a vehicle to a short ID such as its license number (e.g. CA77CD88), which is automatically unique. Though vehicles travel fast on a highway, due to their low relative speeds, they usually stay in contact with each other for a few seconds to several minutes. During the contact time, once a vehicle knows another vehicle's ID, it does not need to frequently transmit that vehicle's image.

There are two types of messages being exchanged in OmniView system. One is Identify message, which is used to resolve the ID of a vehicle from an image; the other is Map message. Fig. 4 shows the message exchange. A vehicle, say I, sends an Identify message (the question in the figure) with an image of a vehicle, say J, when it tries to resolve the ID of J. Vehicle J responds with its ID, along with its own local map. Then, vehicle I records the mapping between J's image and ID, and also merges J's map with its own map. In case a vehicle does not receive a response to its query, after a few, say two, attempts, it assumes that the target vehicle is a non-participant and associates "unknown" ID to that image, to minimize futile queries. When Identify and Map messages are being exchanged between vehicles I and J, other surrounding vehicles, such as K, that overhear these messages, will also record Image-ID mapping for J and update their maps.

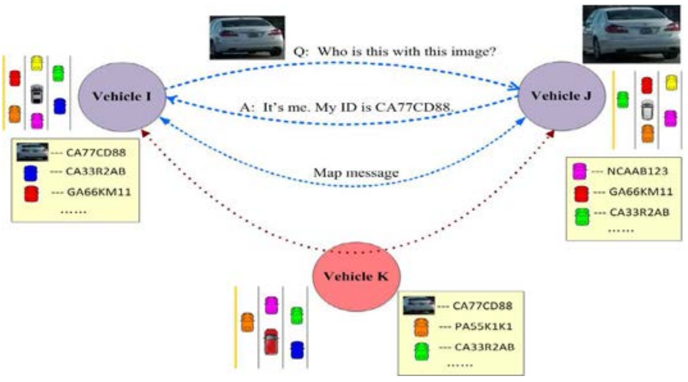


Fig. 4. Message exchange in OmniView: Vehicle I detects J, sends an Identify message to J, and J responds with its ID. I and J then exchange Map messages. K could overhear these messages and update its local map. Each vehicle caches Image-ID mapping for vehicles nearby.

C. Image Matching

When a vehicle receives an Identify message, it needs to compare the received image with its self-images. Furthermore, when a vehicle detects another vehicle, it has to match the detected image with the images stored in the cached Image-ID mapping, to check whether it already resolved the ID for this vehicle. In OmniView we achieve this through image matching. OmniView needs to run on a smartphone and perform image matching as quickly as possible. Based on our testing, we observed that ORB [7]-based image matching algorithm could run efficiently on smartphones. So, we choose ORB to do image matching.

D. Position Calculation

The relative position of a vehicle could be specified with <lane, distance>. Below, we describe how one vehicle determines another vehicle's lane and estimates distance to it.

1) Lane Determination: The OmniView system needs to determine the lane-level position of vehicles, i.e. who is in front and who is on the adjacent left lane, etc. We adopt two strategies to get the lane position information.

The first strategy is to extract the lane markings (as in Fig. 5) to determine relative lane positions of ego-vehicle and the detected vehicles. This lane position information is included in Identify message, which not only helps the receiver get the sender's position, but also helps a receiving vehicle filter Identify messages not meant for it without performing image matching. This saves significant computational overhead.

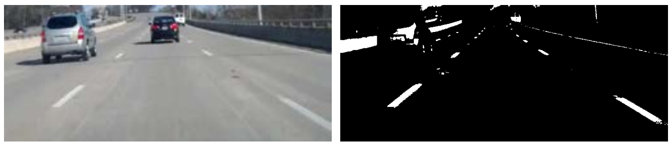


Fig. 5. (Left) Road with multiple lanes. (Right) Extracted lane markings that help identify the lane positions of the ego-vehicle and the detected vehicles.

The second strategy for determining relative position of vehicles is from image matching. When one vehicle receives an image, it matches that image with its left-side, rear-end, and right-side self-images. The best match indicates which side the sending vehicle is. Table I shows the number of matched feature points between two sets of images taken at different distances and from two directions (left-side and rear-end) by two different phones. Clearly, same-direction images have many more matched features than different-direction images.

TABLE I. MATCHING IMAGES FROM DIFFERENT DIRECTIONS

Distance(m)	Rear vs. Rear	Rear vs. Left	Left vs. Left
10	386	75	542
20	191	50	256

When the lane markings are not very clear and the exact lane position of vehicles can not be determined, OmniView uses image matching based method as a fallback option.

2) Distance Estimation: We compute the distance between vehicles in two ways: The first method is used when a vehicle detects another vehicle and the second method is applied when a vehicle receives an Identify message with its image.

When a vehicle is detected, the distance from the ego-vehicle to the detected vehicle (Z), the focal length (f), the width of the vehicle in the camera view (w) and the actual width of the vehicle (W) satisfy the relationship (see Fig. 6):

$$\frac{W}{Z} = \frac{w}{f} \quad (1)$$

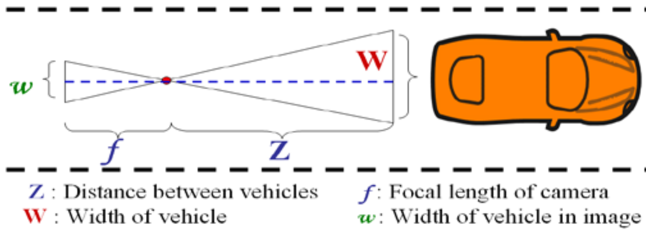
Here, focal length f of the smartphone's camera could easily be obtained through SDK, whereas w depends on the pixel size of the camera sensor, which is hard to obtain. We overcome this problem by having the driver take an image of her own vehicle, with width W_0, from a known distance Z_0, for calibration of the camera. From this self-image, we have Eq. (2). When using the system, once it detects other vehicle in front, we have Eq. (3). In (3), W_{new} is width of the detected vehicle and Z_{new} is the distance between the ego-vehicle and the detected vehicle. w_{new} is the width of the detected vehicle in the camera sensor.

$$\frac{W_0}{Z_0} = \frac{w_0}{f} \quad (2) \qquad \frac{W_{new}}{Z_{new}} = \frac{w_{new}}{f} \quad (3)$$

From Eq. (2) and Eq. (3), we could get the distance Z_{new}:

$$Z_{new} = \frac{W_{new}}{W_0} \times \frac{w_0}{w_{new}} \times Z_0 \quad (4)$$

In Eq. (4), $\frac{w_0}{w_{new}}$ is the ratio of the pixel numbers of ego-vehicle and the detected vehicle, regardless of the pixel size. The only unknown is W_{new}. When the dimensions of the detected vehicle are not known, OmniView chooses a default value for W_{new}. Once the detected vehicle responds to the Identify message, its actual width will be available and can be used to calculate the distance more accurately.


w
W
f
Z
Z : Distance between vehicles
f: Focal length of camera
W : Width of vehicle
w: Width of vehicle in image
Fig. 6. Relationship between vehicle and its image in camera.

When a vehicle receives Identify message with its image from another vehicle, it can estimate the distance to the sender using a method based on image matching.

Initially, when a driver takes her own vehicle's self-images from known distances, the self-images satisfy Eq. (5). Here w_0 is the width of the vehicle in a self-image and Z_0 is the known distance when driver takes the self-image. f_A is the focal length of the corresponding smartphone camera.

$$\frac{w_o}{f_A} = \frac{W}{Z_0} \quad (5) \qquad \frac{w_{new}}{f_B} = \frac{W}{Z_{new}} \quad (6)$$

When the vehicle is detected by another vehicle, we have Eq. (6). Here w_{new} is the width of the vehicle in the detected image; f_B is the focal length of the smartphone camera in that vehicle; Z_{new} is the distance between the two vehicles.

From equation (5) and (6), we could get:

$$Z_{new} = \frac{w_0}{w_{new}} \times \frac{f_B}{f_A} \times Z_0 \quad (7)$$

In this equation, $\frac{w_0}{w_{new}}$ could be derived from image matching. When matching two images, the scale ratio of the two matched objects (here vehicles) could be calculated. f_A and f_B are usually different, unless the models of the two smartphones are identical. For different smartphone models, we could pre-measure $\frac{f_B}{f_A}$ for all the popular smartphone pairs on the market, and then pre-install these ratios in OmniView. To do that, we collect images taken at same distances with different phones, and then match these same-distance images to get the ratios between different phone models. Fig. 7 exemplifies the focal ratio of different phone models. In communication, sender tells

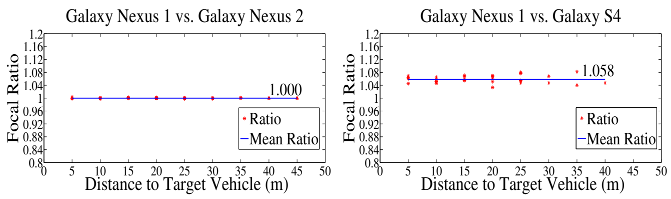


Fig. 7. Focal ratio between phones. Pairs of images for the same vehicle are taken by compared phones at each distance (10 to 45m) and direction (Rear, Left, Right). Each pair of images are matched to calculate the focal ratio between different phones. Left shows the ratio between two Galaxy Nexus. These two are same model phones, so the focal ratio is 1.0; Right shows the ratio between Galaxy Nexus and Galaxy S4.

the opposite the smartphone model it is using. Therefore the receiver could select proper $\frac{f_B}{f_A}$ to calculate the distance.

For communication purpose, we need to scale down the original image to be a smaller size. For instance, when sender detects receiver at distance $Z_{original}$, the size of the receiver vehicle in the original image is $w_{original}$, so we have Eq. (8).

$$\frac{w_{original}}{f} = \frac{W}{Z_{original}} \quad (8) \qquad \frac{w_{scaled}}{f} = \frac{W}{Z_{scaled}} \quad (9)$$

After scaling down, the size of the receiver vehicle in the resulting image becomes w_{scaled} (which is the w_{new} in (7) from the receiver's point of view) and the distance $Z_{original}$ becomes Z_{scaled} which meets Eq. (9) (here the scaling down is carried out at the sender, so f is unchanged, which is the focal length of the sender's smartphone camera):

So, Z_{new} computed by equation (7) is indeed Z_{scaled} in equation (9). From the equations (8) and (9), we get:

$$Z_{original} = Z_{scaled} \times \frac{w_{scaled}}{w_{original}} \qquad (10)$$

In this equation, $\frac{w_{scaled}}{w_{original}}$ is the scale factor S, which is transmitted along with the image.

By combining equation (7) and (10), the actual distance could be derived from the following formula:

$$Z = \frac{w_0}{w_{new}} \times \frac{f_B}{f_A} \times Z_0 \times S \qquad (11)$$

E. Map Computation

From vehicle detection, OmniView could obtain the lane position and distance of the detected vehicle, hence put the detected vehicle in its local map. Besides, vehicles periodically exchange local maps to help each other form more complete and accurate maps. Each time a vehicle receives a map from another vehicle, it searches for common nodes between its local map and the received map. If a common node exists, OmniView fuses the two maps by adding the distinct nodes from the received map into its local map. Map fusion is done through geometric computing based on the common node and the position information provided in the received map. For example, in Fig. 1, initially in E's local map, only A, C and E exist (E detects A and C). Meanwhile, F detects B, E and G; D detects B and E. When E receives map from F, it locates F, B, G into its local map through geometric computing based on the common node E of the two maps and the position information provided in F's map. Fig. 8 illustrates how vehicle B from F's map is added into E's local map. Here, d_{EF} and d_{BF} are derived from F's map, and w is the standard lane width, which could easily be known or estimated. Then, OmniView can compute the value of h, whereafter, by referring to d_{EF}, OmniView could locate vehicle B into E's local map.

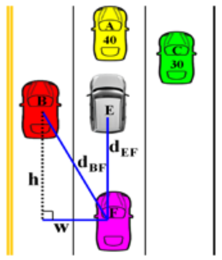


Fig. 8. Locate vehicle from received map into local map.

Each time OmniView updates the local map, it checks whether any potential hazards exist. We define different levels of alerts for different hazards, from flashing icon on the map to acoustic warning when the danger of collision increases based on the positions of neighboring vehicles.

III. Evaluation

Since OmniView uses a vehicle's image to identify it, the resolution/size of the image affects the matching accuracy and communication overhead. Therefore, first, we find a suitable size to be included in Identify messages, and then evaluate the accuracy of distance estimation, reliability of vehicular communication, and the overall efficiency of OmniView.

A. Image Size and Matching Confidence

To study the trade-off between image size and matching confidence, we collect two sets of vehicle images. The first set contains 530 pairs of images and each pair contains two images for the same vehicle. The second set contains 1090 pairs and the images in each pair are for different vehicles. The results of image matching for these two sets are shown in Fig. 9. It is evident that if two images are for different vehicles, the number of matched feature points is rarely above 10. On the other hand, if two images are for the same vehicle and the image size is bigger than 10 KB, more than 20 features points match. Therefore, by choosing a threshold like 15 for the number of matched points and image sizes 10∼14 KB, OmniView could identify a vehicle with high accuracy through image matching. When OmniView detects a vehicle, it scales down the detected vehicle image to this range, to reduce communication overhead.

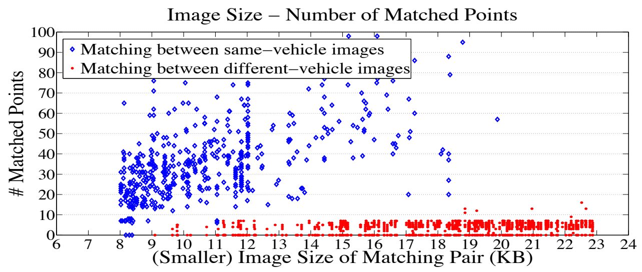


Fig. 9. Image matching between pairs of same-vehicle and different-vehicle images (x-axis shows the size of the smaller image in each pair).

B. Distance Estimation

To assess the accuracy of distance estimation by OmniView, we took pictures of 10 vehicles, with four phones (Galaxy S4, iPhone 4S and two Galaxy Nexus) from 3 directions (left, right, and rear) and 10 different distances (ranging from 5 to 50 meters at increments of 5 meters). We kept the vehicles stationary in this preliminary evaluation, since it is not easy to obtain the ground truth on distance between vehicles, when both are moving. The images taken by one Galaxy Nexus (named as Galaxy Nexus 1 here) are treated as self-images. The focal ratio between phones is precomputed as described earlier in Section II-D2. The measured distance vs. ground truth is plotted in Fig. 10. It shows that OmniView can estimate the distance accurately, particularly up to 45 meters. Admittedly, these results, while very encouraging, are obtained when the vehicles are stationary. A part of our on-going work is to assess the accuracy of OmniView in real driving scenarios.

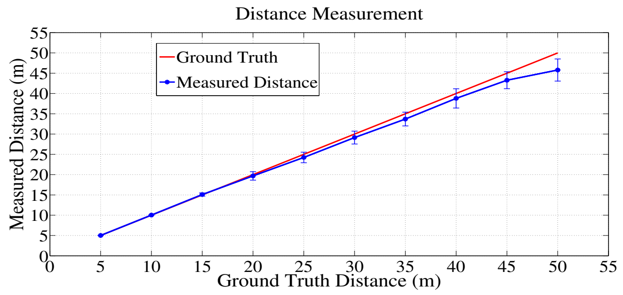


Fig. 10. Distance estimated by OmniView compared with the ground truth.

C. Vehicular Communication

Conducting a large-scale evaluation of vehicular communication on the road is very difficult and requires large amounts of resources that are beyond our reach. So in this work, we use simulation to study the performance of the vehicular communication in the OmniView system. We use SUMO [8] to generate the vehicle mobility, which is fed into NS2 to carry out the network simulation. The details of the parameters used in our simulation setting are listed in Table II.

TABLE II. Simulation Setting

Parameter	Remark
Simulation Period	200 s
Number of Vehicles	300 (4 types of vehicles with different length, speed, acceleration, and deceleration.)
Speed	20∼30 m/s (45∼67 mph)
Traffic Density	Sparse (54 vehicles/km) Dense (94 vehicles/km)
Wireless Protocol	DSRC 802.11p
Antenna Type	OmniAntenna
Radio Propagation Model	Two Ray Ground
Data Rate	6 Mbps (QPSK) DSRC could support up to 27 Mbps, but [9] shows that 6 Mbps performs the best.
Message	Identify (10∼14 KB, sliced into 1-KB packets) Map (512 B)
Message Life Time	Identify message: 0.6 s Map message: 0.4 s
Transmission Frequency	Identify message: 0∼3 images every 3.5±2.0 s [3] Map message: Once every 0.4±0.2 s [4]
Transmission Range	60 meters

We measure the message reception rate, which is the ratio of number of nodes that received the message and number of nodes in sender's transmission range. The message reception rate for Identify and Map messages in two traffic modes are shown in Fig. 11. We can see that Map message could be exchanged very reliably in both sparse and dense traffic modes.

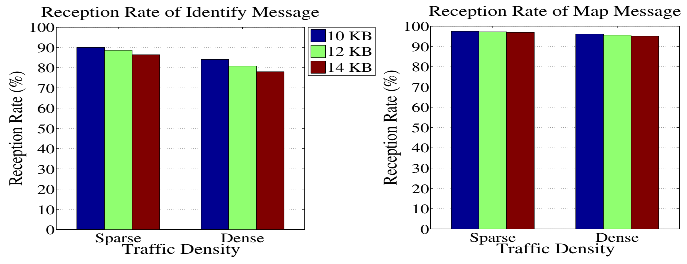


Fig. 11. Reception rate of (a) Identify and (b) Map messages.

In case of Identify message, OmniView could achieve about 80% reception rate in dense mode and up to 90% in sparse mode. Since OmniView is a collaborative system, every vehicle will likely be detected by more than one vehicle moving behind it (in the same lane or different lanes). All these vehicles moving behind a vehicle will send Identify messages to it. Thus each one will have more chances than what is shown in Fig. 11 to receive at least one of those Identify messages. Once it receives one and announces its ID, the vehicles behind it could construct the Image-ID mapping. Therefore, the subsequent Identify messages related to this vehicle will turn into Map messages, which could be received more reliably. So with OmniView, every vehicle will have high probability to obtain the positions of its neighboring vehicles.

From the simulation, we also measured the end-to-end communication latency for both Identify and Map messages, which is shown in Fig. 12. While the delay associated with the Map message is insignificant, in the order of 1 ms, that with Identify message is not high, mostly below 40 ms.

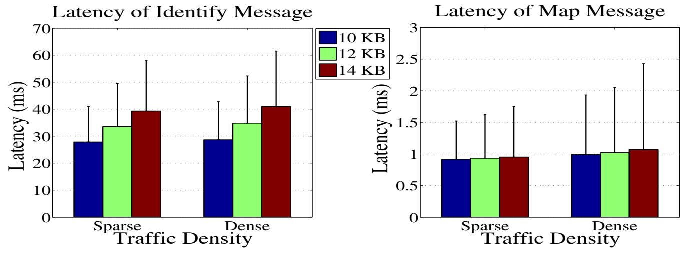


Fig. 12. Communication latency of (a) Identify and (b) Map messages.

D. Computational Overhead

In the OmniView system, image matching is the most computation-intensive work. If OmniView is busy with image matching all the time, the system might not be able to reflect what is happening around in real-time. Here, we examine the computational overhead of OmniView.

Fig.13 shows how many Identify messages each vehicle receives per second on average, which corresponds to how much image matching work each OmniView-enabled vehicle needs to do. Although each vehicle receives around 4 images per second in dense mode, with the knowledge of lane position (as described in Section II-D), OmniView could easily filter out the Identify messages which are not targeting it. So it does not need to do image matching on all the received images. Besides, smartphone's CPU is becoming more and more powerful. With many of them having multi-cores nowadays, the image matching task could be carried out in parallel on multiple cores, so the time on image matching will become even less.

[3]On highway, most relative speeds between vehicles are less than 20 mph (i.e. 9 m/s), so it takes $\geq$ 6.6s to travel 60 meters. Even with the maximum interval—5.5s, if one vehicle detects another, it has at least two chances to send Identify message to resolve the ID.

[4]For Map message, on average, within 0.4s, the distance between two vehicles changes less than 3.6 meters, so the hazard will not change severely between two Map messages. Each vehicle would have sufficient time to learn about the hazard if it occurs.

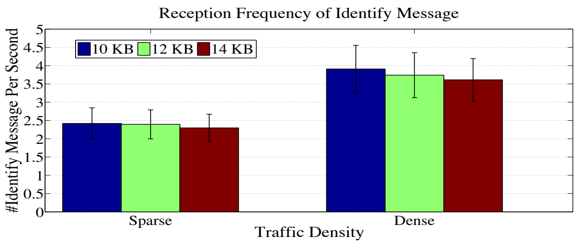


Fig. 13. The number of Identify messages each OmniView-enabled vehicle receives every second.

E. Overall Efficiency

In the OmniView system, time is mainly consumed in three parts: vehicle detection, message transmission and image matching. We measured the vehicle detection on a real road. When vehicle occurs in the video frame, OmniView could detect the vehicle within 111 $\pm$ 60 ms (measured on a Galaxy Nexus). To measure the time of image matching, we prepared 406 vehicle images with the sizes range from 9 KB to 20 KB. Among the 406 images, some of them are for the same vehicle, others are for different vehicles. We measured the time of image matching with all possible image pairs on a Galaxy Nexus and a Galaxy S4; Table III shows the time taken.

TABLE III. TIME TAKEN FOR IMAGE MATCHING

Phone	Time
Galaxy Nexus	195$\pm$74 ms
Galaxy S4	108$\pm$44 ms

From the above measurement, we can expect that in OmniView, the total time spent on vehicle detection, communication and image matching is about 400 ms, which is real-time. Here, Galaxy Nexus is a relatively old phone (released in 2011). Galaxy S4 (released in March, 2013) is more powerful. It is reasonable to expect that, over time, as more powerful phones come out, OmniView would perform more efficiently.

IV. LIMITATIONS AND FUTURE WORK

Since OmniView is based on computer vision techniques, light conditions could affect vehicle detection and also the feature points extraction in image matching. One way to make OmniView work under poor light conditions is to detect vehicles by the car lights (different car models usually have differently-shaped lights) and use the width between the left and right rear-end lights to estimate the distance to the detected vehicle. However, that requires further study and we concede that current OmniView is effective only during the daytime.

Another limitation of OmniView is that when two vehicles have the same appearance, both of them might respond to an Identify message. We observed the traffic on a highway for 2 hours, and find out that there are only 3 out of 1000 scenarios where same-looking vehicles occur in another vehicle's communication range. But in city, we believe the probability will be higher. We have partially addressed this problem by including the lane positions of the ego-vehicle and the target vehicle in the Identify message. This could solve the problem when two same-looking vehicles are on different lanes. But if they are moving together on the same lane (one behind another), OmniView can not distinguish them.

Although we illustrated in Fig. 1 that OmniView does not require full participation of vehicles, we haven't studied the lowest participation rate needed, which will be our future work.

V. RELATED WORK

Some vehicles nowadays are equipped with radar and/or camera-based systems to assist drivers in detecting the objects around the vehicles. But most of these systems are designed for short distance and low speed, usually helping the driver in backing or parking [1]. Researchers have studied various methods to help drivers be aware of their surroundings. Authors in [10] mount cameras near the side mirrors, to detect vehicles moving in front and on two sides. The work in [11] also suggests a vision-based system to discover surrounding vehicles. It requires the multi-camera system to be mounted on top of the vehicle. These systems requires particular hardware to be installed at some unusual place and work in isolation. GPS-based solutions [12] [13] [14] [15] suffer from position error (could not provide lane-level accuracy) and the requirement of high penetration rate. Compared with the multi-camera or radar-based standalone systems and GPS-based collaborative systems, OmniView can provide relative position information of neighboring vehicles even with partial participation.

VI. CONCLUSION

In this paper, we introduced OmniView, a smartphone-based collaborative system for assisting drivers. With OmniView, each vehicle detects other vehicles and estimates their positions to form a local traffic map. The vehicles exchange the detected vehicle images as well as their local maps to help each other form a more complete and accurate map with the positions of neighboring vehicles. Our evaluation shows that OmniView could work reliably and provide a map of the traffic surrounding a vehicle in real-time.

REFERENCES

[1] "Surround view systems," http://www.mobileye.com/technology/applications/surround-view-systems/.

[2] "Gps accuracy," http://www.gps.gov/technical/ps/2008-SPS-performance-standard.pdf.

[3] "Differential global positioning system," http://en.wikipedia.org/wiki/Differential_GPS.

[4] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *CVPR 2001.*

[5] R. Lienhart and J. Maydt, "An extended set of haar-like features for rapid object detection," in *Image Processing*, vol. 1, 2002.

[6] J. B. Kenney, "Dedicated short-range communications (dsrc) standards in the united states," *Proceedings of the IEEE*, vol. 99, 2011.

[7] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "Orb: an efficient alternative to sift or surf," in *Computer Vision (ICCV)*, 2011.

[8] "Sumo simulator," http://sumo-sim.org/.

[9] D. Jiang, Q. Chen, and L. Delgrossi, "Optimal data rate selection for vehicle safety communications," in *VANET 2008.*

[10] T. Gandhi and M. Trivedi, "Dynamic panoramic surround map: motivation and omni video based approach," in *CVPR Workshops, 2005.*

[11] H. Cheng, Z. Liu, N. Zheng, and J. Yang, "Enhancing a driver's situation awareness using a global view map," in *Multimedia and expo, 2007.*

[12] B. Xu, O. Wolfson, and H. J. Cho, "Monitoring neighboring vehicles for safety via v2v communication," in *ICVES 2011.*

[13] A. Corti *et al.*, "A centralized real-time advanced driver assistance system based on smartphones."

[14] J. A. Misener *et al.*, "Cooperative collision warning: Enabling crash avoidance with wireless technology," in *12th World Congress on ITS*, 2005.

[15] J. Yang, J. Wang, and B. Liu, "An Intersection Collision Warning System using Wi-Fi Smartphones in VANET," in *GLOBECOM*, 2011.