2016 8th International Conference on Communication Systems and Networks (COMSNETS)

DriverTalk: Enabling Targeted Communication Between Drivers

Rufeng Meng*, Romit Roy Choudhury†, Song Wang*, Srihari Nelakuditi*
*University of South Carolina,†University of Illinois at Urbana-Champaign

***Abstract*—Communication between neighboring drivers on the road is critical for safe and smooth driving. Drivers currently communicate by sounding horn or blinking lights, which sometimes is too coarse-grain to precisely convey the message, i.e., it is not apparent who the target and what the intent of the message is. Towards a better alternative, we explore the possibility of utilizing the smartphones of drivers to enable targeted communication between them. We propose a system, called *DriverTalk*, with which a driver can directly talk to convey messages such as you-are-blocking-traffic to the driver of the vehicle in front and do-not-tailgate-me to the driver behind. Furthermore, *DriverTalk* gives drivers the opportunity to seek and provide information as and when necessary during their driving. To identify the senders and receivers while exchanging messages, without an a priori knowledge about the drivers and their smartphones/vehicles, *DriverTalk* utilizes Visual IDs, i.e., the appearance images, of vehicles. Our evaluation shows that *DriverTalk* could deliver what a driver says to the target driver in about 1 second and with more than 95% success rate under various traffic conditions.**

I. Introduction

When driving on the road, in many situations, drivers need to communicate with other drivers of surrounding vehicles, for safe and smooth travel. For example, a driver may want to tell the driver of a vehicle moving behind not to tailgate, because that is dangerous. When a driver wants to change lane, she wants other drivers in the target lane to be aware of her intent and give her space. When driving in an unfamiliar area, drivers typically have the need to acquire some information about this area, for instance, a highway exit to a local landmark.

While there have been many advances in automobile industry, the function allowing drivers to talk to other drivers in-situ is still not there. Current vehicles mainly have two systems allowing drivers to express some maneuver intents. One is light indicator. For example, before changing lane, a driver blinks turn indicator lights. The other is horn. Drivers could honk to alert others. But, neither of these could convey fine-grained information precisely to the targeted drivers. If the targeted driver is not in a position to notice that the other vehicle's light is blinking or illuminated, the function of light indicator becomes ineffective. When a driver honks to alert someone, all other drivers around also hear it and have to figure out whether they are the intended target and what the intended message is, causing unnecessary cognitive load on them. Besides being perceived as an impolite behavior, honking has become a serious source of noise pollution in countries like India [1].

Another common desire and behavior during driving is inquiring information from others. When driving in an unfamiliar area, drivers typically have the need to get to know the direction/route to their destination. Although a navigation device could help, in many countries, especially developing countries like India, map information is incomplete or out-of-date. Some drivers usually stop to ask a local resident, which is not always possible (e.g. moving on highway). If a driver could ask the neighboring drivers, they are very likely to know and provide relevant information. Many drivers currently do that by rolling down the window and shouting to one or more neighboring drivers, which is not only inefficient but also dangerous, because the drivers have to deviate from their driving maneuver.

The core limitation of existing modes of communication between drivers through horns and lights is that they are inadequate in conveying a message to and only to the intended target. To address the need for direct communication between neighboring drivers to convey intents of their maneuvers actively and precisely, as well as a safe and efficient way for drivers to seek and provide necessary information from/to neighboring drivers, we propose a smartphone-based system, *DriverTalk*, leveraging the smartphones of drivers. *DriverTalk* enables a driver to talk to a particular driver, some, or all surrounding drivers, and clearly express her driving intents and/or query necessary information.

A key issue we need to resolve in realizing *DriverTalk* is, how to address the receiver(s) of a message. Note that *DriverTalk* has to allow drivers on the road, with no prior acquaintance, to talk to each other, with zero configuration. We propose to resolve this by utilizing the appearance images of vehicles as their Visual IDs (henceforth we use the term VID) to refer to the senders and receivers of messages. To use the system, each participant takes pictures of her own vehicle from left-back, rear-end and right-back (as shown at upper left of Fig. 1) in advance, which are saved locally in the *DriverTalk*-powered smartphone as self-images. The smartphone is mounted on dashboard or windshield, with its camera watching the vehicles moving in front (as shown at bottom left of Fig. 1). The system uses these self and detected vehicle images as VIDs to represent the senders and receivers of talking messages.

Note that we can only gather the VIDs of vehicles in front. The question then is how to address the other surrounding vehicles. Below, based on who a driver wants to talk to, we sketch how *DriverTalk* addresses them in three typical scenarios.

1) Talking to driver(s) moving behind: Suppose a driver (say in vehicle E in Fig. 1) wants to talk to drivers moving behind (i.e. D, F, G). This could be to notify target vehicles of the traffic situation ahead (e.g. accident), convey intent of maneuver (e.g. changing lane) or even request the driver behind (e.g. request F not to tailgate). Depending on which vehicle moving behind a driver wants to talk to[1], *DriverTalk* broadcasts the self-image(s) taken in the corresponding direction as VID, along with the

[1]In this paper, talking to a vehicle means talking to the driver in that vehicle.

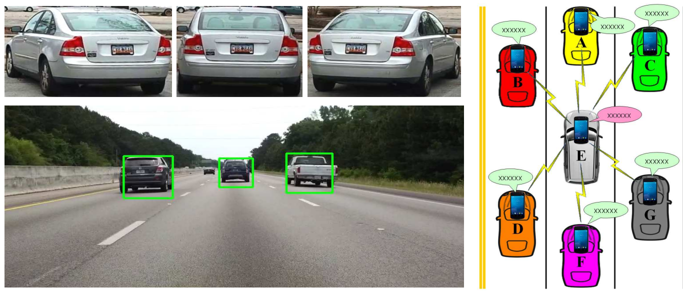

Fig. 1. (Upper-left) Sample self-images of a vehicle. (Bottom-left) Detected vehicle images. (Right) Talking between neighboring vehicles/drivers.

information the driver wants the receiver to know. When the vehicle moving behind receives the message, it compares the received VID with the vehicle images in its view to find out whether one of the vehicles moving in front is talking to it and which one is the sender.

2) Talking to driver(s) moving in front: Driver of E may want to talk to the drivers moving in front of her (i.e. A, B or C). This could be to notify the driver(s) moving in front about the situation behind, e.g. vehicle A is blocking the traffic. Then, *DriverTalk* extracts the image of the target vehicle in its view, and uses it as VID along with the information the driver wants to convey. Upon receiving the message, *DriverTalk* in the target vehicle compares its self-images with the received VID to check whether someone is talking to it.

3) Talking to all drivers around: A driver may want to talk to all surrounding drivers to inquire about some information or alert them about a hazardous situation. In this case, *DriverTalk* broadcasts the self-images or the vehicle images it currently sees along with the information the driver wants to convey. Surrounding vehicles which receive this message compare the received VIDs with their self-images or vehicle images in their own view to infer the sender. If the message is a question and the receiver happens to know the answer, the receiver could send a message back to answer. In the answer message, the questioner's VID (which is received in the question message) is used to tell who the target of the answer is.

Before proceeding to describe the design of *DriverTalk*, we allay some potential concerns with it. i) *Why not simply use a vehicle's license number as its address?* While that is an option, a vehicle's license number is only legible for most smartphones' cameras within a very limited distance and angle. ii) *Why not a GPS based system?* When a driver intends to talk to a particular counterpart, say the driver moving in front in the same lane, others moving in front in the left/right lane should not be confused. Due to the large location error in GPS relative to lane width, it is not reliable to distinguish the lane positions. Sometimes, GPS could even confuse fore-and-aft relationship between vehicles, when they are moving closely. iii) *Drivers are now burdened with two systems, one for talking to* DriverTalk*-enabled drivers and the second for signalling others.* We concede that this is a limitation of our current design of *DriverTalk*, which still takes an important step towards exploration of alternative forms of inter-driver communication. Moreover, the recent developments (Android Auto [2] and Apple CarPlay [3]) in letting a smartphone power a car's dashboard, hold promise towards an integrated interface for communication between drivers. iv) *Wouldn't the system be distracting to drivers?* With *DriverTalk*, drivers send and receive audio messages, they don't need to operate on their smartphones while driving. Drivers still focus on their maneuvers and keep their eyes on the road. Overall, by leveraging smartphones, *DriverTalk* could help most drivers, from high-end vehicle owners to legacy and/or economy vehicle owners, in improving safety and reducing stress while driving.

Our contributions with this work can be summarized as follows: *i)* We propose *DriverTalk* as the first system allowing drivers with no prior acquaintance to talk to each other; *ii)* We design a mechanism which uses VID, instead of conventional network address, to identify sender and receiver during network communication as well as the relative position of each other. *iii)* We study the feasibility of *DriverTalk* by evaluating its reliability and efficiency in both highway and city scenarios.

II. Design and Implementation

We now present the design and implementation details of *DriverTalk*. The overall flow in *DriverTalk* is shown in Fig. 2.

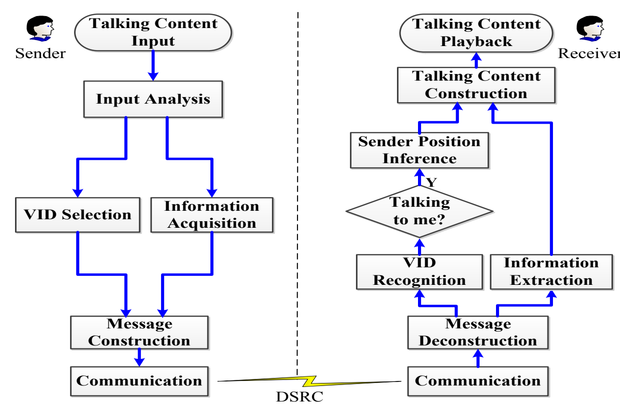

Fig. 2. *DriverTalk* system flow. At the sender side, the system analyzes a driver's voice input and constructs message by combining the information input by the driver and the image selected as VID based on the driver's intent. At the receiver side, the VID is extracted and checked against the self-images of receiver or the vehicle images seen by the receiver to decide whether the message is for this driver. If this driver is the target receiver, the message and the inferred sender's position is combined and played back to the driver.

A. Vehicle Detection

To talk to a vehicle, the first thing *DriverTalk* needs to do is detecting the vehicles in its view. We trained a Haar Cascade classifier [4] [5] on vehicles' rear-end, left-back, and right-back sides images. Sample vehicle detection results in *DriverTalk* are shown as green rectangles in Fig. 3 (left). The subimages within the rectangles are extracted as VIDs for these vehicles.

Besides detecting vehicles, *DriverTalk* also continuously extracts the lane markings on the road (as shown as blue lines in Fig. 3 (left)) to learn the relative lane positions between the detected vehicles and the ego-vehicle[2].

In *DriverTalk*, we define eight types of relative positions between a vehicle and its surrounding vehicles, as depicted in Fig. 3 (right). Note that, these 8 positions just indicate the relative directions from the ego-vehicle, while their distances

[2]In a context which involves several vehicles, the one which is chosen as the subject in narration is called ego-vehicle.

TABLE I. MAPPING OF KEYWORDS–PREDEFINED MESSAGE–TARGET VEHICLE

Index	Keyword	Message	Target Vehicle(s) of Talking (Position)
1	Change Left	I want to change to the left lane.	Vehicles on the left lane moving parallel and behind (④ ⑥).
2	Change Right	I want to change to the right lane.	Vehicles on the right lane moving parallel and behind (⑤ ⑧).
3	Tailgate	Don't tailgate.	Vehicle moving behind on the same lane (⑦).
4	Block	You are blocking the traffic.	Vehicle moving in front of the ego-vehicle on the same lane (②).
5	Lane Closed	This lane is closed ahead.	Vehicle moving behind on the same lane (⑦).
6	Accident	There is accident ahead.	All vehicles moving behind on the same or adjacent lanes (⑥ ⑦ ⑧).
7	Problem	There is something wrong with my car.	All vehicles moving around.
8	Slow Down	I will slow down.	Vehicle moving behind on the same lane (⑦).
9	Move	Please move.	Vehicle moving in front of the ego-vehicle on the same lane (②).
10	Light On	Your light is on.	Vehicle moving in front of the ego-vehicle on the same lane (②).
11	Question	Question:	All vehicles moving around.
12	Answer	Answer:	The sender of the corresponding question message.

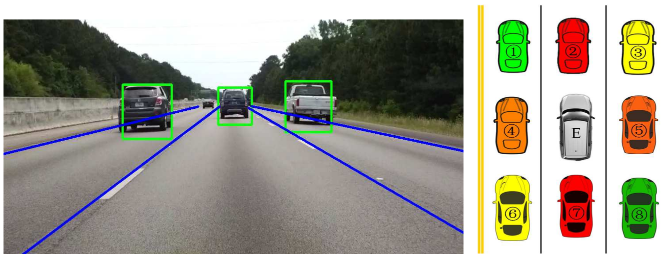

Fig. 3. (Left) *DriverTalk* detects vehicles moving in front and extracts the subimages (in the rectangles) as VIDs for the detected vehicles. It also extracts lane markings to identify the relative lane positions between ego-vehicle and the detected vehicles. (Right) The relative positions of the vehicles moving around the ego-vehicle.

to the ego-vehicle can vary. The only requirement is that they are all at one visual-hop from the ego-vehicle.

B. Input Analysis and Message Construction

1) Input Analysis: To provide hands-free operation while driving, *DriverTalk* accepts voice input from the driver. To discriminate the talking over *DriverTalk* from regular talking that happens between a driver and a passenger inside the vehicle, *DriverTalk* expects a foreword, "OK Driver", akin to "OK Glass" for Google Glass. When a driver wants to talk to other drivers, she says "OK Driver" first, and then expresses what she wants to convey to surrounding drivers. Therefore, the system must recognize the foreword, and hence understand what the driver is talking and whom the driver is talking to. In *DriverTalk*, we use Android SpeechRecognizer. Fig. 4 shows the process of analyzing driver's input in *DriverTalk*.

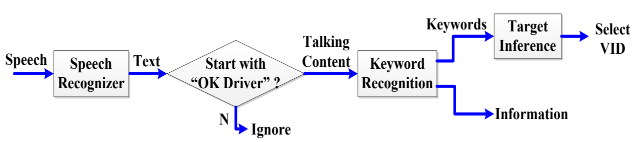

Fig. 4. Process of voice input analysis in *DriverTalk*. This phase extracts the keywords from the speech and infers the target of talking.

DriverTalk transmits what a driver says to another driver over wireless communication. To reduce communication overhead, it converts speech into text and transmits text.

To ease the driver's burden when talking to the drivers around her and make the speech recognition more efficient, we define a set of keywords, which map to a set of pre-defined common messages (a sample set of pre-defined message 1∼10 are listed in Table I). Thus, drivers only need to speak the keywords instead of long sentences (3rd column in the table).

After recognizing the keywords, for the pre-defined messages, *DriverTalk* fetches the corresponding index and only transmits the index. At the receiver, the corresponding pre-defined message is retrieved and played back. The keywords "Question" and "Answer" give more flexibility, which could be used to ask and answer questions. Once *DriverTalk* detects one of these keywords, all that the driver says after the keyword is treated as the content of question or answer. The target of a question message is all the vehicles surrounding the questioner, because it does not know who has the answer, whereas the answer message is only targeted at the corresponding questioner.

2) Message Construction: A message exchanged between *DriverTalk*-enabled vehicles contains two major parts: VID and message content. VID is a vehicle image which is used to identify the sender and receiver of a message. Message content is the information which the sending driver wants the receiving driver to be aware of. Below, we describe how the system forms a message by combining the two parts[3].

i) VID Selection: Following the relationship defined in Table I, *DriverTalk* could infer the position of a target vehicle after recognizing keywords. Then, it automatically selects proper VID from self-images and detected vehicle images in order to talk to the target vehicle. Table II defines what images will be selected as VID based on the target vehicle's position.

TABLE II. RULES FOR VID SELECTION

Target Vehicle	Selected VID
Position ①	Detected vehicle image of vehicle at ①
Position ②	Detected vehicle image of vehicle at ②
Position ③	Detected vehicle image of vehicle at ③
Position ④	Detected vehicle image of vehicle at ① (if exists); otherwise detected vehicle image of vehicle at ②
Position ⑤	Detected vehicle image of vehicle at ③ (if exists); otherwise detected vehicle image of vehicle at ②
Position ⑥	Left-back self-image
Position ⑦	Rear-end self-image
Position ⑧	Right-back self-image
Position ⑥ ⑦ ⑧	Self-images at all directions
All vehicles moving around	Vehicle Problem Warning: All detected vehicle images and rear-end self-image Question: One self-image
Questioner	Image received in the related question message

Here, a special case happens when the target vehicle is at position ④ or ⑤. Smartphone camera has a limited field of view. When two vehicles are moving in parallel, they cannot

[3]Detailed specification of *DriverTalk* message is omitted here due to space limitation.

detect each other, hence they are not able to get the VID of each other. If one of them wants to change lane, even if the driver gives light indicator, the other driver may not see it, so collision might happen. To avoid this hazard, *DriverTalk* utilizes a 3rd-party VID, which is the image of the vehicle(s) moving in front of ego and target vehicles, which could be detected by both of them. Once a vehicle, say A, finds out that the one planning lane change, say B, is detecting the same vehicle, say C, as in its view, vehicle A infers that the message, with VID of C, is from the driver of B which is moving parallel to A, hence proper action could be taken to avoid collision.

When *DriverTalk* recognizes (keyword 1 or 2 in Table I) that the driver wants to change lane, it sends a message with the related left-back or right-back self-image as well as the 3rd-party VID. Thus, *DriverTalk* helps the driver to actively convey the lane changing intent to all the possibly affected vehicles.

DriverTalk allows a driver to pose questions to other drivers around. The questioner only needs to identify itself by using its self-images as VID in the question message. While answering a question, in the answer message, the questioner's VID is specified as the receiver. When the questioner receives the answer message and matches the VID with its self-image, it extracts the answer. In Q & A, from the driver's point of view, the positions of the questioner and answerer are not critical.

ii) Message Content: Message content is used to convey the information about intent, notification, alert, question or answer. For pre-defined messages, only the corresponding index is transmitted in the message; for "Question" or "Answer" messages, the converted text content of the question and answer is transmitted. At the receiver side, the message content in text will be converted back to voice and played back.

C. Vehicular Communication

When two vehicles meet, they don't know each other's conventional address (e.g. MAC, IP). In *DriverTalk*, the VID embedded in the message becomes the address. When transmitting a VID-embedded message, the sender wants only the target driver to get her message, so the VID becomes the address which tells who is the sender or who should be the receiver.

In *DriverTalk*, messages are broadcast over DSRC 802.11p [6]. Current smartphones do not have standard DSRC module installed yet. But after-market DSRC component for smartphone is available now. Besides, some IC manufacturers, such as QualComm [7], have already demonstrated DSRC-enabled reference design phones [8]. We can expect that DSRC will be a standard component in smartphones in the very near future.

D. VID Recognition and Message Playback

1) VID Recognition: When *DriverTalk* receives a message, it needs to find out whether this message is targeted at it, i.e. someone is talking to it. As mentioned earlier, we use VID to identify sender and receiver of a message. Once *DriverTalk* extracts the VID from the received message, it compares with its self-images and/or the detected vehicle images. Image comparison is realized by image matching, as illustrated in Fig. 5. If the number of matched points exceeds some predefined threshold, *DriverTalk* concludes that the message is for it and proceeds to extract the message content.

DriverTalk should perform efficiently to provide good user experience, so image matching should be carried out as quickly as possible on smartphone. *DriverTalk* uses ORB [9] for image matching, as it is found to be efficient on smartphone. According to the measurement in [10], we choose 10∼14 KB to be the image size of VID to be used in messages, which might allow us to make confident conclusion on whether two VIDs are for the same vehicle or not. Meanwhile, it might also be suitable for communication (studied in Section III). If the size of self-images or detected vehicle images exceeds this chosen size range, *DriverTalk* scales down the images to make them close to this range before transmission.

When *DriverTalk* receives a message, it compares the received VID with its self-images and its detected vehicle images to identify the relative position (left, front, etc.) of the sender and tell its driver where the peer is. Table III shows the relationship between the position of the sender and image matching result.

TABLE III. RULES FOR INFERRING SENDER'S POSITION

Matched Image Pair	Sender's Position
Received VID matched with detected VID at position ①	①
Received VID matched with detected VID at position ②	②
Received VID matched with detected VID at position ③	③
Received VID matched with detected VID at position ① or ②; VID Type = Detected 3rd-party VID	④
Received VID matched with detected VID at position ③ or ②; VID Type = Detected 3rd-party VID	⑤
Received VID matched with the left-back self-image	⑥
Received VID matched with the rear-end self-image	⑦
Received VID matched with the right-back self-image	⑧

Besides, in the message, the target direction and relative lane position information is included in the message, which could help to eliminate some unnecessary matching and makes the system more efficient. For example, if a message is targeted at a vehicle on the right lane to the sender, the vehicle that is moving on the left-most lane could immediately ignore it, because it could not be the target receiver.

2) Message Playback: If a received message is a pre-defined message, *DriverTalk* fetches the complete content of the predefined message corresponding to the index received in the message; otherwise it extracts the message content for question or answer from the message itself. Besides, to make the receiving driver understand the message better, the inferred sender's position is also attached to augment the message. The final talking content for playback is in the form like:

Sender's Position: Message Content

For instance, "The driver in front says: Don't tailgate."

DriverTalk uses Android text-to-speech engine to convert text content back into voice and plays it back to complete the whole talking between two drivers.

E. Avoiding Abuse

DriverTalk provides drivers opportunities to talk to each other during driving. It benefits drivers, but it is also possible that some unruly drivers abuse this system. In *DriverTalk*, several mechanisms are adopted to prevent abuse.

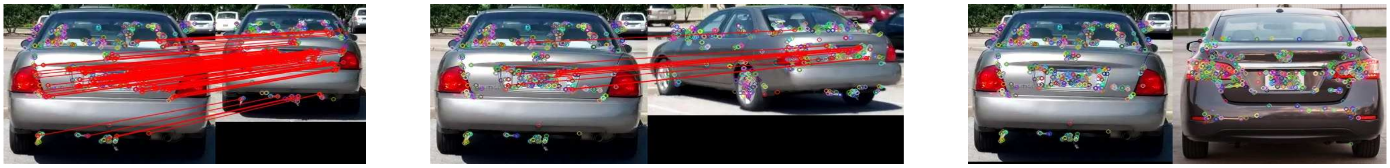

Fig. 5. (Left) Image matching between a self-image and a detected vehicle image in the same direction for the same vehicle. (Middle) Image matching between a self-image and a detected vehicle image in different directions for the same vehicle. (Right) Image matching between vehicle images of different vehicles. The two same-direction images for the same vehicle produce more matched points than different-direction images.

1) Regulate the way driver talks: With the definition of a set of keywords and pre-defined messages, *DriverTalk* regulates how and what the drivers could talk.

2) Set different levels for message: We define 3 levels for the importance and criticality of the messages. The first level is for warning. Every driver should be aware of it because this level of message is related to immediate safety. The second level is for reminding. This message level is not directly related to any immediate driving safety, but it is used to reduce the pressure from others or make the driving more comfortable. The last level is for question and answer.

In *DriverTalk*, the first level of messages are always enabled. Drivers are allowed to decide whether to send/receive messages at the other two levels or not through configuration.

3) Regulate message frequency: DriverTalk defines maximum number of messages each *DriverTalk*-enabled vehicle could send within a certain period. In addition, the system also allows drivers to configure the maximal message frequencies they are happy to receive from other drivers. Besides, between a particular sender and a particular receiver, *DriverTalk* also regulates the message frequency. For example, when *DriverTalk* detects that messages are being received too frequently from a particular counterpart, it automatically filters to reduce the messages being played back to its driver.

III. Evaluation

To make *DriverTalk* a usable system, the following aspects should be studied. First, when a driver talks to another driver, the target driver should correctly receive it and others should not be confused. Second, the messages must be exchanged reliably over wireless under various traffic conditions. Last, to make the talking experience like in real life, the system should work very efficiently. Below, we study the performance of *DriverTalk* to show that it could meet all these requirements.

A. VID Recognition

As a talking system, *DriverTalk* should be able to correctly recognize who is talking to whom. In this system, we use VID to identify the sender and receiver, so the system should be able to confidently distinguish whether a received VID matches one of its self-images and currently detected vehicle images. The problem could be divided into two aspects: On one side, if two VIDs are for the same vehicle, whether the system could accurately conclude that they are the same. On the other side, if two VIDs are for two different vehicles, whether *DriverTalk* could reliably tell that they are different.

To show the performance of *DriverTalk* in recognizing same-vehicle VIDs and distinguishing different-vehicle VIDs, we organize two sets of vehicle images. The first set contains 913 pairs of vehicle images, each pair contains two images for the same vehicle. The second set has 28245 pairs, the two images in each pair are for different vehicles.

We carry out image matching for each pair. Fig. 6 shows the accuracy of *DriverTalk* in recognizing same-vehicle images and distinguishing different-vehicle images with certain thresholds for the number of matched points [10]. It is evident that *DriverTalk* could recognize two VIDs are for the same vehicle with a high probability (e.g. 99% for threshold = 15). Further, it could distinguish two different vehicles very accurately (close to 100% for threshold $\geq$ 10).

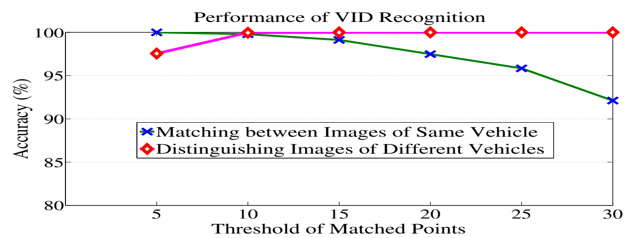

Fig. 6. Performance of *DriverTalk* in recognizing VIDs for same vehicle and distinguishing VIDs for different vehicles.

Along with the above study, one natural question arises: What if there are two same-appearance vehicles showing up within a third vehicle's communication range? If the third vehicle talks to one of them, the other one might be confused. We study to see how often this kind of case could happen in reality.

We took videos for the traffic of a multi-lane highway both in the rush hour and off-peak hour. From these videos, we randomly extract 1000 "traffic snapshots". Each snapshot starts when a randomly selected vehicle (i.e. the first vehicle in the snapshot) passes a pre-defined baseline (as shown in Fig. 7). We count all the vehicles that pass the baseline after the first vehicle until the first vehicle has traveled about 80 meters away. We extract the images for all the vehicles in the snapshot at the moment they pass the baseline. These vehicles form an 80-meter-spanning snapshot of the traffic.

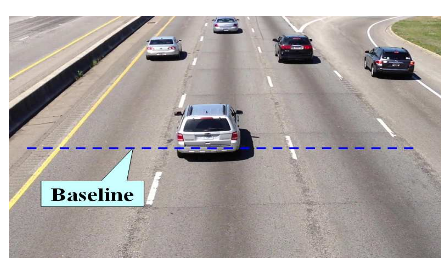

Fig. 7. Highway traffic. When the vehicles pass the pre-defined baseline, their images are extracted to form traffic snapshot.

To learn about the diversity of vehicles in the snapshots, we checked both manually and through image matching. From manual checking, we found that only 3 of the 1000 snapshots contain two same-appearance vehicles. So we believe that the probability of two same-appearance vehicles occur in the same communication range is low enough for our system to work in reality. During image-matching checking, we treat the result from manual checking as ground truth and compare all possible image pairs in each snapshot. Fig. 8 shows the accuracy of image matching for these 1000 snapshots. By choosing proper threshold, the system could identify VIDs correctly.

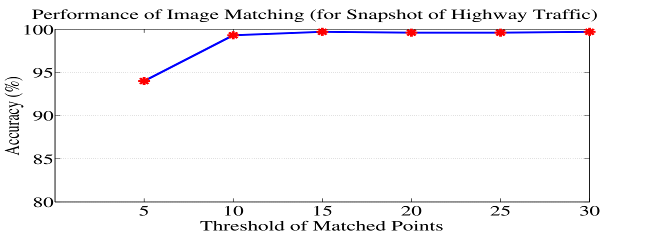

Fig. 8. Performance of *DriverTalk* in recognizing VIDs for snapshots of highway traffic.

One thing we need to point out is, when we took the videos for the diversity study, we were on an overpass bridge over highway. The angle at which the camera sees these vehicles is slightly different from the actual situation the *DriverTalk* system will see. When the system is used in a vehicle, it sees exactly the left-back, rear-end or right-back sides of other vehicles. While in our videos, we also see the roof of the vehicles due to the angle from the bridge. This might slightly affect the result. But considering the appearance of vehicles, for two vehicles, their similarities and differences are usually consistent from their roofs to their left/rear-end/right sides.

Even in the case there are two exactly same-appearance vehicles in one communication range, the relative lane position and direction information provided in the message could help to identify which one the sender is talking to.

With the above study, we believe that *DriverTalk* could reliably recognize the receiver or sender of a message.

B. Simulation of Vehicular Communication

In *DriverTalk* system, vehicles need to exchange VID-embedded talking message. Carrying out a large scale deployment requires a great amount of resources, which are unobtainable for us. Therefore, we use simulation to study the large-scale communication among *DriverTalk*-enabled vehicles.

We use SUMO [11] to generate vehicular mobility model and import it into NS2 [12] to drive the network communication.

We simulated both highway and city environments. For the highway traffic, we also studied with two different traffic modes. One is dense mode, in which the traffic flow is relatively heavy and each vehicle has many neighboring vehicles, thereby the communication between vehicles will be more intensive. The second is sparse mode, in which every vehicle has fewer neighbors than dense mode and the message transmission in the network will be less. Although the speed of vehicles on highway is high, the relative speed between vehicles is no more than 20 mph in most cases. Once two vehicles meet, they will at least stay within the communication range of each other for a while.

For the city scenario, the vehicles are moving in a 1km x 1km Manhattan-like area (as shown in Fig. 9 (left)). Traffic lights are deployed at intersections regulating the traffic flow, which lead to the pile-up of vehicles (see Fig. 9 (right)).

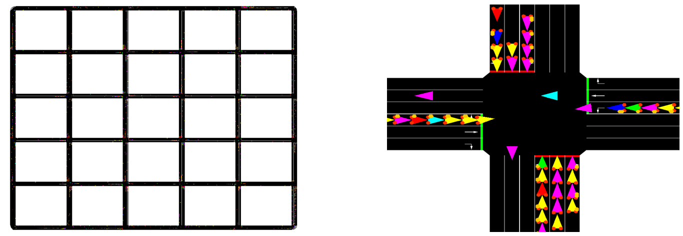

Fig. 9. (Left) An 1km x 1km city layout in the simulation (each segment is 200 meters). (Right) A close-up view of an intersection with vehicles (the triangle-like icons stand for vehicles) piling up due to traffic lights.

The detailed setting in our simulation is listed in Table IV. In our simulation, to stress test the system, we assume every driver will talk once every 60±30 seconds.

TABLE IV. SETTING OF *DriverTalk* SIMULATION

Parameter	Value
Simulation Period	Highway: 1800 s; City: 400 s
Number of Vehicles	Highway: 400 (8 types of vehicles with different length/speed/acceleration/deceleration.)
	City: 1200 (10 types of vehicles used.)
Speed	Highway: 22∼36 m/s (50∼80 mph)
	City: ∼16 m/s (35 mph)
Traffic Density	Highway: Sparse (57 vehicles/km); Dense (110 vehicles/km)
	City: 1200 vehicles move around randomly.
Wireless Protocol	DSRC 802.11p
Antenna Type	OmniAntenna
Propagation Model	Two Ray Ground
Data Rate	6 Mbps (QPSK), which is the optimal rate [13].
Message Size	Each VID is 10/12/14 KB, which is sliced into 1-KB small packets; size of other information in message is too small and ignored.
Transmission Method	Broadcast
Message Frequency	1∼4 messages/vehicle, every 60±30 s
Transmission Range	Highway: 60 meters; City: 40 meters

We measure the message reception rate to see how reliably the system could deliver messages to targeted receivers. The overall message reception rates are shown in Fig. 10. On highway, *DriverTalk* can achieve a reception rate of about 97% both in dense mode and sparse mode. In city, due to the pile-up of vehicles at intersections, the reception rate decreases slightly to 95%. From this evaluation, we believe *DriverTalk* could reliably deliver what a driver says to the target driver(s).

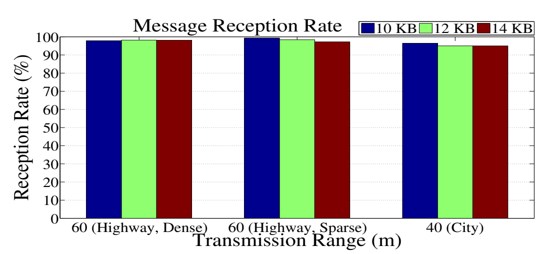

Fig. 10. Message reception rate in dense/sparse highway traffic and city.

C. Efficiency

To provide good user experience, *DriverTalk* should perform efficiently to allow drivers to talk smoothly. Here, we study the end-to-end latency of the system from from the time a sender says something to the time the receiver starts to hear it.

In the *DriverTalk* system, time is mainly consumed in the following steps: vehicle detection, vehicular communication, VID recognition, speech and text conversion. In this section, we measure the time spent in each step to get an estimate about the end-to-end delay of talking.

1) Efficiency of Vehicle Detection: For vehicle detection, we measured the time on a real road. When a vehicle occurs in the video frame, *DriverTalk* could detect the vehicle within 97 $\pm$ 17 ms (measured on a Galaxy S4).

2) Time Taken for VID Recognition: To measure the time taken for VID recognition, we prepared 406 vehicle images with the sizes range from 9 KB to 20 KB. Among these images, some of them are for same vehicles, others are for different vehicles. We measured the time of image matching with all possible image pairs on a Galaxy S4, the one-time image matching could complete within 108$\pm$44 ms.

To estimate the total effort on image matching, i.e. how many times a *DriverTalk*-enabled vehicle needs to do image matching per second, we check how many messages each one will receive, which corresponds to the number of received VIDs that need to be checked. Fig. 11 illustrates the message reception frequency. In most settings, on average, every vehicle will receive no more than one message per second.

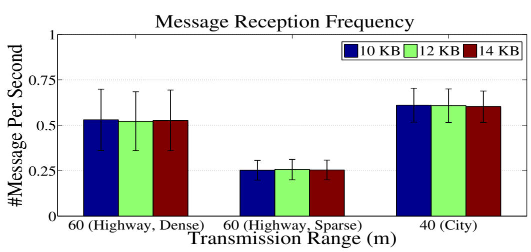

Fig. 11. The number of messages each *DriverTalk*-enabled vehicle receives every second in dense/sparse highway traffic and city.

Upon receiving a message, the system needs to identify whether itself is the target receiver by comparing the VID in the message with its self-images and the currently detected vehicle images. As mentioned before, with the knowledge of the relative lane position and direction information from the message, in many cases *DriverTalk* does not need to compare all its self-images and detected vehicle images with the received VID, reducing time spent on image matching. Besides, smartphone's CPU is becoming more and more powerful; some even have 8 cores now. The VID recognition task could be carried out in parallel and the time will be consumed even less. We believe the overhead of VID recognition is fairly acceptable, which allows the device to carry out other tasks.

After recognizing the target receiver as self, *DriverTalk* should allow time for the message to be played back. Here, we examine how often every participant is chosen as the target receiver by other drivers. Fig. 12 shows the frequency of each vehicle being the target receiver. As per our stress-testing scenario, about every 16$\sim$38 seconds, one driver could become the talking target of some other drivers. Considering the normal speed of talking, people could easily speak 20$\sim$30 words within 10 seconds, providing sufficient time for a message to be played back before another message could arrive.

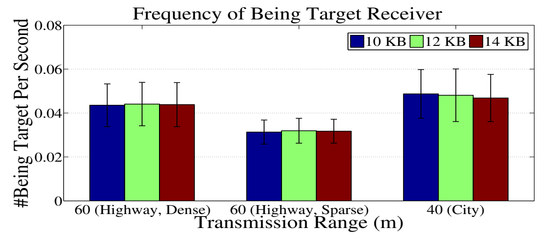

Fig. 12. The frequency of each *DriverTalk*-enabled vehicle is chosen as target receiver by others in dense/sparse highway traffic and city.

3) End-to-End Latency of Vehicular Communication: From the simulation, we also measured the end-to-end latency of message in vehicular communication. Fig. 13 shows the latency in both highway and city scenarios. In all cases, messages could be delivered within 55 ms on average.

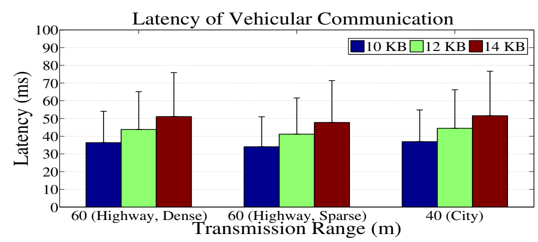

Fig. 13. Latency of communication in dense/sparse highway traffic and city.

4) Speech & Text Conversion: We measured the efficiency of speech-to-text and text-to-speech conversion. For speech-to-text, we measured the time from the end of speech to the completion of conversion into text. For text-to-speech, we measured the time from the end of the acquisition of text to the completion of the conversion into voice. We drove around in downtown as well as on highway with a LTE-enabled Samsung Galaxy S4 to measure the time of conversion with Android speech-to-text service and text-to-speech engine. The sentences we used vary from one word to thirty words. Table V lists the time on speech & text conversion.

TABLE V. TIME OF SPEECH & TEXT CONVERSION

Speech $\rightarrow$ Text	Avg. 251 ms (95th percentile: 554 ms)
Text $\rightarrow$ Speech	Avg. 12 ms (95th percentile: 22 ms)

By summarizing all the above measurements, we can expect that in *DriverTalk*, when a driver says something, the target driver could start hearing it within about 1 second, which could guarantee smooth user experience for this talking system.

IV. RELATED WORK

In both academia and industry, vehicle related systems are being developed to improve driving experience and safety.

Ford is known to be working on a "Talking Cars" project [14], which allows cars to talk to each other and expect this could help cars avoid crashes and reduce fuel consumption. Yet the details about how it works and when the system will be available is not clear.

CarSpeak [15] is a V2V collaborative system. Each CarSpeak-enabled vehicle senses the environment along the road. Every participant could query information captured by others for interested regions. In this way, the driver could know about the obstacle, pedestrians in the region which is out of her line of sight, which helps to improve safety.

RoadSpeak [16] and Social Vehicle Navigation [17] are two client-server based vehicular social network systems which allow drivers to do voice chatting during their commuting. Drivers join certain voice chatting groups based on interests, timely location [16] or routes [17]. The participants could talk on common topics and report road situations. Although these systems allow drivers to talk to each other, but all the messages are aggregated by central servers. A driver could not identify any particular neighbors to have in-situ talking.

Authors in [18] proposed a GPS, CAN, radar and camera based cooperative safety system, which tries to infer drivers' intents. It expects to allow the participants exchange intents via V2V and V2I communication. This work only analyzes the potential benefit in reducing collision, how accurately the system could infer driver's intent is unknown. Due to the GPS error and camera view range, it is hard for a driver in the proposed system to convey intent to another particular driver.

A work correlated to *DriverTalk* is OmniView [10], which also uses a form of VID. However, the purpose of OmniView is to provide drivers with real-time maps about surrounding traffic, which is quite different from facilitating talk between drivers.

Different from all existing systems, *DriverTalk* aims to improve driving experience and safety by allowing drivers to talk to each other directly to convey information. To the best of our knowledge, *DriverTalk* is the first system that enables in-situ talking between drivers with no prior acquaintance.

V. Limitations and Future Work

DriverTalk uses appearance images of vehicles to represent both sides, source and target, of talking. It requires the appearance of vehicles in the communication range to be diverse, for it to perform well. While we observed only less than 1% highway scenarios contain two same-appearance vehicles within the communication range in our preliminary study, we still need to throughly evaluate the performance of *DriverTalk* in cities where taxis and auto rickshaws are prevalent. We also need to explore whether and how the proposed approach will work in scenarios involving two-wheeler vehicles.

DriverTalk relies on computer vision algorithms to do image matching for VID recognition. Like many other computer vision technologies, the light condition will affect the performance of image matching, because the detailed characteristics on the detected vehicle images will become hard to discriminate when the light condition is poor. So the system will not work at night or when it is dark in bad weather. Considering the extent of driving that happens during the day, we argue that our system is still valuable to drivers.

As a next step, we plan to study the characteristics of feature points in vehicle images. If we could find some small-size feature points which could uniquely represent a vehicle, *DriverTalk* may send the feature points instead of the whole vehicle image, the system will perform more efficiently.

We have thus far presented DriverTalk as a pure ad-hoc networking system. While it does not have to rely on any centralized infrastructure for talking function, having a centralized controller in the cloud can help its operation. For instance, we can tune and adapt the system better by having *DriverTalk* periodically report meta data on received and sent messages. Furthermore, based on such message statistics, it is easier for a centralized controller to prevent abuse. We will explore the addition of cloud component to *DriverTalk* in our future work.

We also plan to conduct real world experiments with some after-market DSRC components and arrange vehicles to do real talking when moving both on highway and in city.

VI. Conclusion

In this paper, we explored the feasibility of enabling targeted communication between drivers, as a better alternative to the existing modes of coarse-grain signalling through horns and lights. Towards that end, we proposed a smartphone-based system, called *DriverTalk*, that allows neighboring drivers on the road, with no prior acquaintance, to talk to each other. We have presented the details of *DriverTalk* system, specifically, how it identifies the senders and receivers of messages in different communication scenarios, utilizing appearance images of vehicles as their Visual IDs. We have evaluated the system by simulating the traffic in both highway and city scenarios and shown that it could perform efficiently and reliably. Next, we plan to conduct real world experiments with some after-market DSRC components and several vehicles/drivers.

References

[1] "70% noise pollution due to honking," http://timesofindia.indiatimes.com/city/gurgaon/70-noise-pollution-due-to-honking/articleshow/21473250.cms.

[2] "Android auto," https://www.android.com/intl/en_us/auto/.

[3] "Carplay," https://www.apple.com/ios/carplay/.

[4] P. Viola *et al.*, "Rapid object detection using a boosted cascade of simple features," in *IEEE CVPR 2001*.

[5] R. Lienhart *et al.*, "An extended set of haar-like features for rapid object detection," in *ICIP 2002*.

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

[7] "Cooperative ITS for all: Enabling DSRC in mobile devices," http://tinyurl.com/omvfgfj.

[8] "How snapdragon and honda are working to save lives with smartphones," https://www.qualcomm.com/news/snapdragon/2015/06/16/how-snapdragon-and-honda-are-working-save-lives-smartphones.

[9] E. Rublee *et al.*, "Orb: An efficient alternative to sift or surf," in *IEEE ICCV 2011*.

[10] R. Meng *et al.*, "OmniView: A mobile collaborative system for assisting drivers with a map of surrounding traffic," in *ICNC*, 2015.

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

[12] "The network simulator - ns-2," http://www.isi.edu/nsnam/ns/.

[13] D. Jiang *et al.*, "Optimal data rate selection for vehicle safety communications," in *VANET*, 2008.

[14] "Ford's 'talking cars' could reduce crashes, fuel use," http://gigaom.com/2011/06/01/fords-talking-cars-could-reduce-crashes-fuel-use/.

[15] S. Kumar *et al.*, "Carspeak: A content-centric network for autonomous driving," in *SIGCOMM*, 2012.

[16] S. Smaldone *et al.*, "Roadspeak: Enabling voice chat on roadways using vehicular social networks," in *SocialNets*, 2008.

[17] W. Sha *et al.*, "Social vehicle navigation: Integrating shared driving experience into vehicle navigation," in *HotMobile*, 2013.

[18] A. Doshi *et al.*, "Communicating driver intents: A layered architecture for cooperative active safety applications," in *ITSC*, 2010.