Towards Robust Vehicular Context Sensing

Hang Qiu, Jinzhu Chen, Shubham Jain, Yurong Jiang, Matt McCartney, Gorkem Kar, Fan Bai, Donald Grimm, Marco Gruteser, Ramesh Govindan

***Abstract*—In-vehicle context sensing can detect many aspects of driver behavior and the environment, such as drivers changing lanes, potholes, road grade and stop signs, and these features can be used to improve driver safety and comfort, and engine efficiency. In general, detecting these features can use either on-board sensors on the vehicle (*car sensors*), or sensors built into mobile devices (*phone sensors*) carried by one or more occupants, or both. Furthermore, traces of sensor readings from different cars, when *crowd-sourced*, can provide increased spatial coverage as well as disambiguation.**

In this paper, we explore, by designing novel detection algorithms for the four different features discussed above, three related questions: How is the accuracy of detection related to the choice of phone vs. car sensors? To what extent, and in what ways, does crowd-sourcing contribute to detection accuracy? How is accuracy affected by phone position? We have collected hundreds of miles of vehicle traces with annotated groundtruth, and demonstrated through evaluation that our detection algorithms can achieve high accuracy for each task (e.g. $>$ 90% for lane change determinations) and that crowd-sensing plays an indispensable role in improving the detection performance (e.g. improving recall by 35% for lane change determinations on curves). Our results can give car manufacturers insight into how to augment their internal sensing capabilities with phone sensors, or give mobile app developers insight into what car sensors to use in order to complement mobile device sensing capabilities.

I. INTRODUCTION

Industry is moving towards making automobiles programmable and customizable through apps. Automakers have created app developer portals, versions of mobile operating systems such as iOS and Android exist for cars, and cars increasingly provide rich network connectivity options (LTE cellular Internet connectivity, Bluetooth and WiFi).

The problem space: Vehicular context sensing. This convergence between mobile computing and automobiles motivates the problem space we consider: vehicular context sensing. We use the term *vehicular context* to include both the environment surrounding a vehicle at any point in time, and also whatever actions or operations the vehicle is performing at any point in time. Examples of vehicular context include traffic regulators (stop signs, traffic lights, speed limit signs), road surface anomalies (potholes, bumps), road topography (grade, banking), as well as vehicular actions (decelerations, lane departures, speeding).

Vehicular context information can be used in several ways. Maps augmented with traffic regulators can be used by navigation devices and apps to warn inattentive drivers. Crowd-sourced road-anomaly detection can help transportation agencies identify and prioritize road surface maintenance. Road topography information can enhance the efficiency of vehicular transmission subsystems, since, for example, a road-grade or banking-aware transmission system can efficiently deliver power. Finally, a record of vehicular actions can be used by insurance companies to offer good driver discounts.

The design space of vehicular context sensing. There are two general approaches to detecting (or *sensing*) vehicular context.[1] One approach is to use the smartphone[2] [27, 10]. The high degree of penetration of mobile devices ensures that almost every vehicle is likely (at least in developed countries) to have an occupant (driver or passenger) with a smartphone. These devices come both with positioning hardware and software (GPS, WiFi based positioning, *etc.*) and many sensors (accelerometer, magnetometer, barometer, and so on).

A second, less well-known, approach is to use the sensors embedded in a car [16, 21]. Some modern cars have several hundred physical and virtual (i.e., derived from physical) sensors onboard, which describe, in near-real time, the operation of several of the internal subsystems of the car. Examples of sensor readings available over the CAN bus include: vehicle speed, throttle position, transmission lever position, automatic gear, cruise control status, radiator fan speed, fuel capacity, and transmission oil temperature. These sensor readings are used to control subsystems of the vehicle, but can also be exported to an external device using the standard On-Board Diagnostics (OBD-II) port available on all vehicles. Due to business, privacy, and security considerations, many of these sensors were not previously exported to external devices but recently, Ford and General Motors have made about 20 sensor types available

This work is supported by the National Science Foundation (NSF Grant No. 1330118).

H. Qiu and R. Govindan are with the Viterbi School of Engineering, University of Southern California, Los Angeles, CA, 90089, USA (e-mail: {hangqiu, ramesh}@usc.edu). J. Chen, F. Bai and D. Grimm are with the General Motors Research Center, Warren, MI, 48090, USA (e-mail:{jinzhu.chen, fan.bai, donald.grimm}@gm.com). S. Jain is with the Department of Computer Science, Old Dominion University, Norfolk, VA 23529, USA (e-mail: jain@cs.odu.edu). Y. Jiang is with the Hewlett-Packard Labs, Palo Alto, CA, 94304, USA (e-mail: yurong.jiang@hpe.com). M. McCartney is with the Applied Minds at Burbank, CA 91504, USA (e-mail: mmccartney@appliedminds.com). G. Kar and M. Gruteser are with the School of Engineering, Rutgers University, North Brunswick, NJ, 08902, USA (e-mail: {gkar87, gruteser}@winlab.rutgers.edu).

[1] A third approach that has been investigated for some forms of vehicular context, such as stop signs, is to use computer vision techniques. Despite limited success (some vehicles now ship with vision based lane departure systems), the efficacy of these approaches can be low under poor lighting or adverse weather conditions. We leave it to future work to explore this approach in greater detail.

[2] Vehicular context sensing requires continuous sensor acquisition. While prior work [22, 29] has pointed out that continuous sensing in mobile devices can impede battery life, the car is one environment continuous sensing is feasible because of the availability of power.

through their OpenXC platform and GM Developer Network, respectively.

It is tempting to believe that car sensors will always be superior and that in the long run vehicles will incorporate all useful sensors. This is not true. First, smartphone platforms evolve more rapidly (in 1-2 years), while the average lifetime of cars is more than a decade [6]; thus, smartphones will always have more modern sensors, which can include new types of sensors or more accurate sensors. Second, car sensors are specialized for vehicular control, not for context sensing, so it is likely that the general purpose sensors on phones may be more appropriate for some sensing tasks. Finally, it is unclear that cars will co-opt phone sensors: especially for mass market vehicles, adding new sensors can be expensive since these need to be engineered for long lifetimes and may require careful design and engineering.

A degree of freedom available to both approaches is *crowd-sourcing*. Crowd-sourcing vehicle context is quite practical for both car-sensing and phone-sensing. For many years now, some car manufacturers have had continuous telemetry systems for trouble shooting (e.g., GM's OnStar [17] system. These systems relay car sensor data to a cloud service. More generally, users can share traces of their car sensor readings or phone sensor readings via a cloud service, since both cars and phones are equipped with cellular connectivity. For example, crowd-sourced navigation systems like Waze collect GPS traces from users.} While crowd-sourcing raises privacy concerns, we consider this design dimension in this paper in order to understand how much crowd-sourcing would benefit vehicular context sensing if and when privacy concerns are addressed. Crowd-sourcing provides spatial coverage (e.g., data from multiple cars can detect a road anomaly across a larger area), can increase detection confidence, and can help disambiguate contexts (e.g., traffic light vs. stop sign).

Finally, an important constraint in this design space is *phone position*: users can place phones in positions which can reduce sensing accuracy for specific tasks, so understanding how position impacts the accuracy of context sensing is essential to an exploration of the design space.

Contributions and Findings. In this paper, we make three contributions.

Design space exploration. We provide a preliminary understanding of the design space of vehicular context sensing by exploring four qualitatively different case studies of vehicular context: lane change detection, pothole detection, road-grade estimation and stop sign determination. These contexts are qualitatively different in the sense that one of them measures driver behavior, another assesses the state of road infrastructure, a third measures a feature of the topography and a fourth measures a traffic regulation device. Moreover, stop sign and road grade represents persistent road features that do not change over years, while a pothole is a road feature that could be updated in weeks and lane change detection reflects the highly dynamic nature of driver behaviors on roads.

Novel context sensing algorithms. For each of these context sensing tasks, we design efficient car-sensing and phone-sensing algorithms. In all cases, we design novel car-sensing algorithms: to our knowledge, no one else has explored the design of stop-sign detection, lane change detection and pothole detection by using previously proprietary car sensors accessible via the vehicle CAN bus. Moreover, our design of crowd-sensing for each of these tasks is also novel, as is our exploration of the impact of phone position. For lane-change detection and road-grade estimation, our phone-sensing algorithms are also novel. Moreover, for each task, at least one of these algorithms has high accuracy (85% and above).

Results. Using empirical traces collected from multiple drivers in different locations, we evaluate the accuracy of these algorithms in order to understand whether one approach (car-sensing or phone-sensing) strictly dominates the other.

	Car / Phone	Phone Position*	Crowdsourcing
Lane Change	Car >> Phone	C > P > W	Improves performance in curvature (+40% recall +6% precision)
Pothole	Car >> Phone	W ≈ C >> P	Reduces False Positives (+27% precision) Increases coverage (+20% recall)
Road Grade	Car ≈ Phone	P > C > W	Corrects inaccurate measurements due to acceleration and deceleration (-10% error)
Stop Sign	Car ≈ Phone	W ≈ C ≈ P	Differentiates traffic lights *vs.* stop signs (+15% precision)

*C: cup-holder, P: pocket, W: windshield

Fig. 1—Summary of Findings

Our findings (Figure 1) suggest that neither approach is strictly better than the other, but that crowd-sourcing is essential for both. For example, car-sensing is superior for lane change determination primarily because the wheel angle sensor can unambiguously determine shift maneuvers. However, just because the car has a specialized sensor, that does not mean phones cannot achieve comparable accuracy: for lane change determination, although there exists a specialized yaw rate sensor that can be used to compute lateral displacement of the vehicle, phone sensors perform well in determining this quantity also. For each of these algorithms, crowd-sourcing plays a crucial, but qualitatively different role: in some cases, it increases the confidence of the detection, in other cases it provides spatial coverage, helps compute an unknown quantity, or disambiguates between two contexts that have similar manifested behaviors. Finally, we find that different phone-sensing algorithms are sensitive to the position of phone in different ways. Drivers may mount phones on the windshield, keep it in a cup-holder, or inside their pocket. We find, for example, that a windshield mount is pathologically bad for lane change detection because the phone's gyroscope is adversely affected by the car vibrations.

Collectively, our results suggest that, going forward, developers of algorithms for vehicular context should actively seek to fuse phone and car sensor information, use crowd-sourcing in designing vehicular context sensing, and carefully explore the impact of phone position on accuracy.

II. Methodology

We consider four vehicular context sensing tasks: determining when a driver has executed a *lane change* maneuver; determining the locations of *potholes* and other anomalies on a road surface; estimating *road grade*; and determining whether (and in which direction) an intersection is governed by a *stop sign*. All of them can plausibly be detected either using only phone sensors or only car sensors. All of them also depend on tracking motions or micro-movements of the vehicle: longitudinal speed changes for stop signs, lateral movements for lane changes, vertical movements (bumps) for potholes and tilt for road grade.

However, they are also qualitatively different among some dimensions. They sense different types of vehicular contexts (anomalies, topography, vehicle dynamics etc.). Some make binary decisions (lane changes), others estimate continuous values (road grade). Prior work has explored some tasks extensively (potholes), but others to a lesser extent. However, none have explored the broader question comparing car-sensing and phone-sensing approaches, in part because prior work has not had access to car sensors.

Our methodology is empirical. For each task (e.g., lane change determination), we design one sensing algorithm using car sensors alone. We then design a similar algorithm (to the extent possible) using phone-sensors alone. This approach enables a head-to-head comparison between the two approaches for each task, which we evaluate using traces from several hundred miles of driving. By examining situations where one approach succeeds and the other does not, we are able to get specific qualitative understanding of the strengths and weaknesses of each of these approaches.

Of course, such an approach can never be complete because the space of possible vehicular contexts is large. Our results are thus not intended to be definitive, but rather to take a first step towards understanding this design space.

Both car and phone sensor algorithms can benefit from *crowd-sensing*: using *traces of sensor readings obtained from other vehicles*. But, because the availability of these sensors and their accuracy can differ between cars and phones, the precise methods by which crowd-sourced information is used can differ between car-sensing and phone-sensing, and the benefits of crowdsourcing can also be different between these two approaches. For each algorithm, we devise a crowd-sensing component designed to increase its accuracy.

Putting it all together (Figure 2), we have designed a crowd-sensing platform that collects vehicle and phone sensors. Using this, we have collected hundreds of miles of traces, to evaluate the design space of various individual vehicle context detection tasks.

Finally, the relative accuracy of car-sensing and phone-sensing depends on two other key factors discussed below.

A. Sensor Availability and Accuracy

The same context can often be derived from different sensors, but the achieved accuracy usually varies. For example, when and whether a car is turning can be estimated from inertial sensors, but a steering wheel angle sensor usually can give more accurate information about slight turns. The relative accuracy of car and phone sensing therefore depends on the extent to which different types of sensors are available to applications on the car and phone platform. Even when the same type of sensor (e.g., an accelerometer) is available on both platforms, however, the accuracy of each sensor reading and the update rate can vary between the phone and the car.

In this paper, we have obtained access to several car sensors on late-model GM vehicles and compared them with a standard set of Android smart-phone sensors to derive vehicle movement. Figure 3 lists the sensors that we considered for the context detection tasks that are described in this paper. Each of the vehicle sensors can be accessed, in near real-time, on a smartphone using a Bluetooth enabled dongle in the OBD-II port of the vehicle. While a few of the sensors listed (e.g., vehicle speed or outside air temperature) have been available as part of the OBD-II standard on most vehicles, the majority of these sensors report their readings in a proprietary format on the CAN bus. Access to such sensors is only becoming gradually available to external applications through special vehicle manufacturer developer programs. We have used an extended version of the CarMA software [16, 21] to collect traces of these sensor readings for our evaluations, from several different vehicles.

The car provides a fairly complete set of sensors that describe different driver actions such as activating turn signals, turning the steering wheel, or opening the throttle, which are unavailable on the phone. Both platforms carry GPS and inertial sensors for measuring vehicle motion. However, the phone platform tends to provide higher update rates and contain a more complete set of inertial sensors.

B. Sensor Placement and Movement

The accuracy of sensor readings can further depend upon the exact location and orientation of the sensor in the vehicle. Examples include readings from inertial sensors but also GPS receivers, where the location and orientation of the antenna has a significant effect on the received signal strength. Further, while car sensors are generally mounted at a fixed position, the phone position is often unknown and dependent on driver behavior. The phone position might even change while driving, if the phone slides or is moved by its user.

To understand how the accuracy of phone sensing depends on phone position and movement, we consider three possible positions: in a windshield mount, in the cup-holder, and in the driver's pocket (right side). These choices represent commonly used positions that exhibit different movement characteristics. In the windshield mount, the phone is mounted to the vehicle body. In the cup holder the phone can slide occasionally when larger acceleration forces act on the vehicle. In the pocket position, the phone can be frequently affected by leg and body movements of the driver. (We have chosen the right pocket, because we expect more frequent movements corresponding to gas and brake pedal use).

In all of these positions, the orientation of the phone in the world coordinate frame and the vehicle coordinate frame is not precisely known. When this information is needed, we estimates the orientation as follows.

World Coordinate Frame Transformation. The Android SensorManager API provides a `getRotationMatrix()` function that estimates the orientation of the device in the world

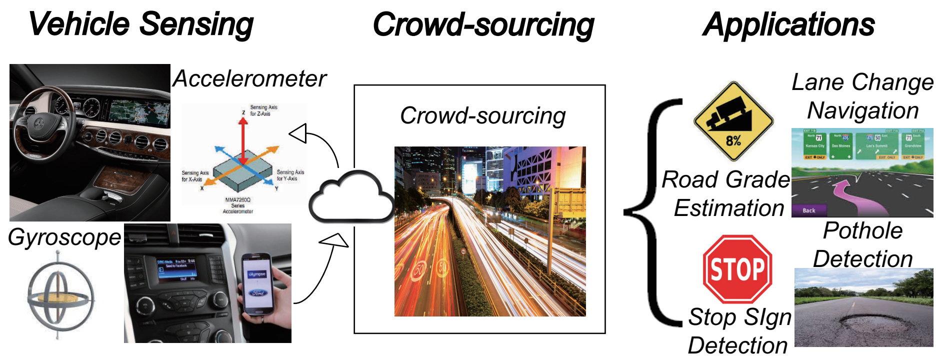


Fig. 2—Crowd-sourcing both car-sensors and phone-sensors to perform various vehicle context detection tasks.

	Car-sensor		Phone-sensor	
	Frequency	Resolution	Frequency	Resolution
Brakes Active	100 Hz	-	-	-
Shifter Position	40 Hz	-	-	-
Vehicle Lateral Acceleration	50 Hz	5.0E-06 m/s^2	~50 Hz	1.0E-11 m/s^2
Vehicle Longitudinal Acceleration	-	-	~50 Hz	1.0E-11 m/s^2
Vehicle Vertical Acceleration	20 Hz	1.0E+00 m/s^2	~50 Hz	1.0E-11 m/s^2
Vehicle Yaw Rate	50 Hz	5.0E-04 deg/s	~50 Hz	1.0E-11 rad/s
Vehicle Orientation Z / Azimuth	-	-	~50 Hz	1.0E-11 rad
Vehicle Orientation X / Pitch	-	-	~50 Hz	1.0E-11 rad
Vehicle Orientation Y / Roll	-	-	~50 Hz	1.0E-11 rad
Vehicle Speed	10 Hz	1.0E-04 mph	~10 Hz	-
Left Turn Signal	Event	-	-	-
Right Turn Signal	Event	-	-	-
GPS Latitude	1 Hz	3.0E-07 deg	~10 Hz	1.0E-12 deg
GPS Longitude	1 Hz	3.0E-07 deg	~10 Hz	1.0E-12 deg
Outside Air Temperature	2 Hz	1.0E-01 ℉	-	-
Throttle Position	10 Hz	1.0E-05	-	-
Steering Wheel Angle	100 Hz	5.0E-03 deg	-	-
Barometer	2 Hz	500 Pa	~30 Hz	0.01 Pa

Fig. 3—List of vehicle CAN sensors and derived phone sensors

coordinate frame based on accelerometer and magnetometer readings. It essentially uses gravity and the earth's magnetic field to estimate the device rotation. In this world coordinate frame, the y-axis points to the magnetic north pole and the z-axis points to the sky.

Vehicle Coordinate Frame Transformation. In the vehicle coordinate frame, the x, y, and z-axis are mapped to the lateral, longitudinal, and vertical axis of the vehicle itself, and can be different from the world coordinate frame. We use the coordinate transformation algorithm presented in [37] to estimate the phone pose in the vehicle coordinate frame. The algorithm first filters the acceleration readings to identify the gravity force, which generates the first unit vector. The second unit vector is obtained by monitoring the axis along which acceleration and deceleration occur when driving on a straight road. By the right hand rule, the third unit vector is orthogonal to the first two. This algorithm provides us with the rotation matrix R, which can be used to rotate the phone's alignment to match the vehicle coordinate frame.

III. Vehicular Context Detection

In this section, we discuss car-sensing and phone-sensing algorithms for the four context detection tasks discussed in Section I. For space reasons, we present only enough detail in our algorithms to help the reader understand the results presented in the evaluation in the following sections.

A. Lane Change Detection

Detecting a lane change is difficult, since lane changes can be conflated with road curvature and with weaving within a lane. Our algorithms address these by (a) finding a segment of the trace (called the *shift segment*) that contains a *shift maneuver*, and (b) measuring the lateral displacement of the vehicle within the shift segment. The first step accounts for curvature and the second deals with weaving behavior within a lane. Both algorithms use crowd-sourced information.

Isolating Shift Maneuvers. We use two algorithms to identify shift maneuvers, one each for car-sensing and phone-sensing. To our knowledge, these algorithms, and their use of crowd-sourcing, is novel.

Car-sensing: Our algorithm for lane shift determination for car-sensing is motivated by Figure 4 which shows the raw vehicle sensor values of the yaw rate (the angular velocity of the car about the vertical axis) and steering wheel angle, as well as other inertial sensors from the phone. During a lane change, the angular velocity first increases (or decreases depending on the direction of the lane change), then decreases until it crosses zero in the other direction. Intuitively, at this point, the car is at the point of crossing the lane. Beyond this, the yaw rate decreases some more and returns back to zero. This corresponds to the car straightening up in the target lane, and is the key to distinguishing between lane changes and turns at an intersection. The steering wheel angle is positively correlated with the yaw rate and exhibits a similar behavior.

Our detection algorithm declares any segment that contains this sinusoidal pattern to be a potential shift segment (i.e., one in which a shift maneuver occurs). It uses the steering wheel angle sensor for this purpose since that sensor shows a more pronounced pattern. An ideal algorithm for identifying the shift interval (t_1, t_2) on a straight lane is: (a) when the wheel angle at t_1 and t_2 are zero (i.e., the car is heading in the same direction at the beginning and at the end), The shift segment interval can be long or short depending on the driver's propensity, so we need a technique to verify that a shift maneuver corresponds to a lane change; we use the lateral displacement calculation below for this.

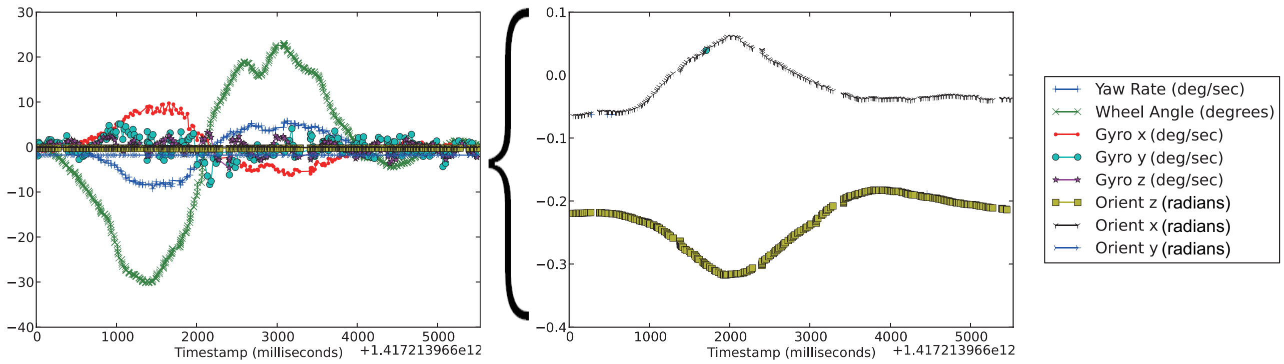


Fig. 4—Relevant car-sensor and phone-sensor pattern during a lane change on a straight road segment. Generally, rotation descriptors, such as yaw rate, wheel angle and gyro-meter display a sinusoid pattern, while absolute orientation sensor performs only a half of the sinusoid.

To identify a shift segment when the car is on a curved road, we exploit the insight that *crowd-sourcing can be used to determine what the wheel angle for other cars was at the locations corresponding to* t_1 *and* t_2 (call these locations l_1 and l_2). The key challenge here is to establish a baseline for the sensor pattern corresponding to the curvature without lane changes. Specifically, we take the median wheel angle from other traces, traces without lighting turn signals, in $\mathcal{S}$ at l_1 and l_2 (call them w_1 and w_2 respectively). With the baseline, we can revise our ideal algorithm for identifying the interval (t_1, t_2) as an interval containing a shift maneuver as: (a) when the wheel angle at t_1 is w_1 and at t_2 is w_2 and (b) the difference between wheel angle sensor readings and the crowd value exhibits a sinusoidal pattern. To deal with sensor noise, if two wheel angles are within a small fudge δ_w of each other, we declare them to be the same.

Phone-sensing: Our phone sensing algorithm detects shift segments using changes to the car's orientation, as computed from the phone's inertial sensors (gyroscope, magnetometer, and the accelerometer [5]). During a shift maneuver, one expects orientation to increase first, then decrease until it reaches the original heading. We use this intuition to identify the shift segment (t_1, t_2) in a manner similar to that for car-sensing.

For phone-sensing as well, curved roads pose a problem, but crowd-sourcing helps. In this case, we could take the orientation readings at any location l between l_1 and l_2 from the crowd-sourced traces $\mathcal{S}$, and use these in a manner similar to that discussed above. However, this requires that all phones are mounted consistently with the same frame of reference, which may not be the case since the phones can have random poses when sensing in a car. Rather than transforming the absolute orientation to the same vehicle frame, which can introduce error, we compute only the relative deviation from the curve by comparing the change in the phone's orientation and the change in the curve's, the latter of which is obtained using the radius of curvature computed from crowd-sourced traces (described later in Equation (1)). Our final algorithm for identifying a shift interval (t_1, t_2) is: (a) the maximum deviation in orientation of the car between t_1 and t_2 is comparable to δ_o, and (b) the orientation difference increases and then decreases between t_1 and t_2.

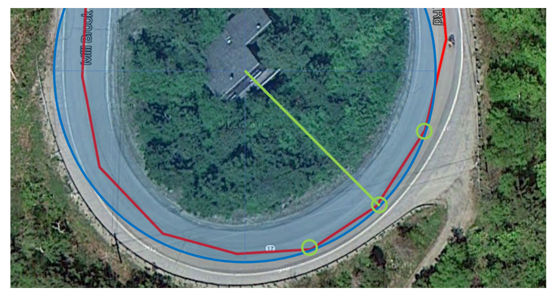


Fig. 5—Curvature Data from Digital Maps: Sequences of coordinates (green circles) define the paths of roads (in red). Digital maps provide curvature as the radius of the circle (in blue) which the closest three points form.

Computing Displacement. For both car and phone sensors, we compute the displacement within a shift interval the same way: using the yaw rate (for the phone, this is computed from the gyroscope sensor after performing appropriate coordinate transformations to account for differently oriented phones). On a straight road, we can integrate the yaw rate sensor ω to calculate the total angular displacement (the total change in heading) $\theta(t)$ at any time t within the shift interval. Then, integrating vehicle speed (either from the car sensor or from phone GPS) with respect to different angle ($\sum_{t=t_1}^{t_2} v(t)sin(\theta(t))\Delta(t)$), we can compute the total lateral displacement. If this displacement approximately equals the standard lane-width, we declare a lane change has occurred.

Detection on Curve: To account for road curvature, one straightforward approach is to use existing digital maps that provide road curvature data. Digital maps, however, use sequences of coordinates to define the paths of roads. Specifically, the way these map services provide curvature is to use the radius of the circle which the closest three points form. Figure 5 shows the connected dots (in red) that define a curve and one example curve radius (in green). This approach has two drawbacks: it cannot provide accurate curvature if all of the coordinates are not at the center of the lane; the granularity of the curvature data is dependent on the density of the defining points. For example, a slightly curved highway on the digital maps can have only a few sparse defining points which are hundreds of meters away from each other. Besides interpolation, which is often unreliable, there are no definitive way for digital maps to provide curvature data between two neighbouring defining points.

Instead, to get a fine-grained detailed curvature description, we compute, from crowd-sourced traces, the angular velocity component $\bar{\omega}$ that can be attributed to the curve. To do this, we assume that, during a short time interval at a given location, the radius of curvature of the lane is uniform. Then, we estimate the average radius of curvature for each trace at that location and use the average radius estimation to compute the angular velocity induced by the curvature at that location. Given the radius $\bar{R}$ and a vehicle instantaneous speed v, the angular velocity component $\bar{\omega}$ can be estimated by the speed divided by the radius of curvature (Equation (1)),

$$\bar{\omega} = \frac{v}{\bar{R}} = \frac{v}{\frac{1}{N_S}\sum_{i \in S} \frac{v_i}{\omega_i}} \tag{1}$$

where N_S is the number of crowd traces in S, v_i and ω_i is the linear and angular velocity of trace i at the same location. To estimate lateral displacement $x(l)$ at location l, we subtract from the car's angular velocity at a given location, the angular velocity $\bar{\omega}$ induced by the curvature at that location, then use the procedure discussed above (Equation (2)).

$$x(I) = \sum_{i=0}^{I} v\Delta t(i) \sin\left(\sum_{j=0}^{i} (\omega - \bar{\omega})\Delta t(j)\right) \tag{2}$$

B. Pothole Detection

Car repair costs from potholes are estimated to be \$6.4 billion annually [2], and potholes can cause accidents [28]. Detecting potholes is difficult: other road surface anomalies like expansion joints, railroad, potholes, speed bumps, curbs can induce similar vibration patterns as potholes; and different cars (or even the same car during different drives) may experience different vibration patterns from the same pothole differently, depending on the exact angle of impact.

The goal of pothole detection is to identify, in each trace, each location l that marks a pothole on the road. We detect potholes from sensors that measure vertical acceleration, and disambiguate them from other road surface anomalies by observing that potholes can have asymmetric impact on a vehicle. Finally, we use crowd-sourcing to increase detection confidence.

Phone-sensing for pothole detection has been extensively studied [14, 11, 27, 15] and has resulted in a commercially available app (Street Bump) for pothole detection, which we use in this paper. In the rest of this section, we describe our *car-sensing* algorithm for pothole detection, which, to our knowledge, has not been described in the literature before.

Detecting Vertical Acceleration. Cars contain a *Rough Road Magnitude* (RRM) car sensor, which continuously measures (at 2Hz, Figure 6) the deviation of the car's vertical acceleration (caused by, say, hitting a pothole) from its at-rest baseline value. To minimize the impacts caused by minor road surface

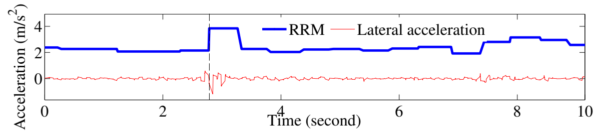


Fig. 6—RRM and lateral acceleration signal when car hits a pothole. The vertical dashed-line marks the pothole.

irregularities and inherent sensor noise, our algorithm only considers RRM sensor measurements above a threshold value (τ_v, determined from extensive training data) as the triggering condition for vertical detection.

Detecting Asymmetric Impulse. Road anomalies like small cracks or expansion joints can also generate substantial vertical acceleration. We observe, however, that most potholes have irregular shapes and are of limited size, so they usually impact only one side of the car wheels at a time, slightly tilting the car to the other side. This tilting can be measured by the car's lateral acceleration sensors (Figure 6). To accurately detect the lateral tilting effect caused by potholes, we calculate the peak-to-peak value of the lateral acceleration within the window where the RRM sensor is above τ_v, then compare it against a lateral acceleration determination threshold whose value we determine from training data.

Crowd-Sourcing. To increase detection confidence, we flag a pothole at a location l only if a majority of traces that pass l detect a pothole at that location l.

C. Road Grade Estimation

Road-grade measurements can be used to optimize cruise control fuel efficiency settings [3] or as input to a stability control system in estimating sideslip [32]. Road grade can be estimated from elevation changes, using either barometric sensors or inertial sensors. There are web services that, given a GPS location, output an elevation. In our experience these are not fine-grained enough, for example, to form inputs to stability control systems. We are unaware of any public available accurate and fine-grain road grade data. We obtained survey maps from the LA Department of Transportation, but found that these maps have only coarse-grained elevation measurements. Moreover, as of this writing, *no car sensors can estimate road grade accurately*. Some cars have a barometer, but these have poor resolution. For example, in a 2008 Cadillac CTS, the resolution of barometric pressure is 0.5kPa which is approximately equivalent to 40 meters elevation change at sea level. The inertial sensors are insufficient for road-grade estimation. For example, our test vehicle has a lateral acceleration sensor, no longitudinal acceleration sensor, and a processed vertical acceleration sensor designed for a specific task (rough road measurement). Therefore, in the rest of this section, we discuss phone-sensing algorithms that can provide fine-grained and robust road-grade measurement for vehicles.

Phone-sensing. Our phone-sensing algorithm makes novel use of a combination of inertial and barometric sensors. Inertial road-grade measurements are most precise when there are no external accelerations acting on the vehicle (when it is moving at a constant speed or is stationary). The barometer can estimate road grade in an accelerating vehicle, but can be affected by local air currents. We propose to *combine these two sensing approaches* to obtain accurate road grade measurements, *using measurements either from the same car or using crowd-sourcing*. The accelerometer can correct for any discrepancies in the barometer under no acceleration conditions, and the barometer can continue to estimate road grade under regular acceleration and deceleration conditions. The phone can determine whether the car is accelerating or decelerating by transforming the accelerometer readings to the vehicle's frame.

The atmospheric air pressure obtained from a barometer on a phone can be converted to elevation using a standard pressure-height equation [26]: $h = 44330 * (1 - (\frac{p}{p_0})^{\frac{1}{5.255}})$. Here, p_0 is the air pressure at the sea level and p is the measured air pressure at current location. Once elevation changes are known, road grade can be determined using differences in height of successive readings, and the distance traveled.

For an inertial sensor mounted with its axis aligned to the direction of vehicle movement and gravity, measuring road inclination translates to computing the pitch angle of the sensor. Pitch is defined as the forward tilt of the device and can be obtained from the accelerometer readings on a smartphone. These readings are first transformed into the vehicle frame of reference discussed earlier.

Then, as the car moves up an incline, gravity now has components on both the y and z axis (with respect to the car frame). The pitch angle, α, is calculated around the x-axis as, $\alpha = \arctan(A_y/A_z)$. Here A_y and A_z are the raw accelerometer readings along the y and z axis respectively, while α represents the road grade.

D. Stop Sign Detection

A stop sign detection algorithm must address several challenges: drivers rarely come to a full stop; stopping can be conflated with congestion or traffic lights; and any detection algorithm must distinguish 2-way and 4-way stop signs. Our algorithms are based on detecting a prevalent characteristic of stopping at a stop sign: a deceleration followed by an acceleration. They address other challenges either using map information, or crowd-sourced traces.

Determining Stops. To determine a stop pattern, our car-sensing and phone-sensing algorithms identify a *stop segment* within a trace where a stop is most likely to have occurred.

Car-sensing: Figure 7 shows the timeseries of several car sensors at a stop sign. This figure motivates the following algorithm to identify a stop segment: (a) the segment begins at the point where the brake sensor transits from being active to being inactive, (b) it ends at the rising edge of the throttle position, and (c) the car speed reaches zero during some point in the interval.

This is an idealized description. Some drivers may not come to a complete stop, so we use a small speed threshold: if the speed is below this threshold, a stop is said to have occurred. Moreover, a car may stop several times if it is queued up behind other cars at the stop sign. In this case, we use the last speed reading before the rising edge of the throttle position in order to make a stop determination.

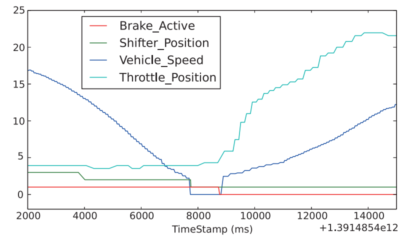


Fig. 7—Typical Relevant Sensor Pattern Passing a Stop Sign

Phone-sensing. Phones do not, of course, have access to sensors that directly measure human activity (braking etc.). Motivated by Figure 7, we use the vehicle speed to determine stop segments. We use the haversine formula [19] to derive estimated speed from two successive GPS coordinates. Then the estimated speed is obtained by dividing the distance by the difference of the timestamps associated with each coordinate. Other elements of the algorithm are similar to car-sensing.

Disambiguation. To distinguish stopping at a stop sign from other stopping activity, we use map information: to qualify as a stop segment, the car's location must be within a distance threshold of an intersection (as determined from an online map). To distinguish from congestion-related stops, a significant fraction of stop segments must exist at intersection I before that intersection is marked as having a stop sign. Finally, to distinguish between 4-way and 2-way stop signs and between stop signs and street lights, *we use crowd-sourcing*. If there exists a stop segment S at intersection I, but k other traces with the same heading as S (where k is a small integer) that do not contain a stop segment at I we say there is *no* stop sign at I in that direction.

IV. Evaluation

We use the four previously described context sensing applications to evaluate the relative accuracy of car-sensing and phone-sensing, both with and without crowd-sourcing for the best phone position for the given sensing task. We then evaluate how accuracy for these tasks varies with phone position. To conduct these experiments, we have built infrastructure that continuously captures car and phone sensor readings, uploads them to a cloud database, and computes spatial indexes to improve query speeds. Describing this infrastructure is beyond the scope of this paper.

A. Car-sensing vs. Phone-sensing

1) Lane Change: **The Dataset.** To evaluate lane change algorithm, we collected traces from six different drivers both on a flat and straight urban road (dataset **Straight**) and a hilly area with straight and curved road segments (dataset **Curve**). In each experiment, a passenger collected ground truth measurements by explicitly recording lane changes made by the driver using a custom-built mobile app. In total, our traces cover around 200 miles, containing over 300 instances of lane changes for which we have ground truth, so we use

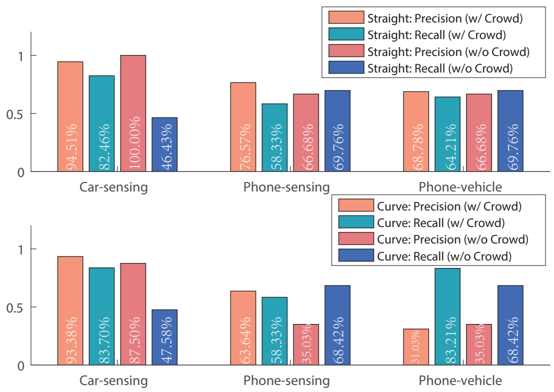


Fig. 8—*Car-sensing* vs *Phone-sensing*: Lane Change Detection

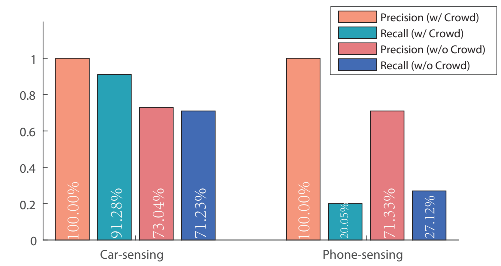


Fig. 9—*Car-sensing* vs *Phone-sensing*: Pothole Detection

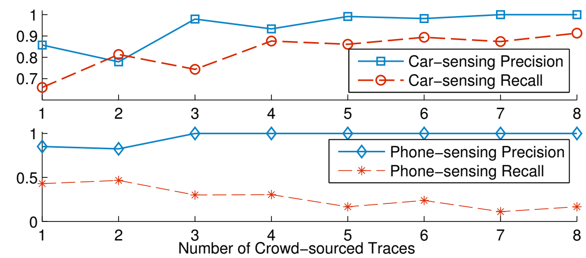


Fig. 10—Crowd-sourcing Contribution for Pothole Detection.

these to compare car-sensing and phone-sensing. To extract crowd-sourced road curvature, we use 20% of the traces.

For phone-sensing, for reasons discussed below, we rigidly mounted the phone on the center console. Then, we transformed the inertial sensors to the world frame of reference. We also report results for *phone-vehicle*, an alternative in which inertial sensor readings are transformed to the vehicle frame discussed earlier.

Results. Since our lane-change algorithm is essentially a binary classifier (did a lane-change happen or not?), we use standard measures of accuracy for binary classifiers, *precision* and *recall* [25]. Figure 8 discusses the results of our evaluation. It is interesting that our novel car-sensing algorithm has high precision both on **Straight** (94.51%) and **Curve** (93.38%) roads. Crowd-sourcing further significantly improves recall in curvy road (from 47.58% to 83.70%) where curvature is unknown in previous work [12]. In contrast, phone-sensing has significantly lower performance, especially recall (58.33%). When transforming sensor readings from global to vehicle coordinate frame (the phone-vehicle case), straight road has similar phone-sensing precision (68.78%) and recall (64.21%). However, motion sensor errors introduced by curvy roads affect the accuracy in determining the second unit vector, which could potentially reduce algorithm precision. Thus, car-sensing performs significantly better than phone-sensing for lane-change determination.

The insight for this performance difference is as follows. Both car-sensing and phone-sensing are able to robustly compute lateral displacement. Even though the car-sensor has a dedicated yaw rate sensor that is designed to provide angular velocity about the vertical axis, the phone's inertial sensors are also able to achieve comparable accuracy with careful re-orientation and compensation. The real difference in the results comes from the shift maneuver determination step. The wheel angle sensor, which measures shift maneuvers directly, can be used to accurately estimate these maneuvers even on curved roads, but this step is much less accurate when using the orientation sensors on the phone.

We have also evaluated the efficacy of crowd-sourcing for this task. It turns out that crowd-sourcing is crucial, especially in the dataset **Curve**, where most lane changes happen on curved segments. In this case, detection precision is improved from 87.50% to 93.38%, recall from 47.58% to 83.70%. Crowd-sourcing provides an accurate description of road curvature (either through curve radius or road orientation) which other traces can use as a baseline to estimate lane displacements; without this, as other work has shown [12], it is hard to estimate lane changes. By contrast, the benefits of crowd-sourcing for phone-sensing is less-evident (precision is improved by 28.61% (9.89%) in dataset **Curve** (**Straight**), with 10.09% (11.43%) recall trade-off; estimating curvature from crowd-sourcing is less accurate in this case, since the coordinate frame transformation introduces significant error. Today's maps do not have road curvature information at sufficiently fine granularity for our purposes (and it's not clear they ever will), so crowd-sourcing will likely play an important part in lane change determination.

2) Pothole: **The dataset.** Our dataset was collected on a stretch of 4-mile road segment with various types of potholes. Simultaneously, we also firmly mounted the smart phone on the windshield to collect detection results from the Android street bump application [11]. This application records the pothole traces, including timestamps, GPS locations and the smartphone accelerometer measurements. For ground truth identification, we used another windshield-mounted smartphone to record the video during the entire data collection. We manually identified, by inspecting the collected videos, a total of 23 potholes on this four-mile road; the overhead of manual identification limits the scale of experiments we can do in this case. For evaluating our crowd-sourcing steps, we collected multiple traces (8) on this road segment, among which 15 (10%) random selected pothole encounter are used for both training and testing data.

Results. For a similar reason as lane-change determination, we use precision and recall to evaluate our pothole detection algorithms. Figure 9 shows the average precision and recall of

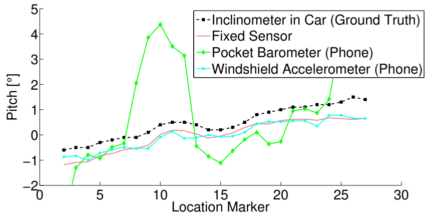


Fig. 11—Flat Road Experiment

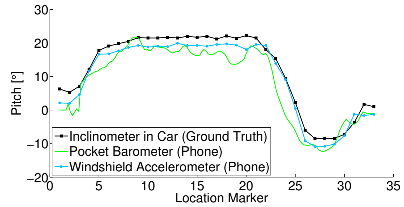


Fig. 12—18° Inclined Road Experiment

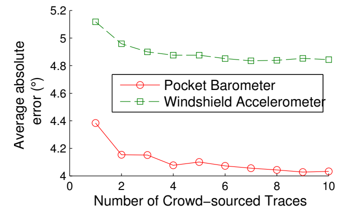


Fig. 13—Crowd-sourcing

pothole detection. Without taking crowd-sourcing into account, both car-sensing and phone-sensing have similar average precision (73.05% and 71.33%). The phone-sensing based approach has much less recall because it fails to detect many small potholes. However, with crowd-sourcing, our car-sensing based pothole detection has 100% precision and 91.28% recall. Thus, crowd-sourcing improves the precision and recall by about 30%. In contrast, for phone-sensing approach, while it has very high precision (about 100%) with crowd-sourcing, but its recall performance is fairly dismal (27%) at higher levels of crowd-sourcing. This is because the crowd-sourcing does not drastically improve the already inferior recall performance of phone-sensing. Figure 10 shows the detection results of both car-sensing and phone-sensing after crowd-sourcing. In this figure, the x-axis represents x randomly chosen traces, and we report the precision and recall averaged over these x random choices. This illustrates the benefit of increasing levels of crowd-sourcing.

The drastic difference between car- and phone-sensing approach is primarily because the phone sensors are much less sensitive to the road vibrations and can only detect very significant potholes (even though the phone is mounted on the windshield). In other words, car-sensing has higher accuracy because cars have specially engineered sensors calibrated and positioned to detect rough road conditions and lateral accelerations (since these sensors are important for stability control). The improvement from crowd-sourcing in accuracy comes from the fact that not all vehicles traversing a lane will encounter the pothole depending on where the pothole is; crowdsourcing improves spatial coverage.

3) Road Grade: **The Dataset.** To evaluate the efficacy of the road grade sensing techniques, we conducted experiments along two selected roadways of different grades. One was a nearly flat road, while the other had an 18° incline. We collected two datasets, with ten traces each, on these streets. We marked 30 locations on each road segment, separated by a meter. One dataset was collected by coming to a standstill at each marker. At this point we recorded the ground truth by placing an inclinometer on the car floor, obtained accelerometer data from the fixed sensor and from the smartphone and then moved to the next spot and repeated the process. Recall that, in the real world this data can be collected when the car is moving at a constant speed or is stationary. Our second dataset was obtained by simply driving on this road segment with no stopping. At each iteration, we collected the ground truth from the car, the barometer readings, the accelerometer in the sensor and the smartphone.

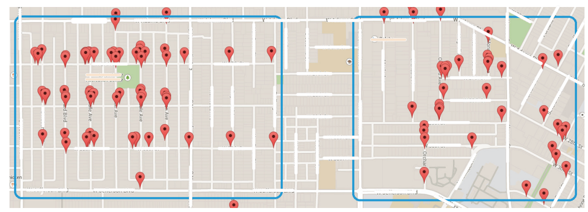


Fig. 14—Google Map Augmented with Stop Sign.

Results. We assess the accuracy of road-grade algorithms by measuring the error with respect to ground truth. The calculated road grade is shown in Figures 11 and 12 for flat and inclined road experiments, respectively. In Figure 11, the fixed sensor and accelerometer data from the phone (windshield mounted position) was collected using the first dataset (with stops), and the barometer dataset (driver's pocket position) was collected using the second dataset (no stops). It is evident from this experiment that road grade estimations from the emulated car sensor (fixed) and smartphone inertial sensors are very close to the ground truth. The barometer, however, does not work well for small variations in road grade, and exhibits large errors. This may have been caused by frequent, sudden accelerating and braking. For the 18° incline, with the accelerometer in windshield position and barometer in the pocket position, it is evident that both the barometer and the inertial sensor measurements are inline with the ground truth.

To examine if crowd-sourcing can provide us with better accuracy, we compute road grade using the barometer and inertial sensors for our second dataset (no stops). We calculated these values using a different number of traces each time and computed the error. As evident from Figure 13, the average error for the barometer approach improves slightly with crowd-sourcing, but is not significantly affected. It must be noted that in the continuous driving dataset, the inertial sensor readings at the beginning and end of a trace are not accurate due to acceleration and deceleration of the vehicle. This causes a small error in accelerometer measurements, that corrects itself as the number of traces increases.

In summary, road-grade estimation is a context sensing task that can be accurately implemented using phone sensors, but cannot be realized using currently available car sensors.

4) Stop Sign: **The Dataset.** We collected traces from 6 different drivers, during different times of day and different days of a week over a period of around 9 months. The traces

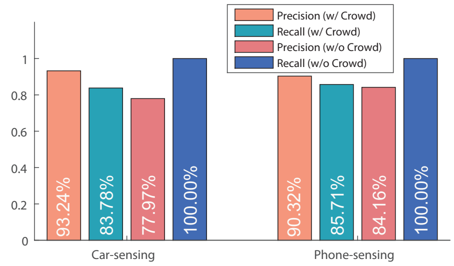


Fig. 15—*Car-sensing* vs. *Phone-sensing*: Stop Sign Detection

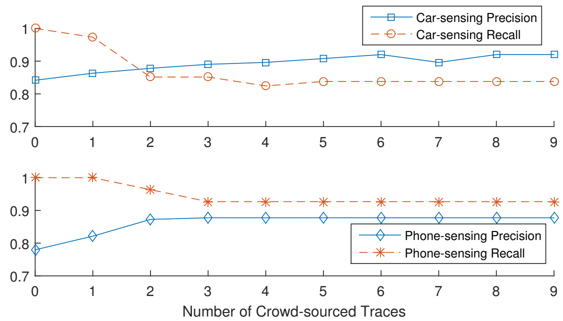


Fig. 16—Crowd-sourcing Contribution for Stop Sign Detection

Fig. 17—Phone Position: Windshield, Cup-holder, Pocket

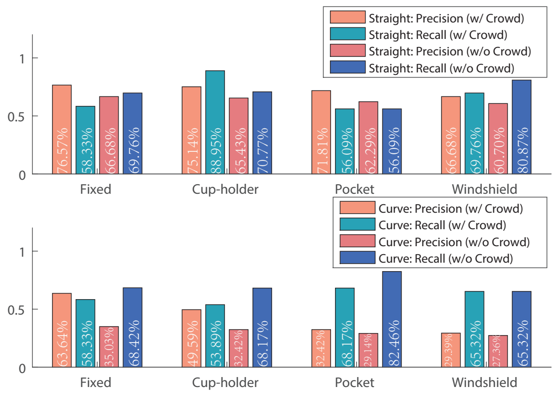


Fig. 18—Phone Position Sensitivity: Lane Change Detection

cover over 500 miles, with 11 traffic light, 74 stop signs at 55 intersections, among which 14 of them are two way stop signs, 16 are one way stop signs, and the rest 4-way stop signs. We collected ground-truth by recording stop signs as they encountered them, using a custom-built mobile app. Figure 14 shows the groundtruth stop signs detected.

Results. In our evaluation, even though car-sensing (precision 93.24%, recall 83.78%) uses more dedicated sensors such as the brake and throttle, phone-sensing has comparable precision (90.32%) and recall (85.71%) (Figure 15). Phone-sensing has slightly lower precision when a vehicle passes through a green light at speeds lower than the speed threshold, yielding false positives. Car-sensing does not suffer from this problem because it uses additional signals: the brake and the throttle. Thus, for this task, it appears that phone sensing and crowd sensing are qualitatively similar.

Furthermore, crowd-sensing appears to play an important part in increasing the accuracy of stop sign detection (Figure 16). For car-sensing, crowd-sourcing increases precision by nearly 15% but commensurately reduces recall, due to potential inappropriate stop sign behaviors, such as not decreasing the speed low enough. For phone-sensing, crowd-sourcing increases precision by 6% and reduces recall less significantly. Moreover, we also find that car-sensing needs fewer crowd-sourced traces to converge to its highest accuracy than phone-sensing: this is because the car-sensors can generally detect stops more accurately by directly measuring breaking and throttling activity, requiring less disambiguation.

B. Sensitivity to Phone Positions

Phone-sensing performance has assumed a favorable fixed position. To understand how the phone-sensing results change when the phone is carried in a less favorable position, we now revisit each of the applications and compare the phone sensing results across the windshield mount, cup-holder, and driver pocket positions (see Figure 17).

Lane Change. Figure 18 shows that the precision and recall for the phone-sensing lane change detection varies significantly with phone positions. The cup-holder performance (75.14%) is close to the original fixed position (76.57%), while the pocket and windshield positions show degraded performance, particularly on curved roads.

One might expect the highest performance with a rigid mounting to the vehicle body and performance to diminish when the phone is in the drivers pocket and subject to driver movements. We were surprised, however, by the relatively poor performance of the windshield mount. We now suspect that the mount amplifies vibrations that affect the gyroscope readings, a cornerstone of the algorithm that is used to calculate the lateral displacement.

Pothole. Figure 19 shows the phone-sensing pothole detection performance for different phone placements. While the performance of the windshield and cup-holder positions is quite close, the pocket position is an outlier: the phone barely detects any potholes at all in this position. We believe that this is because the bump is largely absorbed by the seat and human body.

If the crowd-sourcing mechanism is not engaged, we also observe that the position of windshield mounted phone has similar precision to the position of cup-holder, while recall at the windshield is much worse than in the cup-holder. We attribute this to the cup-holder being close to the center of the vehicle and therefore feeling bumps on any of four wheels. In contrast, the windshield mounted phone is biased towards the

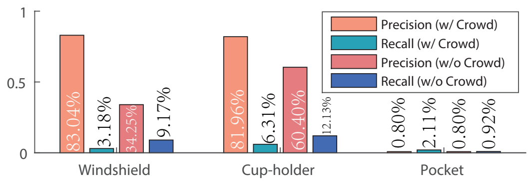


Fig. 19—Phone Position Sensitivity: Pothole detection

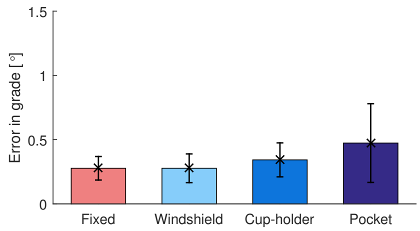


Fig. 20—Phone Position Sensitivity: Road Grade Estimation.

front wheels and may not detect the bump if only a rear wheel hits the pothole.

It is also worth noting that crowd-sourcing reduces both precision and recall for phone-sensing approach, in both the cup-holder position and windshield position. Our hypothesis is that phone-sensing in these positions is likely to produce inconsistent detection results across different traces, but our current crowd-sourcing mechanism (based on majority voting) requires consistent observations to produce a consensus. We have left a detailed understanding of this to future work.

Road Grade. Figure 20 shows the pitch errors encountered with the inertial phone-based road grade estimation across different phone placements. We concentrate here on the inertial approach since phone placement is unlikely to affect barometer sensors. We observe that the windshield mounted position provides us with results that are comparable to the fixed inertial sensor unit (which emulates an embedded car sensor). The pitch error with respect to the ground truth is about 0.25° in both cases and could likely be further reduced through improved calibration. However, the error in road grade estimation increases when the phone is placed in the cup holder or in the driver's pocket. This can be caused by small changes in the phone orientation due to leg movement or sliding in the cup holder. Note, however, that even in the pocket position the mean error is only about 0.5°.

Stop Sign. Figure 21 shows the performance of phone-sensing based stop sign detection across the different phone positions. The results are not very sensitive to phone placement. Since our algorithm only uses GPS and not inertial sensors, it appears that the phone was able to receive a sufficiently strong GPS signal in all positions during our experiments. One might expect that the results do become more sensitive to phone placement in situations when the GPS signal quality is diminished.

The implications of these results are summarized in the next section.

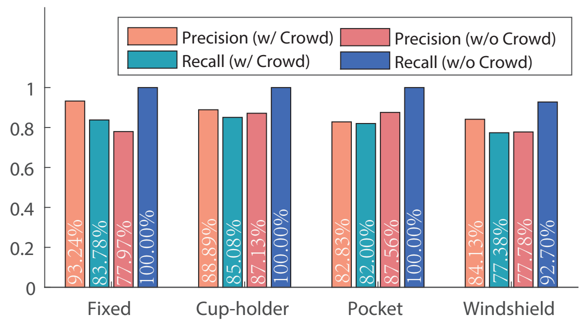


Fig. 21—Phone Position Sensitivity: Stop Sign Detection

V. LESSONS LEARNED

Our primary finding is that neither car-sensing nor phone-sensing alone is likely to satisfy all applications, and that a hybrid-sensing approach, in which car sensors are paired with phone sensors will be necessary to compute vehicular context. An equally important result is that crowd-sensing of these hybrid traces can significantly improve accuracy.

This finding is important because the choice between phone and car sensors will not go away in the future. Smartphone platforms evolve more rapidly (in 1-2 years), while the average lifetime of cars is more than a decade (11.4 years in 2012 [6]); thus, smartphones will always have better sensors than cars. Second, car sensors are specialized for vehicular control, not for context sensing (this is discussed in more detail below), so it is likely that the general purpose sensors on phones may be more appropriate for some sensing tasks. Finally, cars are unlikely to co-opt phone sensors: especially for mass market vehicles, adding new sensors can be expensive since these need to be engineered for long lifetimes.

Our four case studies highlight the importance of hybrid-sensing and crowd-sourcing. In lane change determination, car-sensing outperforms phone sensing but for a very subtle reason: even though it has a dedicated yaw rate sensor, phone sensors can equally well compute lane displacements. What really makes a difference is the fact that the phone sensors cannot reliably measure when a shift maneuver has actually happened, while the car sensors have a direct measure of this quantity. In pothole detection, the presence of well engineered sensors in the car that directly measure frame vibrations resulting from rough road, and also lateral acceleration, helps car sensing be much more accurate. In road grade estimation, our study shows the opposite result: in this case, phones have general-purpose barometer and inertial sensors that are quite accurate in estimating road grade, but at least the cars we had access to do not have any sensors that we could have used for road-grade determination. Finally, in stop sign detection, car-sensing and phone-sensing performed comparably well. Even though there are specialized sensors in the car to directly measure stopping activity initiated by a driver, phone sensors perform quite well in part because crowd-sourcing compensates for the fact that phone sensors can only indirectly measure stopping activity.

A second interesting lesson that emerges is the design philosophy of sensing between the car and the phone. Cars have a large number of sensors, some of which are aggregated or processed virtual sensors from some underlying physical sensors. A good example is the yaw rate sensor, which returns the angular velocity about the vertical axis, computed from an on-board gyroscope. This gyroscope, however, is not directly accessible. Phones, on the other hand, have a few sensors to which software has direct access (the gyroscope and the accelerometer are examples). This is not surprising: cars have not been designed for programmability, but phones have, and phone sensors are intended to serve several different applications, while car sensors are designed to serve specific control needs, and were not originally intended to be exposed to external software dynamically. This is another reason why we believe that the hybrid-sensing model is the one that is most likely to meet the needs of vehicular context sensing.

A challenge in incorporating phone-sensing, lies in its sensitivity to phone position. The best phone position depends on the exact measurements taken. A rigid windshield mount generally works well but performs poorly in applications that depend on precise inertial measurements while the car is moving, since the windshield mount can act as a lever and amplify vibrations. With the phone in the drivers' pocket, the accuracy is generally reduced compared to more rigid phone positions. The performance in this position is particularly poor for vertical acceleration measurements (e.g., pothole detection) since the seat and body dampen the vertical forces. In our experiments, the cup-holder position showed the most consistent results across applications but it carries the risk that the phone itself will move inside the cup-holder when stronger acceleration forces act on the car.

We also learned that crowd-sourcing helps different algorithms in different ways. For lane changes, crowd-sourcing helps compute a various spatial quantity, namely curvature, curve orientation, etc.. For pothole detection, crowd-sourcing helps increase detection confidence, and for road-grade it can enhance spatial coverage.

These observations are qualitatively reinforced by another task, *stop-sign detection*, for which we designed car-sensing algorithms, and used an existing phone-sensing algorithm. We have omitted a detailed discussion of this task, for space reasons. However, for this task, phone-sensing performs comparably to car-sensing, and both algorithms are insensitive to phone position. Crowd-sourcing can significantly affect precision and recall for this task, since it can be used to distinguish between stop signs and traffic lights (where a significant fraction of vehicles do not stop at the intersection).

These observations also point to opportunities to support developers of vehicular context through system services and context sensing frameworks. Most important, such infrastructure should facilitate hybrid-sensing with phone and car sensors but also allow for crowd-sensing. It should accommodate the need for trace augmentation, the derivation of a type of sensor information from other related sensors, when a specific sensor is unavailable. In addition, such infrastructure should offer mechanisms for detecting and adjusting to different phone placements and orientations. Our experience suggests that designers of vehicular contexts can leverage such capabilities for a broad range of future vehicular context sensing applications.

VI. Related Work

Lane Change Detection. The automotive industry has incorporated vision-based lane departure sensors [1] inspired, in part, by lane marker detection algorithms from the computer vision community [4, 18]. In general, these approaches are known to suffer from occlusion and poor visibility.Other work has used smartphone inertial sensors to detect vehicle dynamics [38, 12, 13], such as detecting turns and phone poses [38], or detecting turns, curves, lane changes [12] and abnormalities such as weaving, swerving, side-slipping, U-turn [13] on straight roads. Dongyao *et al.* [12] proposed using lateral displacement to detect lane changes. In contrast, our paper discusses the first design for lane change detection both for straight and curved roads, using inertial and other sensors from both vehicle and mobile devices. Both our car-sensing and phone-sensing approach can detect lane changes on curved roads with novel crowd-sourcing techniques.

Road Surface Anomaly Detection. Road surface assessment used a variety of sensing technologies. Vision-based pothole detection [23, 24] is sensitive to ambient light, while laser imaging (LiDAR) techniques [39] and sound pressure-based techniques [15] are expensive. In an early accelerometer-based approach [14], potholes were detected using a high resolution accelerometer mounted to the vehicle. This line of work has led to a mobile app [11], which we use for our comparisons. Another piece of work [27] proposed a phone-sensing based approach for pothole detection. We are aware of no other work that has attempted to quantify the efficacy of car-sensing based pothole detection.

Road Grade Estimation. Road grade is important information widely used in various vehicle applications [32, 3]. Several existing road grade estimation approaches rely on vehicle kinematic information [36, 34] but require knowing vehicle mass, which can vary with loading, or require other aspects of vehicle geometry and assume limits to road grade [8]. High accuracy GPS is has also been used to estimate grades with or without inertial sensors [30, 7, 32, 33], but it is known that GPS exhibits more than 10m inaccuracy in obstructed environments. Elevation data from cloud services [35] can be used to estimate roadgrade, but are often erroneous on multi-level road infrastructures. Prior work has used specialized barometers to estimate the road grade [9], sometimes to complement GPS elevation estimation [31]. In contrast, our work explores the efficacy of phone-based road-grade estimation, using barometric sensing and inertial-sensing.

Stop Sign Detection. Previous work [10, 20] collected GPS traces for a specific set of intersections, and differentiated stop signs from traffic lights, using either heuristics or machine learning. Our phone-sensing algorithm is inspired by theirs, and our accuracy results are comparable to theirs, but their work does not incorporate car-sensing.

VII. Conclusion

In the near future, detecting vehicular context, a monetizable quantity, will become an important problem. Mobile operating systems for vehicles will allow apps access to hitherto proprietary vehicle sensors and be able to link with mobile phones.

We provide, to our knowledge, the first analysis of context sensing based on internal vehicle sensors and its comparison with phone-sensing algorithms for a variety of qualitatively different vehicular context sensing tasks, all of which have several applications. Overall, we find that one approach does not dominate another and that phone sensing would benefit from better techniques to compensate for phone position. Thus, a hybrid model, where car manufacturers partner with mobile device manufacturers to develop applications and methods for determining context, and make heavy use of crowd-sourcing, is likely to be most effective in the future for detecting vehicular contexts.

BIBLIOGRAPHY

[1] *2015 chevy tahoe lane departure warning system*. https://www.youtube.com/watch?v=gola2DE5vao (cit. on p. 12).

[2] *AAA: pothole damage costs drivers 6.4 billion a year*. http://www.wusa9.com/story/news/nation/2014/02/24/potholes-damage-cost-us/5773501/ (cit. on p. 6).

[3] M.A Abou-Nasr. "Dynamic adaptation of a vehicle's cruising speed with recurrent neural networks for enhancing fuel economy". In: *Proceedings of IEEE International Conference Cybernetics (CYBCONF)*. 20013 (cit. on pp. 6, 12).

[4] Mohamed Aly. "Real time detection of lane markers in urban streets". In: *Intelligent Vehicles Symposium, 2008 IEEE*. IEEE. 2008, pp. 7–12 (cit. on p. 12).

[5] *Android orientation sensor*. http://developer.android.com/intl/zh-tw/reference/android/hardware/SensorManager.html (cit. on p. 5).

[6] *Average age of U.S. car, light truck on road hits record 11.4 years, Polk says*. http://www.autonews.com/article/20130806/RETAIL/130809922/average-age-of-u.s.-car-light-truck-on-road-hits-record-11.4-years (cit. on pp. 2, 11).

[7] Hong S Bae, JIHAN RYU, and J Christian Gerdes. "Road grade and vehicle parameter estimation for logitudinal control using GPS". In: *Proceedings of IEEE Intelligent Transportation Systems Conference*. 2001 (cit. on p. 12).

[8] Jörg Barrho et al. "Estimation of elevation difference based on vehicle's inertial sensors". In: *International Federation of Automatic Control*. 2005 (cit. on p. 12).

[9] Behdad Yazdani Boroujeni, H Christopher Frey, and Gurdas Singh Sandhu. "Road Grade Measurement Using In-Vehicle, Stand-Alone GPS with Barometric Altimeter". In: *Journal of Transportation Engineering* 139.6 (2013) (cit. on p. 12).

[10] Roberto Carisi et al. "Enhancing in vehicle digital maps via GPS crowdsourcing". In: *Wireless On-Demand Network Systems and Services (WONS)*. IEEE. 2011 (cit. on pp. 1, 12).

[11] F. Carrera, S. Guerin, and J.B. Thorp. "BY THE PEOPLE, FOR THE PEOPLE: THE CROWDSOURCING OF "streetbump", AN AUTOMATIC POTHOLE MAPPING APP". In: *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. 2013 (cit. on pp. 6, 8, 12).

[12] Dongyao Chen et al. "Invisible Sensing of Vehicle Steering with Smartphones". In: *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*. MobiSys '15. 2015 (cit. on pp. 8, 12).

[13] Zhongyang Chen et al. "D 3: Abnormal driving behaviors detection and identification using smartphone sensors". In: *Sensing, Communication, and Networking (SECON), 2015 12th Annual IEEE International Conference on*. IEEE. 2015, pp. 524–532 (cit. on p. 12).

[14] Jakob Eriksson et al. "The Pothole Patrol: Using a Mobile Sensor Network for Road Surface Monitoring". In: *Proceedings of the 6th International Conference on Mobile Systems, Applications, and Services*. MobiSys '08. 2008 (cit. on pp. 6, 12).

[15] D.C. Festa et al. "First results of a new methodology for the identification of road surface anomalies". In: *Proceedings of IEEE International Conference on Service Operations and Logistics, and Informatics (SOLI)*. 2013 (cit. on pp. 6, 12).

[16] Tobias Flach et al. "CarMA: towards personalized automotive tuning". In: *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*. ACM. 2011, pp. 135–148 (cit. on pp. 1, 3).

[17] *GM onStar*. https://www.onstar.com/ (cit. on p. 2).

[18] Raghuraman Gopalan et al. "A Learning Approach Towards Detection and Tracking of Lane Markings". In: (2012) (cit. on p. 12).

[19] *Haversine Formula*. http://www.movable-type.co.uk/scripts/latlong.html (cit. on p. 7).

[20] Shaohan Hu et al. "SmartRoad: a crowd-sourced traffic regulator detection and identification system". In: *2013 ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN)*. IEEE. 2013, pp. 331–332 (cit. on p. 12).

[21] Yurong Jiang et al. "CARLOG: A Platform for Flexible and Efficient Automotive Sensing". In: *Proceedings of the 12th ACM Conference on Embedded Network Sensor Systems*. SenSys'14. 2014 (cit. on pp. 1, 3).

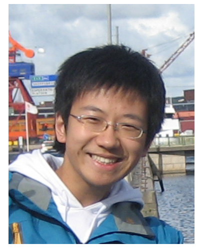
[22] Seungwoo Kang et al. "Seemon: scalable and energy-efficient context monitoring framework for sensor-rich mobile environments". In: *Proceedings of the 6th international conference on Mobile systems, applications, and services (Mobisys'08)*. ACM. 2008 (cit. on p. 1).

[23] J. Karuppuswamy et al. "Detection and avoidance of simulated potholes in autonomous vehicle navigation in an unstructured enviornment". In: *Proceedings of Intelligent Robots and Computer Vision XIX: Algorithms, Techniques, and Active Vision*. Vol. 4197. 2000 (cit. on p. 12).

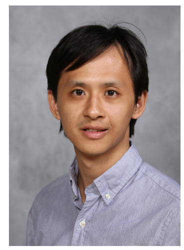
[24] Christian Koch and Ioannis Brilakis. "Pothole Detection in Asphalt Pavement Images". In: *Adv. Eng. Inform.* 25.3 (Aug. 2011) (cit. on p. 12).

[25] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*.

New York, NY, USA: Cambridge University Press, 2008 (cit. on p. 8).
[26] Greg Milette and Adam Stroud. *Professional Android Sensor Programming*. Wiley, 2012 (cit. on p. 7).
[27] Prashanth Mohan, Venkata N. Padmanabhan, and Ramachandran Ramjee. "Nericell: Rich Monitoring of Road and Traffic Conditions Using Mobile Smartphones". In: *Proceedings of the 6th ACM Conference on Embedded Network Sensor Systems*. SenSys '08. 2008 (cit. on pp. 1, 6, 12).
[28] *Motorcyclist killed after hitting pothole*. http://woodtv.com/2014/05/18/motorcyclist-loses-control-of-bike-dies-in-crash/ (cit. on p. 6).
[29] Suman Nath. "ACE: exploiting correlation for energy-efficient and continuous context sensing". In: *Proceedings of the 10th international conference on Mobile systems, applications, and services (Mobisys'12)*. ACM. 2012 (cit. on p. 1).
[30] Xiaoji Niu et al. "Using inertial sensors of iPhone 4 for car navigation". In: *Position Location and Navigation Symposium (PLANS), 2012 IEEE/ION*. 2012, pp. 555–561 (cit. on p. 12).
[31] J Parviainen et al. "Barometer-aided road grade estimation". In: *Proceedings of 13th IAIN World Congress*. 2009 (cit. on p. 12).
[32] J Ryan, D Bevly, and Jianbo Lu. "Robust sideslip estimation using GPS road grade sensing to replace a pitch rate sensor". In: *Proceedings of IEEE International Conference on Systems, Man and Cybernetics (SMC)*. 2009 (cit. on pp. 6, 12).
[33] Jihan Ryu and J Christian Gerdes. "Integrating inertial sensors with global positioning system (GPS) for vehicle dynamics control". In: *Journal of Dynamic Systems, Measurement, and Control* 126.2 (2004), pp. 243–254 (cit. on p. 12).
[34] Per Sahlholm and Karl Henrik Johansson. "Road grade estimation for look-ahead vehicle control using multiple measurement runs". In: 18 (11 2010) (cit. on p. 12).
[35] *The Google Elevation API*. https://developers.google.com/maps/documentation/elevation/ (cit. on p. 12).
[36] A. Vahidi, A. Stefanopoulou, and H. Peng. "Recursive least squares with forgeting for online estimation of vehicle mass and roadgrade: theory and experiments". In: (2011) (cit. on p. 12).
[37] Yan Wang et al. "Sensing Vehicle Dynamics for Determining Driver Phone Use". In: *Proceeding of the 11th Annual International Conference on Mobile Systems, Applications, and Services*. MobiSys '13. 2013 (cit. on p. 4).
[38] Jie Yang et al. "Detecting driver phone use leveraging car speakers". In: *Proceedings of the 17th annual international conference on Mobile computing and networking (Mobicom'11)*. ACM. 2011, pp. 97–108 (cit. on p. 12).
[39] X Yu and E Salari. "Pavement pothole detection and severity measurement using laser imaging". In: *Proceedings of IEEE International Conference on Electro/Information Technology (EIT)*. 2011 (cit. on p. 12).

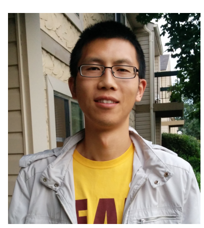


Hang Qiu is a Ph.D. student in Electrical Engineering at the University of Southern California. He received his B.S. degree from Shanghai Jiao Tong University. His research interests include networked intelligence in IOT systems, mobile / edge computing, wireless networking, deep learning and crowdsourcing systems. He has interned at Microsoft Research, Redmond and IBM T.J. Watson Research Center. He has published in venues such as ACM Sensys, ACM Mobihoc, ACM HotMobile, IEEE TVT. He has served as a reviewer for IEEE TMC, IEEE VNC.



Jinzhu Chen is a Researcher at General Motors Research & Development, Michigan, USA. He received the Bachelor's degree in communication engineering from University of Electronic Science and Technology of China, Sichuan, China, in 2005, and received the Ph.D. degree in computer science from Michigan State University, USA, in 2014. From 2005 to 2009, he was an embedded software engineer at Delphi China Technical Center, Shanghai, China. His research interests include wireless sensor networks, cyber-physical systems, vehicular sensing, mobile and cloud computing.

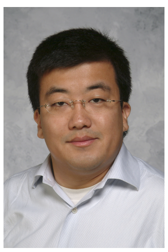
Shubham Jain is an Assistant Professor in the Department of Computer Science at Old Dominion University. Her research interests lie in mobile computing, cyber-physical systems, and data analytics in smart environments. She received her MS and PhD in Electrical and Computer Engineering from Rutgers University in 2013 and 2017, respectively. Her research in pedestrian safety received the Large Organization Recognition Award at AT&T Connected Intersections challenge, and has featured in various media outlets including the Wall Street Journal. She also received the Best Presentation Award at PhD Forum, held with MobiSys 2015. She has interned at Microsoft Research, Redmond and General Motors Research. She has published in venues such as ACM MobiSys, ACM/IEEE IPSN, IEEE PerCom etc.



Yurong Jiang received the Ph.D. degree in Computer Science from University of Southern California, Los Angeles, CA, in 2016. He is now a researcher in Hewlett-Packard Labs, Palo Alto. His research focuses on building end-to-end system to solve interesting real world problems. His research topics include Mobile Computing, Cloud Computing and Deep Learning Systems.

Matt McCartney is a Software Engineer at Applied Minds. While pursuing a MS degree in Computer Science at USC, He was a research assistant in the Networked Systems Lab, and completed a thesis on vehicle sensing with Ramesh Govindan. He received his BS degree in Computer Science from Wright State University while interning at the AFRL Discovery Lab with Robert Williams. His interests relate to transportation and the application of geographic information systems.

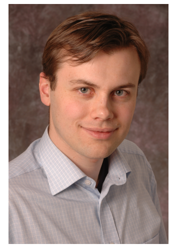
Gorkem Kar was born in Ankara, Turkey in 1987. He received the B.S. and M.S. degrees in Electrical and Electronics Engineering from Bilkent University, Turkey, in 2009 and 2011, respectively. He received his PhD. degree in Electrical and Computer Engineering from Rutgers University, USA, in 2017. His previous works were published in ACM/IEEE conferences.



Fan Bai is a Staff Researcher in the Electrical & Control Systems Lab., Research & Development and Planning, General Motors Corporation. He received the B.S. degree from Tsinghua University and the M.S.E.E. and Ph.D. degrees from University of Southern California. His current research is focused on design and analysis of protocols/systems for next-generation vehicular networks, for safety, telematics and infotainment applications. He published 90 research papers in top-quality conferences and journals including INFOCOM, SECON, Mobicom, Mobihoc, Sensys, IEEE JSAC, IEEE TMC, IEEE/ACM TON, IEEE TVT, and IEEE TWC. In addition, he published 1 book and 6 book chapters.

He received Charles L. McCuen Special Achievement Award from General Motors Corporation in recognition of his accomplishment in area of vehicle-to-vehicle communications for drive assistance & safety. He was featured as "ITS People" in 2014 by IEEE ITS Magazine for his technical contributions to vehicular networks and intelligent transportation systems. He serves as Technical Program Co-Chairs for IEEE WiVec 2007, IEEE MoVeNet 2008, ACM VANET 2011 and ACM VANET 2012. He is an Associate Editor of IEEE TVT and IEEE TMC, and he also serves as guest editors for IEEE Wireless Communication Magazine, IEEE Vehicular Technology Magazine and Elsevier AdHoc Networks Journal. He is a Fellow and Distinguished Lecturer of IEEE.

Donald K. Grimm is a Staff Researcher in the Connected Vehicle Systems group at General Motors Research & Development Center. He has over 25 years of experience at GM where he has worked on the development of in-vehicle networking protocols and simulation tools, and on the development of traditional and wireless-based safety systems. He is actively involved in the development of connected vehicle technologies at GM and in the development of wireless connectivity solutions for automated driving research projects at GM R&D. He has received 3 "Boss Kettering" awards and 4 Charles L. McCuen Special Achievement awards for his work in the development of advanced technologies at GM. He currently holds 43 US patents and has a M.S. degree in Computer Science from Oakland University, Rochester, MI.



Marco Gruteser is a Professor of Electrical and Computer Engineering as well as Computer Science (by courtesy) at Rutgers University's Wireless Information Network Laboratory (WINLAB). Dr. Gruteser directs research in mobile computing, pioneered location privacy techniques and has focused on connected vehicles challenges. He currently chairs ACM SIGMOBILE, has served as program co-chair for several conferences including ACM MobiSys and as general co-chair for ACM MobiCom. He received his MS and PhD degrees from the University of Colorado in 2000 and 2004, respectively, and has held research and visiting positions at the IBM T. J. Watson Research Center and Carnegie Mellon University. His recognitions include an NSF CAREER award, a Rutgers Board of Trustees Research Fellowship for Scholarly Excellence and six award papers, including best paper awards at ACM MobiCom 2012, ACM MobiCom 2011 and ACM MobiSys 2010. His work has been featured in the media, including NPR, the New York Times, the Wall Street Journal, and CNN TV. He is a senior member of the IEEE and an ACM Distinguished Scientist.

Ramesh Govindan is the Northrop Grumman Chair in Engineering and Professor of Computer Science and Electrical Engineering at the University of Southern California. He received his B. Tech. degree from the Indian Institute of Technology at Madras, and his M.S. and Ph.D. degrees from the University of California at Berkeley. His research interests include routing and measurements in large internets, networked sensing systems, and mobile computing systems.