BigRoad: Scaling Road Data Acquisition for Dependable Self-Driving

Luyang Liu[♮], Hongyu Li[♮], Jian Liu[§], Cagdas Karatas[♮],
Yan Wang[⋄], Marco Gruteser[♮], Yingying Chen[§], Richard P. Martin[♮]
(The first two authors are co-primary student authors.)
[♮]WINLAB, Rutgers University, North Brunswick, NJ, USA
[♮]{luyang, hongyuli, cagdas, gruteser, rmartin}@winlab.rutgers.edu
[§]Stevens Institute of Technology, Hoboken, NJ, USA
[§]{jliu28,yingying.chen}@stevens.edu
[⋄]Binghamton University, Binghamton, NY, USA
[⋄]yanwang@binghamton.edu

ABSTRACT

Advanced driver assistance systems and, in particular automated driving offers an unprecedented opportunity to transform the safety, efficiency, and comfort of road travel. Developing such safety technologies requires an understanding of not just common highway and city traffic situations but also a plethora of widely different unusual events (e.g., object on the road way and pedestrian crossing highway, etc.). While each such event may be rare, in aggregate they represent a significant risk that technology must address to develop truly dependable automated driving and traffic safety technologies. By developing technology to scale road data acquisition to a large number of vehicles, this paper introduces a low-cost yet reliable solution, BigRoad, that can derive internal driver inputs (i.e., steering wheel angles, driving speed and acceleration) and external perceptions of road environments (i.e., road conditions and front-view video) using a smartphone and an IMU mounted in a vehicle. We evaluate the accuracy of collected internal and external data using over 140 real-driving trips collected in a 3-month time period. Results show that BigRoad can accurately estimate the steering wheel angle with 0.69° median error, and derive the vehicle speed with 0.65 km/h deviation. The system is also able to determine binary road conditions with 95% accuracy by capturing a small number of brakes. We further validate the usability of BigRoad by pushing the collected video feed and steering wheel angle to a deep neural network steering wheel angle predictor, showing the potential of massive data acquisition for training self-driving system using BigRoad.

Keywords

Self-Driving, Road data acquisition, Smartphone, IMU

MobiSys '17, June 19–23, 2017, Niagara Falls, NY, USA.

DOI: http://dx.doi.org/10.1145/3081333.3081344

1. INTRODUCTION

Advanced driver assistance systems and, in particular, automated driving offer an unprecedented opportunity to transform the safety, efficiency, comfort, and economics of road travel. This has led many organizations in the computing and transportation domains to focusing on self-driving technology research. While this has resulted in a number of prototypes with impressive performance, it remains widely recognized that ensuring dependability under varied traffic conditions remains a key challenge [1, 2, 3].

Self-driving vehicles have to operate safely even under unusual or rare traffic events that are challenging to address and could lead to potential accidents. Developing such technology therefore requires an understanding of not just common highway and city traffic situations but also a plethora of widely different unusual events (e.g., objects on the roadway, pedestrian crossing highway, deer standing next to the road, etc.). While each such event may be rare, in aggregate they represent a significant risk that technology must address to develop truly dependable automated driving and traffic safety technologies. The average human driver achieves on the order of almost 100 million vehicle miles traveled per fatality [4]. Demonstrating driving performance at an above-average, advanced human driver level will therefore require successfully avoiding fatalities with unusual events that might be encountered within a billion miles of driving. This motivates the need for scaling road dataset to billions of miles of driving so that they contain a representative set of such unusual and rare events.

Most existing efforts to collect driving data build on a small fleet of tens of highly instrumented vehicles that are continuously operated with test drivers [5, 6, 7]. In terms of miles recorded, it is challenging to accumulate a sufficiently large dataset with this approach. From the tidbits of published information, we know, for example, that Google's fleet has completed about 2 million testing miles [5]—impressive but still far off from a billion miles. On the other hand, since many existing efforts to develop automated driving technology are proprietary, the data obtained is usually closely guarded. It is not easy for individuals outside vehicle industry to collect driving inputs such as steering wheel angle and pedal operations. The OpenPilot [8] project reverse en-

gineers the OBD-II data from two vehicle models of Hondas and Acuras, and uses the collected data to train the CNN based Adaptive Cruise Control (ACC) and Lane Keeping Assist System (LKAS). However, this solution can only be deployed in specific vehicle models, which makes it harder to recruit very large numbers of vehicles. Government-sponsored naturalistic driving studies have produced similar datasets, such as the UMTRI Naturalistic Driving Data [9] or the 100-vehicle VTTI Naturalistic Driving Study [10] that reached 2 million miles. These studies are more focused on human driving behaviors and still far below the target scope for supporting robust self-driving.

To address this challenge, this paper asks whether data useful for self-driving can be gathered with only minimal instrumentation of vehicles. Such a minimal vehicle instrumentation approach would enable scaling by capturing events from tens of thousands of vehicles rather than only attempting to collect data with a few highly instrumented vehicles or certain vehicle models, as is common practice. A key challenge in creating such minimal instrumentation is the heterogeneity of vehicle designs and the proprietary nature of in-vehicle systems. To be useful for driver assistance and automated driving applications, the dataset must capture the surrounding traffic situation of the vehicle and how the vehicle was driven through this traffic scenario (i.e., it's precise trajectory and the necessary driver input). The latter is especially important for approaches relying on machine learning, which is increasingly used in such systems [11]. Here, the driver input data provides important positive and negative training examples that allow the system to learn how to react to traffic situations. It is also useful for system validation, since it allows automated comparisons of the response of automated driving algorithms with those of a human driver. Any significant deviations can then be more closely examined.

The primary technical challenge is to obtain accurate vehicle movements without extensive instrumentation of the vehicle. While vehicles contain internal sensors to track steering and pedal inputs, the specific sensors vary among models and car makers use different proprietary data formats, if the information is exposed through the OBD-II port at all [12]. This heterogeneity renders scaling to many different vehicles difficult. Global Positioning System tracking of vehicles does not always capture fine-grained steering and speed changes, particularly in urban canyons. Gathering information about the surrounding traffic situation is more straightforward—in most situations, a front-facing camera can provide rich information. It is worthwhile complementing this information, however, in darkness and other situations where visual information may be insufficient.

BigRoad, as shown in Figure 1, aims to minimize instrumentation of vehicles by relying on low-cost inertial sensors that can be affixed to the vehicle. An inertial sensor in a dash-cam or windshield mounted smartphone enhances speed estimation, particularly in low-speed scenarios, therefore providing an indirect measurement of acceleration and braking inputs. A second steering wheel sensor gathers angle information, which allows a much more precise measurement of vehicle turning. To achieve this, BigRoad incorporates algorithms that isolate steering and vehicle movements from other forces acting on the vehicle. It also compensates for unknown orientations of the devices.

To understand whether such data can be useful for self-driving research, we study the performance of a key self-driving component, a self-steering algorithm. When trained with our data, this deep neural network-based algorithm takes road video as input and outputs the desired steering angle.

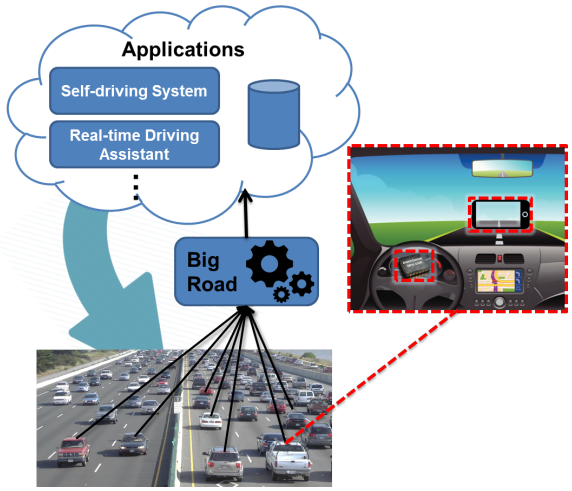


Figure 1: Illustration of BigRoad containing a smartphone and an IMU sensor: scaling road data acquisition.

The contributions of this work can be summarized as follows:

- To our knowledge, this study is the first to analyze whether driving data useful for self-driving research can be collect with a minimal set of inertial and video add-on devices, in contrast to the existing work with significant instrumentation for scaling data acquisition.
- Developing steering wheel angle estimation algorithms by leveraging low-cost sensing devices (i.e., inertial measurement units (IMUs) and smartphones). The key novelty of this approach is that it extracts steering wheel rotation by eliminating vehicle movement from the steering-wheel-mounted sensor measurements.
- Designing robust vehicle speed sensing algorithms that combine GPS with speed delay shifting, and an acceleration based complementary filter.
- Devising an acceleration-based road condition estimator to enhance road condition awareness under poor lighting conditions and to illustrate further uses of the sensed data.
- Collecting and analyzing 40 hours of driving data to determine the accuracy of the estimation techniques and to demonstrate that the fine-grained driving data provided by the proposed framework achieves comparable performance in an automated vehicle steering algorithm.

2. MOTIVATION & APPLICATIONS

Road and driving data are not only useful for constructing validation scenarios but can also support driving algorithm development. Automated vehicles increasingly rely on machine learning. Learning can be employed for specific subsystems such as recognition of traffic signs or traffic participants

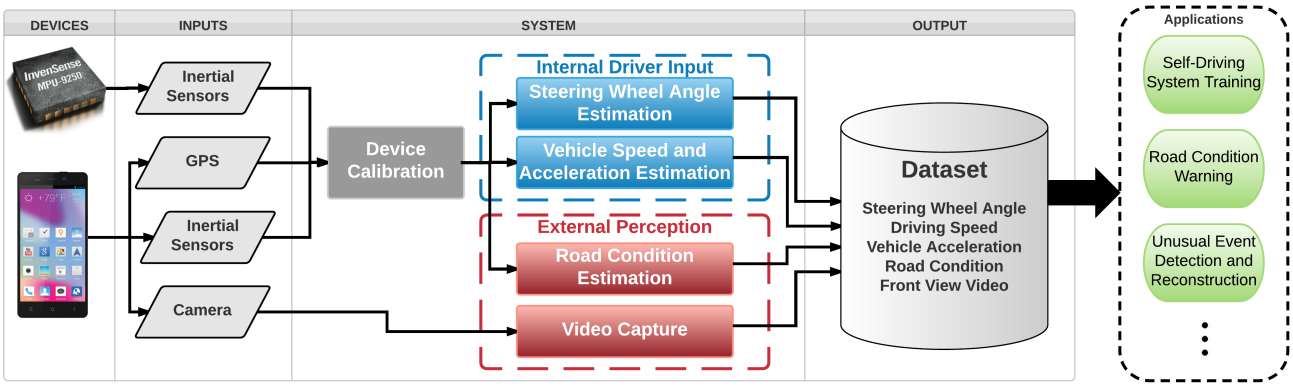


Figure 2: BigRoad system overview.

from video data. Learning is also increasingly explored for vehicle control. In fact, recent research [11] demonstrated how a deep neural network can support end-to-end automated steering. The network takes as input raw video data, without using a separate feature extraction step, and converts this into a steering control signal. Such approaches rely on an extensive dataset for training the network with both video and steering inputs from a human driver.

Furthermore, the massive road data could also support various real-time driving assistant applications. BigRoad can speed up innovation by providing an open large-scale datasets of unusual traffic events, which can be utilized to understand the various scenarios that driver-less vehicles need to be trained and tested on. For example, detecting bad weather conditions or adverse road situations and then adapting the driving parameters accordingly is an important task in self-driving cars. The data collected in BigRoad could serve as a training set for dealing with bad weather conditions and adverse road situations, thereby contributing to faster development of robust and reliable autonomous driving environments. When it rains or snows, not only does slick asphalt pose control issues that a vehicle has to deal with but the vehicle will also need to deal with reckless and negligent drivers who could be more dangerous under adverse road conditions. To help prevent and avoid collisions under adverse road conditions, the self-driving vehicle needs to make corresponding actions such as reducing the turning speed, braking gently and slowly, etc.

In particular, in order to take advantage of the driver inputs and road information collected from BigRoad, we design a crowd sourcing based application associated with BigRoad that can warn the driver with the waiting time of traffic light, as well as the road condition in front. The users of this application can contribute their driving input and traces through BigRoad to our server in real-time, and will in turns benefit from the drivers who uploaded data previously. Previous works demonstrate that the traffic light schedule can be trained based on the crowd sourcing video feed of the front facing camera [13] or the GPS data [14] from vehicles passing through the traffic light. We adopt the second method, which utilizes the crowd sourcing inertial sensor and GPS readings from the smartphone of multiple drivers to predict the schedule of each traffic light. For each end user, the application runs a stop detection and analyzes the vehicle facing direction based on the GPS location and bearing. Through Google Map API, the application could determine whether the driver is stopped for a traffic light and which light the driving is waiting for, then provide the predicted traffic light waiting time. In the same way, drivers could receive the road condition warning through the road condition model which is trained by the crowdsourcing driving data as introduced in Section 5.2.

3. SYSTEM DESIGN

The main goal of BigRoad is to provide a light-weight automated data logging system dedicated to crowdsourcing fine-grained driving data, which can facilitate extensive research in self-driving vehicles and driving safety monitoring. By installing very few off-the-shelf sensing devices (i.e., a smartphone and an IMU) in a vehicle, the system can track vehicles' dynamics, drivers' driving behaviors, and road environments in real time. In addition to logging raw sensing data (i.e., GPS locations, motion sensor readings, and vehicles' front-view video), BigRoad devises novel approaches to derive various fine-grained driving data (i.e., steering wheel angles, vehicle speeds, and vehicle accelerations) by fusing the measurements from various sensors.

The major advantage of BigRoad is that it provides a minimum-effort solution for self-driving companies and researchers, who want to collect large datasets of ready-to-use driving data in real world without concerns of different vehicle types and driving behaviors. Traditional data logging systems only provide coarse-grained sensing data without recording the ground truth. Our system leverages various sensing technology in smartphones and IMUs to provide fine-grained measurements and experimental ground truth in the context of real driving. The information generated by BigRoad is two-fold: 1) The estimated steering wheel angles (directly from drivers), vehicle speeds and accelerations (indirectly from drivers) are considered to be *Internal Driver Input* to reconstruct vehicles' motions and driving behaviors in a fine-grained manner. 2) The estimated road conditions and videos captured by the cameras of smartphones are considered to be *External Perception* that can capture driving environments and provide experimental ground truth.

BigRoad is realized with five main sub-tasks: *Device Calibration*, *Steering Wheel Angle Reconstruction*, *Vehicle Speed and Acceleration Estimation*, *Road Condition Estimation*,

and *Time-stamped Video Capture*. Figure 2 shows the system overview of BigRoad. To participate, the user needs to install a smartphone on the dashboard and an IMU on the center of the steering wheel as illustrated in Figure 1. The system takes measurements of inertial sensors (i.e., acceleration and rotation rate in both the IMU and smartphone) as inputs to *Device Calibration*, which derives the rotation matrices that can calibrate the sensing measurements from both devices to the vehicle's coordinate system, disregarding vehicle models, steering wheel positions and IMU placements. The components of Internal Driver Input including *Steering Wheel Angle Estimation* and *Driving Speed and Acceleration Estimation* fuse GPS location and calibrated sensing measurements from inertial sensors to derive steering wheel angles, vehicle speeds, and accelerations. The components of External Perception including *Road Condition Estimation* and *Time-stamped Video Capture* utilize video camera and inertial sensors to capture critical information in the driving environment (i.e., traffics, road conditions, etc.). Road conditions in inclement weather can severely affect traffic demands, roadway capacity and increase risks of having traffic crashes [15].

The insight of our steering wheel angle estimation is that after applying the *Device Calibration*, which is discussed in Section 6, the rotation angles of the IMU aligned to the steering wheel plane could be used to estimate steering wheel angles. However, accurately estimating the steering wheel angles involves several challenges. First, steering motions from human usually happen in a sudden and change all the time, which makes them extremely hard to be captured by the IMU's inertial sensors. Second, the measurements from the inertial sensors of the IMU are contaminated by motions of the vehicle, such as turning and braking. Third, although the IMU's position on the steering wheel is stable, it is unknown and could be different from time to time. To harness these challenges, We obtain the rotation of the steering wheel based on IMU's sensor readings and uses the smartphone's acceleration to remove the error caused by vehicle motion during driving.

Besides, BigRoad fuses GPS and acceleration from smartphone to predict accurate vehicle driving speed. Since road conditions in inclement weather can severely affect traffic demands, roadway capacity and increase risks of having traffic crashes [15], we propose to use real-time normalized traction force derived from inertial sensor readings during brakes activities to identify binary road conditions (i.e., dry or wet). Moreover, the *Time-stamped Video Capture* utilizes cameras of smartphones to provide the front-view video of the vehicle together with millisecond-granularity time stamps, which serves as the ready-to-use training datasets for various applications such as self-driving system training, road condition warning, and dangerous event detection and recommendation.

4. INTERNAL DRIVER INPUT

Let us now consider the specific techniques to improve the accuracy of estimating human driver inputs such as steering wheel angle and vehicle speed from generic IMU and smartphone sensors that can be widely deployed across vehicle models.

4.1 Steering Wheel Angle Estimation

As one of the most important driver inputs, steering wheel operations play a critical role in self-driving systems. Compared to vehicle speed that is openly available via a standard OBD-II interface, steering wheel operations (i.e., rotation angles) are usually harder to access since it can be carried on separate buses or encoded in proprietary formats only known to the vehicle manufacturers. In order to obtain drivers' steering inputs irrespective of vehicle models with minimal instrumentation, we propose two sensor-based approaches to estimate steering wheel angles: 1) steering wheel IMU based estimation and 2) phone based estimation. The difference between these two approaches is that the former requires an inertial measurement unit (IMU) attached to the steering wheel together with the smartphone in the vehicle, while the latter only needs one smartphone with its internal inertial sensors.

In particular, the steering wheel IMU based estimation includes three steps: 1) the system fuses steering wheel angles estimated based on different inertial sensors in the IMU by using a complementary filter; 2) it removes the angle errors caused by vehicle motions from the sensor measurements; 3) the system calculates the angle biases from the coordinate alignment, and further calibrates the estimated steering wheel angle by removing the biases. Next, we first introduce the three steps for the steering wheel IMU based estimation in section 4.1.1, 4.1.2 and 4.1.3, then we discuss the phone based estimation in section 4.1.4.

4.1.1 IMU Sensor Fusion

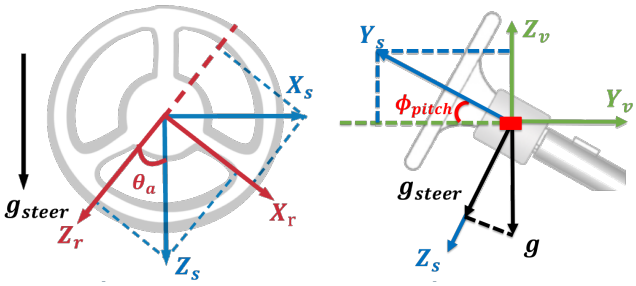
After coordinate alignment using the device calibration algorithms described in section 6, measurements from the inertial sensors in the IMU are aligned to the steering wheel's two-dimensional plane. Intuitively, we could estimate changes in the steering wheel angle by accumulating the angular velocities (ω) from the IMU's gyroscope. The angle change from time t_1 to t_2 ($\Delta\theta_{gyro}^{t_2|t_1}$) is equal to the integration of the ω readings in this period, as shown in $\Delta\theta_{gyro}^{t_2|t_1} = \int_{t_1}^{t_2} \omega dt$. While the gyroscope measures rotation changes precisely, the integration process suffers from accumulating errors and results in large drifts over time. We can also estimate steering wheel angles based on the angular changes derived from the gravity projected onto the different axis from the IMU's accelerometer (θ_a), but the accelerometer readings show significant vibration noise, which causes large dynamic errors. [1]

To address these limitations in both approaches, we devise a complementary filter [16] to fuse the estimated steering wheel angles based on the measurements from the IMU's accelerometer and gyroscope. The design of the complementary filter is shown with the following equation that applies both high-pass and low-pass filters:

$$\theta^{t|t-1} = cc * (\theta^{t-1} + \Delta\theta_{gyro}^{t|t-1}) + (1 - cc) * \theta_a, \qquad (1)$$

where $\theta^{t|t-1}$ is the estimated steering wheel angle at time t based on the angle estimated at time $t-1$ denoted as θ^{t-1}, $\Delta\theta_{gyro}^{t|t-1}$ is the estimated steering wheel angle change obtained by accumulating the IMU's gyroscope measurements around the axis of rotation from time t to $t-1$, θ_a is the steering wheel angle derived from the accelerometer's measurements, and cc is a variable that determines the time scale

[1]We have also considered exploiting magnetometer based approaches, but they are significantly affected by the surrounding magnetic field, which is usually unstable in urban environments.



(a) Steering wheel and rotated steering wheel coordinate system.

(b) Steering wheel and vehicle coordinate system.

Figure 3: Several coordinate systems and angle estimation.

of the high and low-pass filters, which is set to 0.9 empirically. Since the gyroscope integration process is relatively straightforward, we will put more emphasis on the derivation of θ_a.

Steering Wheel Angle Estimation based on Accelerations. We first consider the scenario when the vehicle is static, e.g., the vehicle is parked, in which gravity is the only force applied to the steering wheel. We examine the IMU's accelerations on the two dimensional steering wheel plane as illustrated in Figure 3(a). In the plane, we define two sub-coordinate systems: 1) X_s and Z_s denote the x and z-axis of the *steering wheel coordinate system*, and 2) X_r and Z_r denote the same in the *rotated steering wheel coordinate system*. We assume Z_s points to the the same direction with the gravity projection (g_{steer}) on this plane and y-axis of both sub-coordinate are perpendicular to this surface as shown in Figure 3(b). Note that the *steering wheel coordinate system* is fixed but the *rotated steering wheel coordinate system* changes with the rotation of the steering wheel and IMU. Therefore, the problem becomes how to determine the angle between two coordinate systems.

Intuitively, when there is only gravity acceleration in the static scenario, the steering wheel angle θ_a can be estimated by first calculating the arc-tangent over the projection of the gravity acceleration on X_r and Z_r as shown in $\theta_{static} = atan2(a_{X_r}, a_{Z_r})$, where a_{X_r} and a_{Z_r} can be derived from the IMU's acceleration readings aligned from its own coordinate system to the *rotated steering wheel coordinate system*, which is discussed in in the Device Calibration section. $atan2(x, y)$ is the arc-tangent function that can calculate the angle in all four quadrants. The output of $atan2(x, y)$ ranges from -180° to 180°. Because the steering wheel angle may overflow this range, we find all possible angles by adding multiples of 360° to the θ_{static}, and use the one closest to the last estimation as the estimated steering wheel angle θ_a.

4.1.2 Vehicle Motion Removal

The above steering wheel angle estimation is obtained when the vehicle is static. To have a better understanding of how the motions of vehicles affect θ_a, we revisit Figure 3(a) for driving scenarios. When the vehicle is moving, the inertial sensors in the IMU and smartphone capture other accelerations (e.g., accelerations caused by turning, accelerating, and braking) in addition to the gravity force. Thus the accelerations projected to the X_r and Z_r axes can be derived as:

$$\begin{aligned} a_{X_r} &= sin(\theta_a) * a_{Z_s} + cos(\theta_a) * a_{X_s}, \\ a_{Z_r} &= cos(\theta_a) * a_{Z_s} - sin(\theta_a) * a_{X_s}, \end{aligned} \tag{2}$$

where a_{X_s} and a_{Z_s} are the accelerations aligned to the x- and z-axes of the *steering wheel coordinate system*. By combining two equations, we further derive the steering wheel angle in moving scenarios as below:

$$\begin{aligned} tan(\theta_a) &= \frac{tan(\theta_{static}) - a_{X_s}/a_{Z_s}}{1 + a_{X_s}/a_{Z_s} * tan(\theta_{static})}, \\ &= tan(\theta_{static} + atan(a_{X_s}/a_{Z_s})), \\ \Rightarrow \theta_a &= \theta_{static} - atan(a_{X_s}/a_{Z_s}), \\ &= \theta_{static} - \theta_{error}. \end{aligned} \tag{3}$$

We note that the estimated steering wheel angle in Equation 3 can be considered as the angle estimated in the static scenario (i.e., θ_{static}) calibrated by removing an angular error (i.e., θ_{error}), which is determined by the accelerations in the *steering wheel coordinate system* and independent of the poses of the steering wheel and IMU.

In order to obtain a_{X_s} and a_{Z_s} in Equation 3, we examine the relationship between the *steering wheel coordinate system* and the *vehicle coordinate system* as illustrated in Figure 3(b). We find that a_{X_s} is the same as the vehicle's acceleration on the x-axis of its own coordinate system (a_{X_v}), which can be easily obtained by aligning the measurements of the smartphone's accelerometer to the *vehicle coordinate system* in the Device Calibration. In addition, we find that the steering wheel always has a pitch angle (ϕ_{pitch}) to the y-axis of the vehicle coordinate system. Therefore, a_{Z_s} is the combination of the gravity acceleration and the vehicle's acceleration projecting to the z-axis of the *steering wheel coordinate system* as shown below:

$$a_{Z_s} = g * sin(\phi_{pitch}) - a_{Y_v} * sin(\phi_{pitch}). \tag{4}$$

Finally, the complementary filter is still used to estimate the steering wheel angle by fusing the estimated steering wheel angle based on accelerations (θ_a) and angular velocity derived from IMU's y-axis gyroscope reading after aligned to the *rotated steering wheel coordinate system*.

4.1.3 Angle Bias Removal

The constant rotation of steering wheel makes error existing in the coordinate alignment during Device Calibration, resulting in a bias in the estimated steering wheel angle. To remove this angle bias, our system calibrates itself by examining the average estimated steering wheel angle when the vehicle is driving straight. The intuition is that the ideally estimated steering wheel angle should be zero degrees when driving straight, therefore any non-zero average estimated steering wheel angle found when driving straight is the angle bias that we should remove.

In particular, our system keeps collecting the angular velocity on the z-axis of the *vehicle coordinate system* from the gyroscope of the smartphone ($gyro_{Z_v}$) and that on the y-axis of the *steering wheel coordinate system* from the gyroscope of the IMU ($gyro_{Y_s}$) since the start of each trip. If the angular velocities from both sources are less than a threshold (e.g., $0.01 rad/s$ and $0.1 rad/s$ for $gyro_{Z_v}$ and $gyro_{Y_s}$, respectively), the system considers that the vehicle is driving straight and pushes the estimated steering wheel angle into a sample pool. The average of the samples in the pool is considered as the angle bias to be removed from all the steering

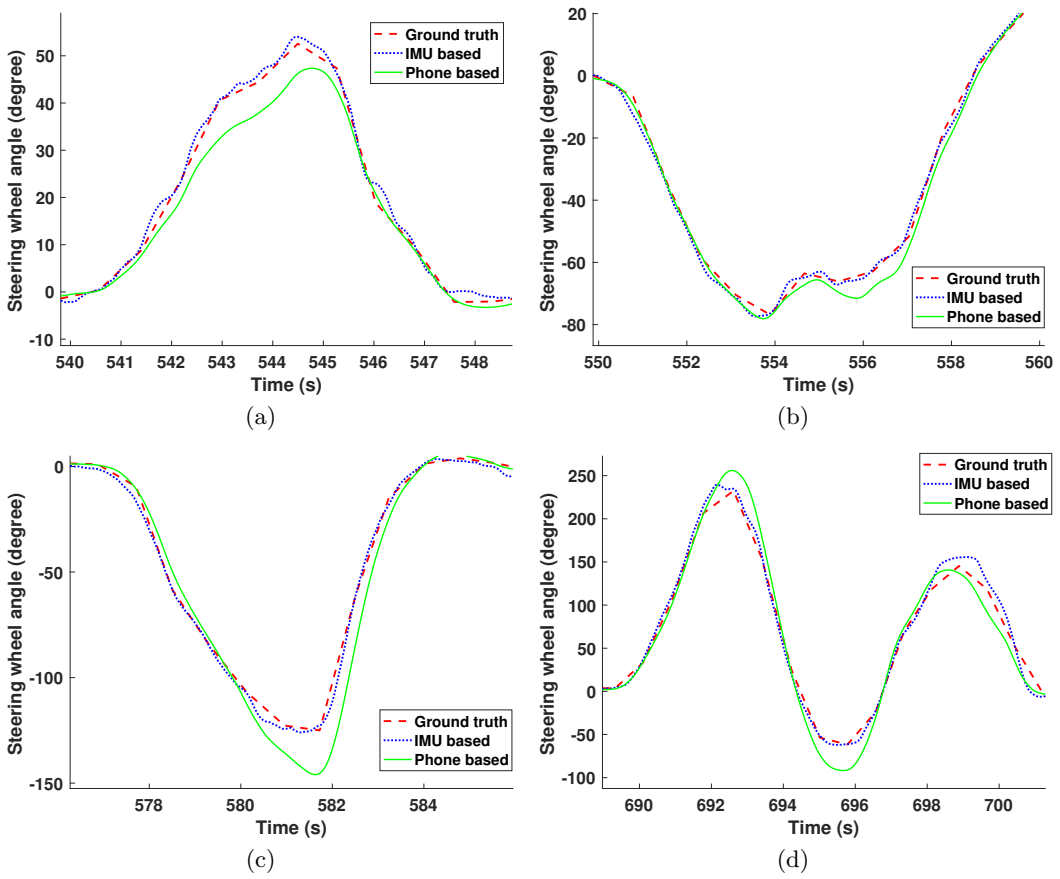


Figure 4: Compare the steering wheel angle estimation results of steering-wheel based and phone based approach.

wheel angles estimated in the rest of the trip. The size of the sample pool depends on the time needed to collect enough samples that can confidently determine the angle bias in every trip. In our experiments, we find that our system can successfully find and remove this angle bias within the first one minute of typical daily commute trips.

4.1.4 *Phone Based Steering Wheel Angle Estimation*

An alternative approach in BigRoad is not to fix the sensor on the steering wheel. Instead, we can just employ the available smartphone inside the vehicle to accomplish the steering wheel angle estimation task. This method can be adapted to any inertial sensors, but we use the smartphone sensor here to minimize the infrastructure needs. Instead of explaining bunches of complex mathematical equations, we simplify the model as shown in Figure 5, in which we assume the turning angle of the two front tires is the same. For common vehicles, the steering wheel angle $\theta_{steering}$ is equal to the front tire's turning angle θ_{tire} multiply the vehicle's steering ratio k. And if the turning radius r and wheelbase l of the testing vehicle are known, θ_{tire} can be calculated in real time:

$$\theta_{steering} = k * \theta_{tire} = k * asin\frac{l}{r}, \qquad r = \frac{v}{\omega} = \frac{v}{gyroz_v}. \tag{5}$$

The vehicle turning radius r can be obtained as shown in Equation 5 based on the physics phenomenon inherited from

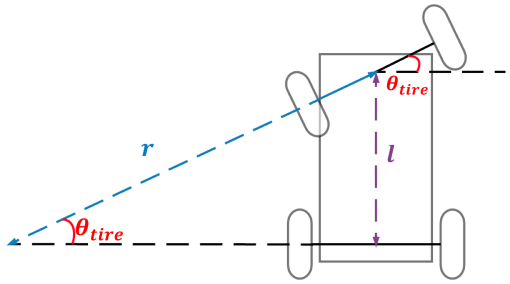


Figure 5: Simplified Ackerman mechanism based steering angle calculation.

inertial sensor readings. Here v stands for the vehicle speed, which we can get from the vehicle speed estimation module presented in section 4.2. ω represents the angular velocity of the vehicle, which can be replaced by the gyroscope reading on the z-axis of the *vehicle's coordinate system*($gyroz_v$). Thus, we can calculate the vehicle turning radius, and then get θ_{tire} leveraging only a smartphone. However, the steering ratios(k) of most commercial vehicles are proprietary. To obtain the steering ratio of our testing vehicle, we apply linear regression on the steering wheel angle ground truth and estimated tire angle from part of our collected dataset, and use the ratio to evaluate the remaining part of the dataset.

To compare these two steering wheel angle estimation ap-

proaches, the phone based approach utilize less infrastructures, but it needs more vehicle information such as steering ratio and wheelbase. The steering-wheel IMU based approach can estimate steering wheel angle of any vehicle model without extra information by exploiting the additional IMU sensor on the steering wheel. Figure 4 demonstrates the estimation result of two approaches and compare them with ground truth. We will show that the performance of steering-wheel IMU sensor based approach is better than the phone based approach in section 7.

4.2 Driving Acceleration and Speed Estimation

In addition to steering wheel angles, driving accelerations and speed are also critical inputs for self-driving research and thus part of the system's outputs. The driving acceleration is the smartphone's acceleration projected to the driving direction when the smartphone is fixed in the vehicle for video recording. Therefore the acceleration can be obtained after the smartphone's measurements are aligned to the vehicle's coordinate system. An intuitive way to obtain the driving speed is exploiting the location service in smartphones' operating systems. However, we found the speed information to deviate on average by more than 1 km/h from the internal vehicle speed. To improve the accuracy, we propose to (i) shift the speed measurements from the system with an average delay and (ii) apply an acceleration-based complementary filter to compensate for large speed changes.

4.2.1 *Speed Delay Shifting*

The speed information provided by smartphones' operating systems are derived by fusing the GPS, WiFi and cellular network information. Compared with the speedometer reading of vehicles logged on the same smartphone from the OBD port, the smartphone speed readings showed a delay. We observed this on a Nexus 5 with Android 5.1 as well as a Nexus 6 and a Nexus 6P with Android 6.1. This delay may be due to specific processing and filtering methods employed in the GPS and Android positioning system. Since this delay appears reasonably similar for the same smartphones across different trips, the system corrects for this delay by adjusting the GPS speed log time stamps. Figure 6 illustrates the time delay between OBD readings and the GPS readings from a Nexus 5 with Android 5.1, and its consistency in a single trip. Based on this observation, we determine the time delay τ by minimizing the average absolute errors between the speed measurements from the OBD and smartphone's GPS through a segment of data of m samples as shown in the following equation:

$$\underset{\tau}{\operatorname{argmin}} \sum_{i=1}^{m} \frac{1}{m} |V_i^{obd} - V_{i+\tau}^{gps}|, \tag{6}$$

where V_i^{obd} and $V_{i+\tau}^{gps}$ are i^{th} speed measurement from OBD and GPS, respectively. The time delay τ will then be applied to shift all speed measurements from the smartphone's GPS.

4.2.2 *Acceleration based Complementary Filter*

The speed reading shifting is able to largely reduce the error from delay, but the speed obtained from a smartphone is still less accurate during rapid acceleration or deceleration, because the update rate of the GPS receiver is as low as one update per second. To capture rapid changes in speed, we fuse inertial acceleration measurements with the speed

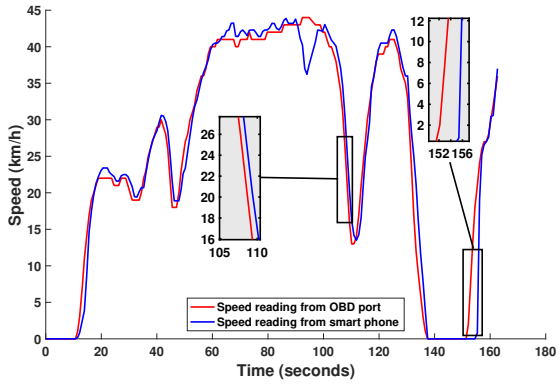


Figure 6: Speed readings from OBD-II and smartphone.

obtained from the GPS. In particular, we estimate the driving speed by using a complementary filter as shown in the following equation:

$$V_t^E = \alpha * (V_{t-1}^E + \Delta V_t) + (1 - \alpha) * V_t^P, \tag{7}$$

where V_t^E and V_{t-1}^E denote the speed of the vehicle at time t and $t-1$, ΔV_t is the speed change between t to $t-1$ derived from the accumulated accelerations of the vehicle, V_t^P is the speed from the GPS, and α is a variable controlling the balance between two speed sources. The acceleration of the vehicle is available from the smartphone's accelerometer measurements after the coordinate alignment procedure described in section 6. Some advanced GPS chipsets [17] integrated accelerometers for tracking speed, while our method is able to improve the accuracy of speed logged by commodity smartphones after an automatic device calibration, which are only equipped with standalone GPS module.

5. EXTERNAL PERCEPTION RECORDING

In addition to driver inputs, the external environment should be acquired in order to provide a perception for self-driving vehicle studies. We focus on cameras and provide a synchronization mechanism in the next subsection since cameras have become the key low-cost sensor for monitoring the external environment. Beyond the visual perception, better knowledge of road surface conditions can also be useful. In section 5.2, we introduce methods to distinguish wet and dry road conditions .

5.1 Time-stamped Video Capture

The most intuitive external perception provided by BigRoad is the real-time front-view video feed from smartphones' rear cameras that capture everything happening on the road in front of the vehicle. The desire of such directly visual information is tremendous, as it has been exploited in many vehicle applications, including self-driving, traffic crowdsourcing, etc. Most of these applications need to synchronize the video frames to sensor data at millisecond granularity for training or evaluation. To this end, BigRoad can record video frames with a list of time stamps for each frame using the smartphone's system current time at millisecond granularity. The fine-grained time information provided by BigRoad can facilitate many vehicle applications. For instance, the steering wheel angle output can be interpolated

to match the time stamp of each frame, and used to train automated steering systems.

5.2 Road Condition Estimation

Next, we introduce how BigRoad provides the estimation of road conditions (i.e., dry or wet) based on sensors integrated in smartphones.

5.2.1 Intuition

In advanced driving assistance and automated driving systems, the awareness of road conditions is one of the critical parameters for adjusting speed and turning angles to ensure the safety of passengers. Most existing studies identify factors that affect the road's friction coefficient by using vehicle-mounted specialized sensors (e.g., optical fibers [18], stereo camera [19], 24-GHz automotive radar [20] and voice recorder [21, 22]), which require either extra cost and installation effort or visibility/lighting environments. Our approach is thus challenging as we do not use any additional sensors (e.g., camera, automotive radar) to directly sense the road's condition. In contrast, we analyze the statistics of braking events captured by phones' inertial sensors through crowdsourcing and estimate road conditions, which is low-cost and easy to deploy.

In particular, we develop a proof-of-concept estimation framework to determine binary road conditions (i.e., dry or wet) by examining inertial sensors measurements in vehicles' brake activities from BigRoad. The intuition is two-fold: First, different road conditions have different maximum friction coefficients, which are usually defined by the following equation [23]:

$$\mu_{max} = max(|\frac{F_x}{F_z}|), \tag{8}$$

where F_x is the longitudinal traction force and F_z is the normal force acting on the tire. By defining the normalized traction force (NTF) of the vehicle as $\rho = \frac{F_x}{F_z}$, the μ_{max} can be represented by the $max(\rho)$, which can be obtained by a smartphone's inertial sensors when the vehicle is experiencing a hard brake. Because dry roads always have much larger μ_{max} than wet roads have, the $max(\rho)$ of a vehicle in dry road conditions should be larger than those in wet road conditions.

Second, in order to avoid skids on wet roads, most drivers would brake earlier and more gently [24] than they do on dry roads, resulting in relatively smaller ρ in normal braking activities. Since drivers usually perform multiple braking activities in each trip due to the stop signs, traffic lights, traffics, and etc, it is possible to crowdsource NTFs in braking activities from a large number of vehicles and drivers to accurately estimate road conditions. We note that video camera in BigRoad is another sensor that we can use to estimate road conditions by using image processing techniques, but it highly depends on the visibility and lighting environments. Therefore, we focus on using the inertial sensors based approach in this paper.

5.2.2 Per-brake NTF Derivation

In order to derive the NTF ρ during each brake, we first detect and segment the braking events using the acceleration readings collected by the smartphone in BigRoad. Specifically, we examine the standard deviation of accelerations on three axes to determine the vehicle's stop period using a threshold-based approach. We then extract six-second measurements before the starting point of the stop period as the segment data in a braking event.

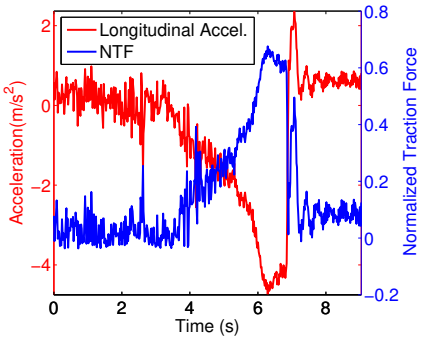
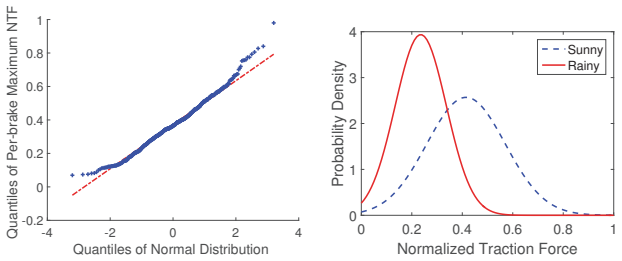


Figure 7: Example of longitudinal acceleration and normalized traction force during a brake.



(a) QQ-plot (vs. normal distribution). (b) Probability density functions.

Figure 8: QQ-plot and probability density functions of maximum per-brake NTFs.

Next, we calculate the NTF in each brake event using the longitudinal acceleration, which can be obtained by the Device Calibration (Section 6). To simply the problem, we consider a bicycle-type vehicle model [23], where the difference between left and right tires is ignored. If we ignore the effects caused by winds and road gradient, the longitudinal force during a brake can be obtained by:

$$F_x = m|a_x| - |F_r|, \tag{9}$$

where m is the total mass of the vehicle, a_x is the longitudinal acceleration and F_r is rolling resistance force which is usually between $0.015mg$ and $0.02mg$.

In addition, the normal forces on the front and rear tires can be calculated as:

$$F_{zf} = \frac{mgL_r - ma_xh}{L}; F_{zr} = \frac{mgL_f + ma_xh}{L}, \tag{10}$$

where L_f and L_r are the distances from the center of gravity to the front and rear axles respectively. h is the distance from the center of gravity to the road surface and L is the wheel base (i.e., $L = L_f + L_r$). Note that we can use either F_{zf} or F_{rf} to calculate the vehicle's NTF ρ depending on the form of the vehicle's drivetrain (i.e., front-wheel drive or rear-wheel drive). Finally, the NTF can be derived using the vehicle's longitudinal acceleration during each brake and a few of the vehicle's basic information. Figure 7 shows an example of the longitudinal acceleration and its corresponding NTF during a brake.

5.2.3 *Crowd-sourcing-based Road Condition Estimation*

The instant NTF derived from each brake can be exploited to estimate road-conditions by being compared to different road-condition models. Such models are nothing but the distributions of NTFs abstracted from a large number of NTFs crowd-sourced from vehicles driving in different road conditions. The crowdsourced data used for building general models should cover different braking types, driving behaviors, etc., so that the models can be resilient to these different situations. Additionally, we empirically observe that the maximum NTF in each brake (e.g., the peak value of the NTF in Figure 7) fits normal distributions. The Quantile-Quantile plot in Figure 8(a) compares over 700 maximum NTFs from 40 daily trips of 5 drivers with normal distribution and verifies this observation. We thus use the least-squares based approach to fit the maximum NTFs collected from brakes in sunny days (i.e., dry condition) and rainy days (i.e., wet condition) to two road-condition models, which are two probability density function following normal distributions, namely pdf_d and pdf_w. Figure 8(b) shows an example of pdf_d and pdf_w generated by a training dataset (i.e., 350 brakes in sunny days and 100 brakes in rainy days) collected by BigRoad from 5 users at different locations. The figure also verifies that drivers brake more gently in rainy days as we expected.

It is important to note that determining the road condition based on the NTF information of only a single braking action is difficult, because the two distributions shown in Figure 8(b) are largely overlapping with each other. BigRoad takes advantage of crowdsourcing techniques and collects a large number of braking events with different braking types and driving behaviors from various on-road vehicles in a specific driving area. The system eventually can respectively calculate the joint probability of all these brakes performed on dry or wet roads, and determine the road conditions accordingly.

Specifically, we can estimate the road condition by comparing the instant maximum NTF with the abstracted road condition models as shown in the following equations:

$$\begin{cases} \text{Road is dry, if} \prod_{i=1}^{N} pdf_d(\hat{\rho}_i) \geqslant \prod_{i=1}^{N} pdf_w(\hat{\rho}_i) \\ \text{Road is wet, if} \prod_{i=1}^{N} pdf_d(\hat{\rho}_i) < \prod_{i=1}^{N} pdf_w(\hat{\rho}_i), \end{cases} \quad (11)$$

where $\hat{\rho}_i$ is the maximum NTF of the i^{th} collected brake, N is the total number of brakes being used.

6. DEVICE CALIBRATION & SYSTEM IMPLEMENTATION

In order to ensure BigRoad work with most vehicle models, the system should be able to calibrate itself to different steering wheel positions, sensor placements, and vehicle models. The main challenge is that both the smartphone and IMU have their own coordinate systems, and the sensor measurements from both devices need to be aligned to a fixed coordinate system (e.g., the vehicle coordinate system) before they are useful.

We develop two modules to automatically align measurements from smartphone and IMU to the *vehicle coordinate system*(X_p, Y_p, Z_p) and *rotated steering wheel coordinate system*(X_r, Y_r, Z_r) when driving, namely *Smartphone Calibration* and *IMU Calibration*. The relationship between the two coordinate systems is shown in Figure 9.

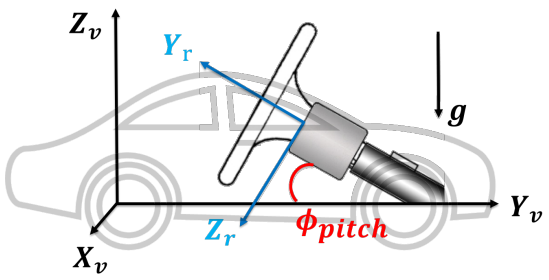


Figure 9: Vehicle coordinate system and rotated steering wheel coordinate system.

6.1 Smartphone Calibration

We implement the coordinate alignment algorithm to align sensor measurements from the smartphone to the *vehicle coordinate system* based on an existing work [25]. Specifically, it utilizes the smartphone's accelerometer and gyroscope measurements when the vehicle brakes while driving straight to derive the corresponding rotation matrix.

6.2 IMU Calibration

The coordinate alignment for IMU is challenging, because the IMU's orientation is subject to the rotation of the steering wheel, IMU's position, and vehicle models. The IMU Configuration adopts two steps to align the sensor measurements from the IMU to the *steering wheel coordinate system*: First, the system aligns the sensor measurements to the *vehicle coordinate system* when vehicle is driving straight using the same approach introduced in the Smartphone Calibration. Since we perform this alignment while driving straight, aligned IMU's coordinate system will depart the *vehicle coordinate system* when steering wheel turns, but its x-axis is always identical to the *rotated steering wheel coordinate system*. Second, the system calculates the steering wheel pitch angle (ϕ_{pitch}) and further rotate other two axes to the *rotated steering wheel coordinate system*.

As shown in Figure 9, ϕ_{pitch} is the angle between Y_v's negative direction and Y_r's positive direction, which is usually adjustable by the driver. And this angle also exists between negative direction y-axis of the aligned IMU and Y_r's positive direction. Thus, ϕ_{pitch} can be derived from the angular velocity projection on aligned IMU's coordinate system's y- and z-axis, as shown in equation 12:

$$\phi_{pitch} = atan(\frac{-gyro_{Y_i}}{gyro_{Z_i}}), \quad (12)$$

where $gyro_{Y_i}$ and $gyro_{Z_i}$ are the gyroscope reading's on IMU's y- and z- axes after aligned to *vehicle coordinate system*. The system automatically picks twenty samples from a steering motion and calculate ϕ_{pitch} based on Equation 12. Then, IMU's measurements can be rotated to the *rotated steering wheel coordinate system* by rotating around x-axis for $(180 - \phi_{pitch})^\circ$. Besides, BigRoad also includes a data synchronization module to remove the delay of IMU data caused by Bluetooth transmission. This module uses the same brake segment used in smartphone coordinate alignment and runs cross correlation on smartphone and IMU acceleration trace to find the delay.

6.3 Implementation

In summary, two components are required in BigRoad:

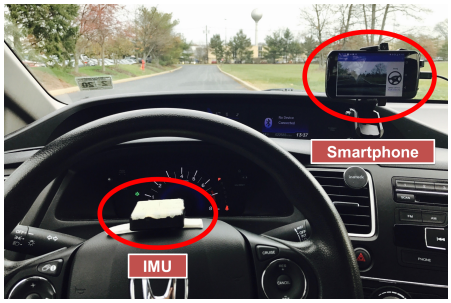


Figure 10: BigRoad setup.

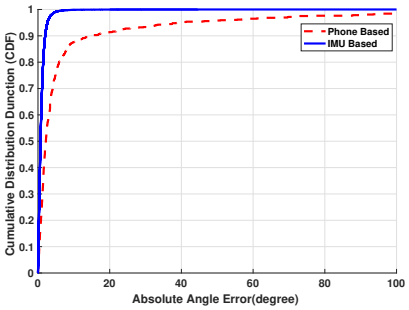


Figure 11: CDF of steering wheel angle absolute error.

an IMU and a smartphone, which are mounted on top of the steering wheel and a phone holder respectively as shown in Figure 10. Specifically, the smartphone is equipped with inertial sensors, a camera and a GPS receiver. The IMU contains inertial sensors and periodically sends sensor data to the smartphone via Bluetooth. In our implementation, we use the Invensense MPU-9150 9-axis motion sensor as the IMU. We implement the BigRoad as an app on Android 4.1, which is applicable to any major brand of Android smartphones. All the data collected by BigRoad will be synchronized to the smartphone's system clock, and could be automatically uploaded to the cloud server via WiFi or cellular network.

7. PERFORMANCE EVALUATION

In this section, we evaluate our system with respect to (i) the accuracy of internal driver input information it provides, (ii) the accuracy of external perception monitoring, and (iii) the usefulness of our collected data, namely whether our collected data could allow training of self-driving system components.

7.1 Experiment Setup

We conduct driving experiments with BigRoad to evaluate its performance. Our experiments include six vehicle models and 7 drivers driving on various types of roads for their daily commute in two states. In total, we collect 143 trips in a 3-month period. During these trips we used different experiment configurations to allow studying different questions as follows:

Internal Driver Inputs Accuracy. We collect 84 trips from three vehicles (i.e., 2015 Honda Civic, 2016 Chevrolet Impala, and 2016 Chevrolet Equinox) and five drivers to evaluate the accuracy of the collected driver inputs. We used these vehicles because we were able to gain access to ground truth steering wheel angle data reported by an internal sensor (for two of the vehicles a carmaker provided us with a specialized device and for one vehicle we were able to reverse engineer the data format of messages captured with a standard OBD-II/CAN interface). We also recorded driving speed from the internal vehicle bus, which is openly available as part of the OBD-II standard. We were able to sample the data from the Honda Civic, Chevrolet Impala and Equinox at 1.43Hz, 100Hz and 10Hz, respectively.

External Perception Accuracy. We evaluate the performance of road condition estimation based on 59 trips from four vehicles (i.e., 2008 Nissan Rogue, 2010 Honda Civic, 2015 Mazda CX-5 and 2015 Mercedes-Benz GLC 350) and five drivers. The ground truth of the road condition and weather for each trip was manually labeled by the drivers.

Data Usefulness. We use 5 trips from three different drivers driving the 2015 Honda Civic on highways to evaluate the usability of BigRoad's video output. The dataset includes about 1.5 hours of video at 30 frames per second and steering wheel angle ground truth from the OBD interface at 100Hz. We were able to achieve a higher sampling rate of 100Hz here, because we do not simultaneously capture vehicle speed information in these experiments.

7.2 Accuracy of Driver Input

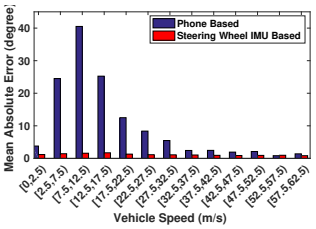
Steering Wheel Angle Estimation. We first evaluate the performance of our steering-wheel estimation approaches based on IMU and phone sensing to quantify the accuracy-complexity trade-off (how accuracy degrades when only phone sensors are used) and to understand whether the accuracy with phone sensors is still sufficient for self-driving data collection. In particular, we measure the absolute error[2] of the estimated steering wheel angle for both approaches by reporting the difference between the estimated value and the ground truth from the OBD readers. Figure 11 shows the CDFs of the steering wheel angle estimation errors of our two approaches.

We observe that the IMU-based approach performs similarly to the tolerances reported for a built-in steering wheel sensor. In particular, the mean error of the steering-wheel IMU based approach is 0.96° with the median of 0.69° and 90-percentile of 1.99°. This compares to 1.5° accuracy reported for an internal steering angle sensor [26].

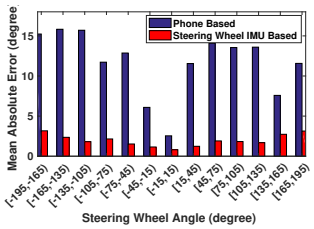
We also observe that the overall errors of the steering-wheel IMU based approach are much smaller than those of the phone based approach. The mean error of the phone based approach is 7.53°, with the median at 2.19° and the 90-percentile at 15.05°.

We further study the impact of the driving speed and steering wheel angle on the estimation accuracy. Figure 12(a) presents the error distribution of the IMU based and phone based steering-wheel estimation approaches with respect to 13 bins of driving speed between 0 to 60km/h. From the figure, we can tell that the steering-wheel IMU based approach has consistently low errors across all driving speeds. The phone based approach, however, shows larger errors in the lower speed range (i.e., between 5-30km/h). We believe this is because the error of the driving speed and angular velocity becomes more obvious in low speed, so the phone

[2]The errors used in the rest of the paper are all absolute error, unless stated otherwise.

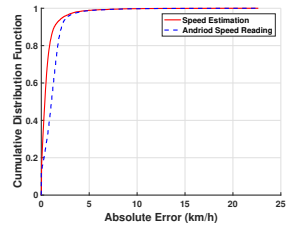


(a) Error distribution with respect to vehicle speed.

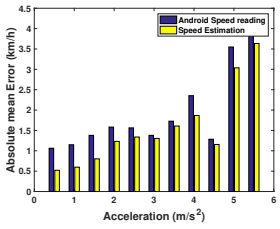


(b) Error distribution with respect to steering angle.

Figure 12: Error distribution with respect to speed and steering angle.



(a) CDF of speed estimation absolute error.



(b) Error distribution with respect to acceleration.

Figure 13: CDF of speed estimation absolute error.

based approach cannot accurately measure vehicle turning radius based on equation 5.. Figure 12(b) presents the error distribution of the two approaches with respect to 13 bins of steering wheel angle ranging from $-180°$ to $180°$. We find that the estimation error of the phone based approach increases significantly for larger steering wheel angles, while the steering-wheel IMU based approach shows comparatively low errors across steering wheel angles. Note that larger steering wheel angles are normally correlated with lower vehicle speeds.

Driving Speed Estimation. Next, we evaluate the performance of the driving speed estimation component of BigRoad by respectively comparing the deviation of the estimated driving speed and the driving speed directly obtained from Android with the internal vehicle speed obtained from OBD-II in Figure 13(a). The smartphones we used in this experiment include a Nexus 5 with Android 5.1, and a Nexus 6 and a Nexus 6P with Android 6.1. From the figure we can observe that BigRoad can achieve much lower deviations than the smartphone-reported speed. In particular, the smartphone driving speeds from the Android location service have an average deviation of 1.17 km/h, while that of BigRoad is only 0.65 km/h. The 90-percentile deviation of the estimated speeds from Android and BigRoad are 2.11 km/h and 1.37 km/h, respectively, which indicate that the delay shifting and complementary filtering techniques in BigRoad can effectively reduce deviations.

To further explore the limitation of our speed estimation method, we plot the mean deviation of the estimated speed from Android and BigRoad with respect to the y-axis absolute acceleration of the vehicle in Figure 13(b). We find that the speed estimation errors of both approaches generally increase with the accelerations of the vehicle although BigRoad

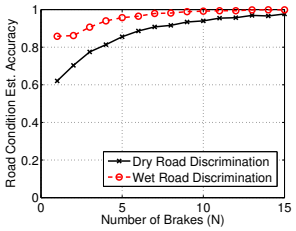


Figure 14: Discrimination accuracy of the road condition estimation.

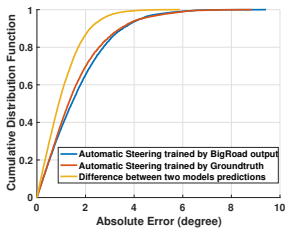


Figure 15: CDF of two trained Automatic Steering Algorithms and their comparison.

can achieve lower errors. We believe this is because both approaches rely on the location updates from GPS, which have a low refresh rate and cannot capture speed changes between two samples.

7.3 Accuracy of External Perception

As the main part of external perception, we mainly evaluate the performance of road condition estimation in this section. As discussed in Section 7.1, we distribute BigRoad systems to 5 users to collect driving data in their daily commutes. In total, we collected 737 braking events from 43 trips in sunny days and 193 braking events from 16 driving trips on rainy days. We use 50% of the braking data for training the road condition models (i.e., pdf_d and pdf_w) and use the rest of the data for the testing purpose.

Specifically, we can identify the road conditions by using the NTF statistics from N collected braking events according to equation 11. Figure 14 shows the road condition discrimination accuracy with a different number of events N. We observe that BigRoad can achieve high accuracy to determine the road's condition (i.e., dry or wet) by only using the normal braking events from crowdsourced driving data. And the system only needs a small number of braking events (e.g., 15) from the vehicles to achieve over 95% discrimination accuracy.

7.4 Usefulness of BigRoad Output

To explore the usefulness of data collected with the proposed BigData approach, in this work we implemented an automatic steering algorithm, which represents an important component of a self-driving system. The automatic steering algorithm is constructed through an end-to-end learning approach that forgoes feature extraction and trains a deep convolutional neural network to directly map raw pixels from a single front facing camera to steering commands, which has recently been demonstrated [11]. The neural network is implemented based on the open source code from [27] with 5 layers, including three convolutional layers and two fully connected layers. To feed the collected data into the deep neural network, we developed a few strategies including synchronizing each video frame with the recorded steering wheel angle and speed for training purposes and video rescaling and cropping. In particular, a Nvidia Quadro K5000 is used to train the network, and the trained network takes $3ms$ to process each input frame. The trained neural network shall have the capability to output the intended steering angle solely based on the front camera view.

We demonstrate the usefulness of BigRoad's outputs by

comparing the results of our automatic steering application when using two different pairs of driving data: the front-view video together with either (i) the estimated steering wheel angles from BigRoad or (ii) the ground truth from the OBD port. We use about 1.3 hours of real-road driving data to train the deep neural network in the application, and use the remaining 0.2 hours of the data for testing. The comparison of the results is shown in Figure 7.4, in which we regard the OBD port steering angle of human drivers as the ground truth. We observe that the median errors of the application are 1.30° and 1.42° for using the steering wheel angles provided by the OBD and BigRoad, respectively. Moreover, we plot the CDF of the sample-to-sample difference between the prediction results from the two scenarios in Figure 7.4. We observe that the median difference between using these two inputs is 0.90°. These small differences suggest that this self-steering application can effectively be trained by using the data from BigRoad.

Furthermore, we find that the state-of-the-art end to end self-steering work based on CNNs [11] does not directly use the driver's steering angle obtained from steering sensors as a training label because the driver's input may not be perfect. It applies a computer vision based calibration to slightly correct the driver's steering angles. We note that such a calibration could also be applied to BigRoad outputs before using them to train the self-steering network, which presumably results in even smaller performance differences.

8. RELATED WORK

There has been extensive work on vehicular sensing and self-driving that BigRoad can contribute to [11, 13, 28, 29, 30, 31, 32, 33, 34]. Machine learning methods are primarily chosen due to the complexity the problem. Pomerleau's pioneering work [33] estimates the vehicle's steering direction from the camera images and a laser rangefinder. [29, 30, 31] incorporate high-resolution video and laser data to track trajectory, obstacle, terrain roughness, etc. Some other studies used only a camera to solve self-driving vehicle problems. [34] and [11] developed an adaptive cruise control and steering systems, respectively. Brubaker et.al. [28] use visual odometry and road maps for a real-time self-localization system. [13] and [32] extract traffic light states from the camera to help route planning and increase energy efficiency. Developing these systems need an enormous amount of data which can be facilitated by BigRoad with minimum infrastructure.

Most existing efforts to collect driving data build on a small fleet of tens of highly instrumented vehicles that are continuously operated with test drivers [5, 6, 7, 8]. Traditional vehicle manufacturers [6, 7] are testing their autonomous driving technologies on redesigned cars. Google's parent company, Waymo [5], has accumulated 2 millions miles of testing drives with their self-driving cars. Comma.ai's project OpenPilot [8] reverse engineering the OBD-II data from two vehicle models of Hondas and Acuras, and uses the collected data to train CNN based self-driving systems. However, limited by the vehicle numbers and models, it is challenging to accumulate a sufficiently large dataset with these approaches.

Smartphones are also used to monitor the vehicle and the driver [25, 35, 36, 37, 38]. Wang et.al. [25] utilize smartphone sensors and OBD device to capture vehicle dynamics. Bo et.al. [35] detect texting and driving using smartphones. Dai et.al. [36] and Johnson et.al. [37] propose driving behavior monitoring systems to track other risky driving events. Chen et.al. [39] introduce a middleware to sense a vehicle's steering using smartphone sensors. However, this work only limits to high-level steering motion, such as lane change. Other approaches [40, 41] track steering wheel usage and angle leveraging smart watches. Different from these studies, which use only the gyroscope measurements, BigRoad fuses the accelerometer and gyroscope readings from both the IMU and smartphone to achieve the fine-grained steering wheel angle estimation. Besides, these studies suffer from the hand-over-hand problem and have relatively low accuracy.

Another body of work focuses on crowdsourcing data for driving applications. CrowdITS and Waze [42, 43] utilized crowd's feedback to gather the traffic conditions. Chen et. al. [44] proposed a semi-automated approach to tag parking spots from speed and driver feedback. SmartRoad [45] locates traffic signals and stop signs from vehicle speed. Finally, Li et. al. [14] predict traffic light state based on observed wait times.

9. CONCLUSION

In this paper, we seek a low-cost yet reliable solution to collect fine-grained driving data that can support developing dependable automated driving and traffic safety technologies. We present BigRoad, a light-weight sensing and driving data logging system that can derive internal driver inputs (i.e., steering wheel angle, driving speed and acceleration) and external perceptions of road environments (i.e., road conditions and front-view video) using very few off-the-shelf sensing devices (i.e., a smartphone and an IMU), which are not dependent on vehicle types. The system uses advanced coordinate alignment techniques to enable driving data acquisition independent of sensor orientation in heterogeneous driving scenarios across different vehicle models and drivers. We develop estimation algorithms based on complementary filtering to derive accurate steering wheel angle, driving speed and acceleration. By crowdsourcing real driving data, we model binary road conditions (i.e., dry and wet) based on the distributions of vehicle's accelerations and use them for real-time road condition estimation. In addition, BigRoad captures front-view videos with millisecond granularity time stamps to facilitate various vehicle applications and research that need to synchronize video frames with high-sampling-rate sensing data. Over 140 trips from six different vehicles and seven drivers show that the system can generate steering wheel angle with 0.69°median error, driving speed with 0.65km/h deviation, and determine binary road conditions with 95% accuracy. We further develop train an automatic steering angle predictor to demonstrate that data from such light-weight sensors can be used without significant performance degradation. We hope that these techniques will allow larger-scale driving data collection and facilitate the development of advanced driver assistance and automated driving technologies.

Acknowledgments

We sincerely thank our shepherd Dr. Souvik Sen and the reviewers for their insightful comments. This material is based in part upon work supported by the National Science Foundation under Grant Nos. CNS-1329939, CNS-1409767, CNS-1409811, and CNS-1566455.

10. REFERENCES

[1] Tom Krisher and Joan Lowy. Tesla driver killed in crash while using car's 'autopilot'. *ASSOCIATED PRESS(June 30, 2016)*.

[2] Chris Urmson. Google Self-Driving Car Project. https://www.transportation.gov/sites/dot.gov/files/docs/AV%20policy%20guidance%20PDF.pdf. Talk at SXSW Interactive 2016.

[3] National Highway Traffic Safety Administration. Federal Automated Vehicles Policy. https://www.transportation.gov/sites/dot.gov/files/docs/AV%20policy%20guidance%20PDF.pdf.

[4] National Highway Traffic Safety Administration. Fatality Analysis Reporting System. https://www.transportation.gov/sites/dot.gov/files/docs/AV%20policy%20guidance%20PDF.pdf. Talk at SXSW Interactive 2016.

[5] Waymo: Google's parent company on self-driving. https://waymo.com/.

[6] Kirsten Korosec. GM's Cruise Automation Is Testing Self-Driving Chevy Bolts in Arizona. http://fortune.com/2016/08/09/cruise-automation-arizona-gm/.

[7] Mercedes-Benz Research & Development, Autonomous Driving. http://mbrdna.com/divisions/autonomous-driving/.

[8] Openpilot. https://github.com/commaai/openpilot.

[9] UMTRI Naturalistic Driving Data. http://www.umtri.umich.edu/our-focus/naturalistic-driving-data.

[10] VTTI Naturalistic Driving Study. http://www.vtti.vt.edu/impact/index.html.

[11] Mariusz Bojarski, Davide Del Testa, Daniel Dworakowski, Bernhard Firner, Beat Flepp, Prasoon Goyal, Lawrence D Jackel, Mathew Monfort, Urs Muller, Jiakai Zhang, et al. End to end learning for self-driving cars. *arXiv preprint arXiv:1604.07316*, 2016.

[12] The OpenXC platform. http://openxcplatform.com/.

[13] Emmanouil Koukoumidis, Li-Shiuan Peh, and Margaret Rose Martonosi. Signalguru: leveraging mobile phones for collaborative traffic signal schedule advisory. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*, pages 127–140. ACM, 2011.

[14] Hongyu Li, Luyang Liu, Cagdas Karatas, Jian Liu, Marco Gruteser, Yingying Chen, Yan Wang, Richard P Martin, and Jie Yang. Towards safer texting while driving through stop time prediction. In *Proceedings of the First ACM International Workshop on Smart, Autonomous, and Connected Vehicular Systems and Services*, pages 14–21. ACM, 2016.

[15] Thomas Maze, Manish Agarwai, and Garrett Burchett. Whether weather matters to traffic demand, traffic safety, and traffic operations and flow. *Transportation research record: Journal of the transportation research board*, (1948):170–176, 2006.

[16] Hyung Gi Min and Eun Tae Jeung. Complementary filter design for angle estimation using mems accelerometer and gyroscope. *Department of Control and Instrumentation, Changwon National University, Changwon, Korea*, pages 641–773, 2015.

[17] LS20126 GPS Receiver. https://www.sparkfun.com/products/retired/9838.

[18] Johan Casselgren, Mikael Sjödahl, and James LeBlanc. Angular spectral response from covered asphalt. *Applied optics*, 46(20):4277–4288, 2007.

[19] Youngmin Kim, Namcheol Baik, and Jonghun Kim. A study on development of mobile road surface condition detection system utilizing probe car. *Journal of Emerging Trends in Computing and Information Sciences*, 4(10), 2013.

[20] Ville V Viikari, Timo Varpula, and Mikko Kantanen. Road-condition recognition using 24-ghz automotive radar. *IEEE transactions on intelligent transportation systems*, 10(4):639–648, 2009.

[21] D Gailius and S Jačenas. Iec detection on a road by analyzing tire to road friction ultrasonic noise. *Ultragarsas*, 62(2):17–20, 2007.

[22] Yunha Kim, Sehoon Oh, and Yoichi Hori. Road condition estimation using acoustic method for electric vehicles. In *The 10th FISITA Student Congress*, pages 30–6, 2010.

[23] Rajesh Rajamani. *Vehicle dynamics and control*. Springer Science & Business Media, 2011.

[24] Lin Zhang and P Prevedouros. Motorist perceptions on the impact of rainy conditions on driver behavior and accident risk. In *Proceedings of the 84th Annual Meeting of the Transportation Research Board, Washington, DC*, 2005.

[25] Yan Wang, Jie Yang, Hongbo Liu, Yingying Chen, Marco Gruteser, and Richard P Martin. Sensing vehicle dynamics for determining driver phone use. In *Proceeding of the 11th annual international conference on Mobile systems, applications, and services*, pages 41–54. ACM, 2013.

[26] Methode Electronic Steering Wheel Angle Sensor. http://www.methode.com/Documents/TechnicalLibrary/Steering_Angle_Sensor_Data_Sheet.pdf.

[27] Eder Santana and George Hotz. Learning a driving simulator. *arXiv preprint arXiv:1608.01230*, 2016.

[28] Marcus A Brubaker, Andreas Geiger, and Raquel Urtasun. Lost! leveraging the crowd for probabilistic visual self-localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3057–3064, 2013.

[29] Hendrik Dahlkamp, Adrian Kaehler, David Stavens, Sebastian Thrun, and Gary R Bradski. Self-supervised monocular road detection in desert terrain. In *Robotics: science and systems*, volume 38. Philadelphia, 2006.

[30] David Held, Jesse Levinson, Sebastian Thrun, and Silvio Savarese. Combining 3d shape, color, and motion for robust anytime tracking. *Proceedings of the 2014 Robotics: Science and Systems, Berkeley, CA, USA*, 1216, 2014.

[31] Jesse Levinson, Jake Askeland, Jan Becker, Jennifer Dolson, David Held, Soeren Kammel, J Zico Kolter, Dirk Langer, Oliver Pink, Vaughan Pratt, et al. Towards fully autonomous driving: Systems and algorithms. In *Intelligent Vehicles Symposium (IV), 2011 IEEE*, pages 163–168. IEEE, 2011.

[32] Jesse Levinson, Jake Askeland, Jennifer Dolson, and

Sebastian Thrun. Traffic light mapping, localization, and state detection for autonomous vehicles. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 5784–5791. IEEE, 2011.

[33] Dean A Pomerleau. Alvinn: An autonomous land vehicle in a neural network. Technical report, DTIC Document, 1989.

[34] Gideon P Stein, Ofer Mano, and Amnon Shashua. Vision-based acc with a single camera: bounds on range and range rate accuracy. In *Intelligent vehicles symposium, 2003. Proceedings. IEEE*, pages 120–125. IEEE, 2003.

[35] Cheng Bo, Xuesi Jian, Xiang-Yang Li, Xufei Mao, Yu Wang, and Fan Li. You're driving and texting: detecting drivers using personal smart phones by leveraging inertial sensors. In *Proceedings of the 19th annual international conference on Mobile computing & networking*, pages 199–202. ACM, 2013.

[36] Jiangpeng Dai, Jin Teng, Xiaole Bai, Zhaohui Shen, and Dong Xuan. Mobile phone based drunk driving detection. In *PervasiveHealth*, 2010.

[37] Derick A Johnson and Mohan M Trivedi. Driving style recognition using a smartphone as a sensor platform. In *IEEE ITSC*, pages 1609–1615, 2011.

[38] Yan Wang, Yingying Jennifer Chen, Jie Yang, Marco Gruteser, Richard P Martin, Hongbo Liu, Luyang Liu, and Cagdas Karatas. Determining driver phone use by exploiting smartphone integrated sensors. *IEEE Transactions on Mobile Computing*, 15(8):1965–1981, 2016.

[39] Dongyao Chen, Kyong-Tak Cho, Sihui Han, Zhizhuo Jin, and Kang G Shin. Invisible sensing of vehicle steering with smartphones. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, pages 1–13. ACM, 2015.

[40] C. Karatas, L. Liu, H. Li, J. Liu, Y. Wang, S. Tan, J. Yang, Y. Chen, M. Gruteser, and R. Martin. Leveraging wearables for steering and driver tracking. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9, April 2016.

[41] Luyang Liu, Cagdas Karatas, Hongyu Li, Sheng Tan, Marco Gruteser, Jie Yang, Yingying Chen, and Richard P Martin. Toward detection of unsafe driving with wearables. In *Proceedings of the 2015 workshop on Wearable Systems and Applications*, pages 27–32. ACM, 2015.

[42] Kashif Ali, Dina Al-Yaseen, Ali Ejaz, Tayyab Javed, and Hossam S Hassanein. Crowdits: Crowdsourcing in intelligent transportation systems. In *2012 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 3307–3311. IEEE, 2012.

[43] WAZE. https://www.waze.com/.

[44] Xiao Chen, Elizeu Santos-Neto, and Matei Ripeanu. Crowd-based smart parking: A case study for mobile crowdsourcing. In *International Conference on Mobile Wireless Middleware, Operating Systems, and Applications*, pages 16–30. Springer, 2012.

[45] Shaohan Hu, Lu Su, Hengchang Liu, Hongyan Wang, and Tarek F Abdelzaher. Smartroad: a crowd-sourced traffic regulator detection and identification system. In *Information Processing in Sensor Networks (IPSN), 2013 ACM/IEEE International Conference on*, pages 331–332. IEEE, 2013.