Journal of Mobile Networks and Applications manuscript No.
(will be inserted by the editor)

WreckWatch: Automatic Traffic Accident Detection and Notification with Smartphones

Jules White, Chris Thompson, Hamilton Turner, Brian Dougherty, and Douglas C. Schmidt

Received: date / Accepted: date

Abstract Traffic accidents are one of the leading causes of fatalities in the US. An important indicator of survival rates after an accident is the time between the accident and when emergency medical personnel are dispatched to the scene. Eliminating the time between when an accident occurs and when first responders are dispatched to the scene decreases mortality rates by 6%. One approach to eliminating the delay between accident occurrence and first responder dispatch is to use in-vehicle automatic accident detection and notification systems, which sense when traffic accidents occur and immediately notify emergency personnel. These in-vehicle systems, however, are not available in all cars and are expensive to retrofit for older vehicles.

This paper describes how smartphones, such as the iPhone and Google Android platforms, can automatically detect traffic accidents using accelerometers and accoustic data, immediately notify a central emergency dispatch server after an accident, and provide situational awareness through photographs, GPS coordinates, VOIP communication channels, and accident data recording. This paper provides the following contributions to the study of detecting traffic accidents via smartphones: (1) we present a formal model for accident detection that combines sensors and context data, (2) we show how smartphone sensors, network connections, and web services can be used to provide situational awarenss to first responders, and (3) we provide empirical results demonstrating the efficacy of different approaches employed by smartphone accident detection systems to prevent false positives.

J. White and H. Turner
Dept. of Electrical and Computer Engineering
Virginia Tech
E-mail: {julesw,hturner0}@vt.edu

C. Thompson, B. Dougherty, and D.C. Schmidt
Dept. of Electrical Engineering and Computer Science
Vanderbilt University
E-mail: {cthompson,briand,schmidt}@dre.vanderbilt.edu

1 Introduction

Emerging trends and challenges. Car accidents are one of the leading causes of death [2] in the US, causing over 100 fatalities daily. In 2007 alone more than 43,100 deaths resulted from 10.6 million traffic accidents. For every 100 licensed teenagers between the ages of 16 and 19, there will be 21 traffic accidents, making car accidents the leading cause of death for that age group in the U.S. [25].

A number of technological and sociological improvements have helped reduce traffic fatalities during the past decade, *e.g.*, each 1% increase in seatbelt usage is estimated to save 136 lives [9]. Advanced life saving measures, such as electronic stability control, also show significant promise for reducing injuries, *e.g.*, crash analysis studies have shown that approximately 34% of fatal traffic accidents could have been prevented with the use of electronic stability control [21]. Moreover, each minute that an injured crash victim does not receive emergency medical care can make a large difference in their survival rate, *e.g.*, analysis shows that reducing accident response time by one minute correlates to a six percent difference in the number of lives saved [12].

An effective approach for reducing traffic fatalities, therefore, is to reduce the time between when an accident occurs and when first responders, such as medical personnel, are dispatched to the scene of the accident. Automatic collision notification systems use sensors embedded in a car to determine when an accident has occurred [26, 7]. These systems immediately dispatch emergency medical personnel to serious accidents. Eliminating the time between accident occurrence and first responder dispatch reduces fatalities by 6% [26].

Conventional vehicular sensor systems for accident detection, such as BMW's Automatic Crash Notification System or GM's OnStar, notify emergency responders immediately by utilizing built-in cellular radios and detect car accidents with in-vehicle sensors, such as accelerometers and airbag deployment monitors. Figure 1 shows how traditional accident detection systems operate. Sensors attached

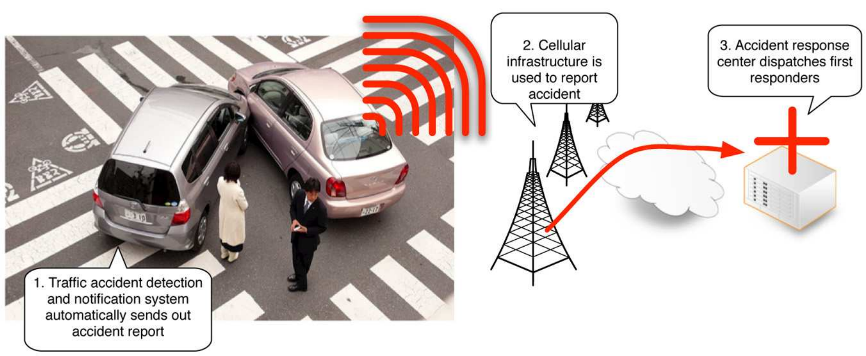


Fig. 1: A Vehicle-based Accident Detection and Notification System

to the vehicle use a built-in cellular radio to communicate with a monitoring center that is responsible for dispatching emergency responders in the event of an emergency.

Unfortunately, most cars in the US do not have automatic accident detection and notification systems. Only in 2007 did automatic notification systems become standard options in GM vehicles and most other non-luxury manufacturers do not include these systems as a standard option. Based on 2007 traffic accident data, automatic traffic accident detection and notification systems could have saved 2,460 lives (*i.e.*, 6% of 41,000 fatalities) had they been in universal use. A key impediment to using these systems is that they are infeasible or prohibitively expensive to install in existing vehicles and add to the initial cost of new vehicles. Moreover, these systems can be rendered obsolete, as evidenced by GM removing 500,000 subscribers from the OnStar service because they were equipped with analog (rather than digital) communications systems, and were therefore incompatible with their newer communication infrastructure.

Solution approach ⇒ Traffic accident detection and notification with smartphones. To address the lack of accident detection and notification systems in many vehicles, smartphones can be used to detect and report traffic accidents when accident detection and notification systems are unavailable. Smartphones, such as the iPhone and Google Android, have become common and their usage is rapidly increasing. In the 2nd quarter of 2010 alone, 325.6 million smartphones were sold [27]. This large and growing base of smartphone users presents a significant opportunity to extend the reach of automatic accident reporting systems. Moreover, smartphones are widely used by the teenage demographic, which is historically the most accident prone driver age group. The number of teenagers using mobile phones has been increasing steadily, from 45% of teens in 2004 to 63% in 2006 and then 71% in 2008 [20].

The low cost of smartphones compared to other traffic analysis and accident prediction systems makes them an appealing alternative to in-vehicle accident detection and reporting systems [23]. Moreover, smartphones travel with their owners, providing accident detection regardless of whether or not the vehicle is equipped with an accident detection and notification system. Furthermore, because each smartphone is associated with its owner, automatic notification systems built on smartphones can aid in the identification of victims and determining what electronic medical records to obtain before victims arrive at the hospital.

The ability to detect traffic accidents using smartphones has only recently become possible because of the advances in the processing power and sensors deployed on smartphones. For example, the iPhone 4 includes a GPS system for determining the geographic position of the phone, an accelerometer for measuring the forces applied to the phone, two separate microphones, and a 3-axis gyroscope for detecting phone orientation. Moreover, smartphones now possess significant sensor data processing power that can support the real-time execution of sensor data noise filtering and analysis algorithms. For example, the HTC Nexus One Android smartphone has a 1Ghz processor and 512MB of RAM.

Another key smartphone attribute for accident notification is that they provide a variety of network interfaces for relaying information back to centralized emergency response centers, such as 911 call centers. The iPhone 4 contains a cellular interface for sending and receiving data over GSM networks. Wifi can also be used by the iPhone 4 to send data to a nearby wireless access point. Smartphones

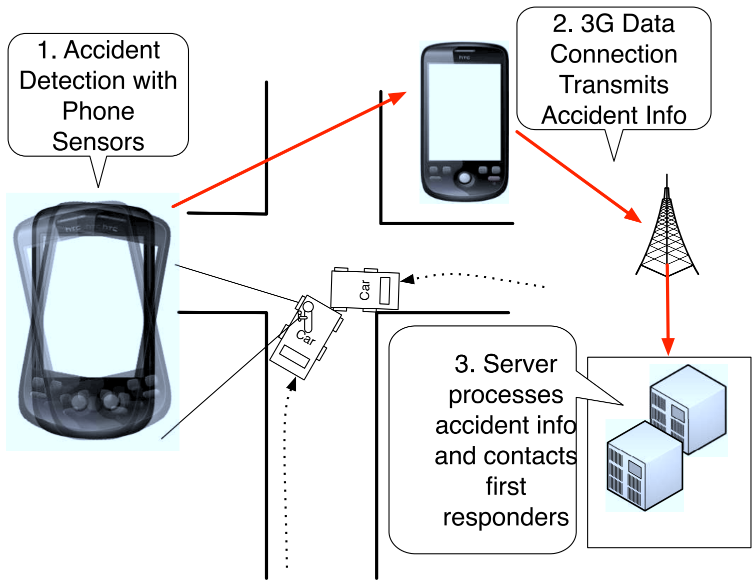


Fig. 2: Smartphone-Based Accident Detection System

also include Bluetooth wireless interfaces that can communicate directly with the onboard computers in many newer cars.

Smartphone-based accident detection applications have both advantages and disadvantages relative to conventional in-vehicle accident detection systems, *e.g.*, they are vehicle-independent, increasingly pervasive, and provide rich data for accident analysis, including pictures and videos. Building a smartphone-based accident detection system is hard, however, because phones can be dropped (and generate false positives) and the phone is not directly connected to the vehicle. In contrast, conventional in-vehicle accident detection systems rarely incur false positives because they rely on sensors, such as accelerometers and airbag sensors, that directly detect damage to the vehicle.

This paper shows how the sensors and processing capabilities of smartphones can be used to overcome the challenges of detecting traffic accidents without direct interaction with a vehicle's onboard sensors. We describe an approach for using smartphones to measure the forces experienced by a vehicle and its occupants to provide a portable "black box" data recorder, accident detection system, and automatic emergency notification mechanism. The approach detailed in this paper uses the sensors on a smartphone to record the G-forces (acceleration) experienced by the vehicle and occupant, the GPS location and speed of the vehicle, and the acoustic signatures, such as air bag deployments or impact noise, during an accident. Figure 2 shows how sensors built into modern smartphones can detect a major acceleration event indicative of an accident and then utilize the built-in 3G data connection to transmit that information to a central server to alert first responders. That server then processes the information and notifies the authorities as well as any emergency contacts.

This paper significantly extends our prior work [8] on traffic accident detection and notification using smartphones [8] in three ways. First, we present a formal model and algorithm for detecting accidents using smartphones. Second, we describe how acoustic data can be analyzed to lower false positives. Third, we include the results of experiments that quantify how acoustic data can help detect accidents and reduce false positives.

Paper organization. The remainder of this paper is organized as follows: Section 2 describes the challenges associated with using smartphones to detect traffic accidents; Section 3 describes techniques we developed to overcome these challenges; Section 4 empirically evaluates how to prevent false positives and accident reconstruction capabilities; Section 5 compares our work on smartphone-based accident detection systems with related work; and Section 6 presents concluding remarks.

2 Challenges Associated with Automatically Detecting Car Accidents

This section explores the challenges associated with detecting car accidents using a smartphone's sensor data. A task of critical importance in accident detection is ensuring that false positives are not reported to emergency services, such as 911. According to the US Department of Justice, 25 to 70 percent of calls to 911 in some areas were "phantom calls" where the caller immediately hangs up [29]. California receives approximately 6 million 911 calls from cell phones and between 1.6 and 3.6 million of these calls are phantoms [29]. Clearly, smartphone traffic accident algorithms must be careful not to increase the volume of phantom emergencies.

It is hard to strike a balance between no accident false positives and fully reporting all traffic accidents that occur. Vehicular accident detection systems, such as OnStar, have a significant advantage since they are integrated with the vehicle and its onboard air bag deployment and crash sensors. Sensor data received by these systems directly correlates to the forces experienced by the vehicle.

In contrast, smartphone accident detection systems must indirectly predict when an accident has occurred based on sensor inputs to the phone. Since phones are mobile objects, they may experience forces and sounds (indicative of a traffic accident) that originate from other sources, such as a user dropping the handset. Accident detection algorithms for smartphones must use sensor data filtering schemes that are resistant to noise, yet provide high enough fidelity to not filter out valid accidents.

2.1 Challenge 1: Detecting Accident Forces Without Electronic Control Unit Interaction

Conventional in-vehicle accident detection systems rely on sensor networks throughout the car and direct interaction with the vehicle's electronic control units (ECUs). These sensors detect acceleration/deceleration, airbag deployment, and vehicular rollover [3, 32]. Metrics from these sensors aid in generating a detailed accident profile, such as locating where the vehicle was struck, number of times it was hit, severity of the collision, and airbag deployment.

Smartphone-based accident detection applications must provide similar information. Without direct access to ECUs, however, it is harder to collect information about the vehicle. Although many cars have accident/event data recorders (ADRs/EDRs), it is unrealistic and undesirable to expect drivers to connect their smartphones to these ADRs/EDRs every time they get into the car. Not only would connecting to ADRs/-EDRs require require a standardized interface (physical and software) to ensure compatibility, but it would require exposing a safety-critical system to a variety of smartphone types and middleware platforms.

These conditions make it infeasible to verify and validate that each rapidly developed smartphone version integrate properly with every ADR/-EDR. Moreover, while many new cars have some form of ADR/EDR, any smartphone application that required interaction with an onboard computer would be useless in cars that lacked one. What is needed, therefore, is to collect the same or similar information utilizing only the sensors present on the smartphone alone. Section **??** explains how we address this challenge by using the sensors in the Android platform to detect accelerations/decelerations experienced by car occupants and Section 4 analyzes device sensor data captured by smartphones and shows that low false positive accident detection is possible.

2.2 Challenge 2: Providing Situational Awareness and Communication with Victims to First Responders

Situational awareness involves being informed of the environment of a specific area at an instant in time, comprehending the state of that environment, and being able to predict future outcomes in that space [11,6]. There are three levels of situational awareness: (1) perceiving emergency indicators in the environment, such as a driver seeing the collision of two vehicles in front of them, (2) comprehending the implications of those indicators, such as the driver realizing that they need to slow down, and (3) possessing an ability to predict what will transpire in the future, such as the driver determining that one of the cars involved in the accident will end up in the left lane [16].

After an accident, accident detection systems can provide critical situational awareness to first responders regarding the condition of the vehicle and occupants. This data can then be used by first responders to comprehend the physical state of the passengers and possibly predict how long they can survive without medical attention. For example, OnStar automatically places a voice call from the vehicle to an emergency dispatch service so that first responders can inquire about the condition of the vehicle's occupants, provide guidance, and predict whether or not an ambulance should be dispatched. These accident detection systems can also determine and report back to first responders information on air bag deployment, which indicates a serious accident. Moreover, accident detection systems, such as OnStar, can pinpoint the GPS coordinates of an accident and relay this information to first responders.

Effective smartphone accident detection systems must be able to replicate the complex situational awareness capabilities that are used by first responders. They must also provide indicators of the environment in a form that can be consumed by first responders. For example, the raw acceleration values of the phone are unlikely to help first responders understand what happened in an accident. Moreover, the system must provide sufficiently rich information to first responders to predict the future state of the driver and passengers, which is hard when the phone cannot directly measure their health or the car's condition. Section 3.5 describes how we use a combination of VOIP telephony, text messaging, mapping, and bystander reporting to provide situational awareness to first responders.

2.3 Challenge 3: Preventing False Positives

Vehicle-based accident detection systems monitor a network of sensors attached to the car to determine if an accident has occurred. One key indicator of a collision is an instance of high acceleration/deceleration due to a large change in velocity of the vehicle over a short period of time. These acceleration events are hard to attain if a vehicle is not actively being driven since it is unlikely that an unattended car will simply roll away from a parked location. Since smartphones are portable, however, it is possible that the phone may experience acceleration events that were not also experienced by the user. For instance, a phone may accidently drop from 6 feet in the air.

Since a smartphone-based accident detection application contacts emergency responders—and may dispatch police/rescue teams—it is essential to identify and suppress false positives. Due to smartphone mobility it is hard to differentiate programmatically between an actual car accident versus a dropped purse or a fall on a hard surface. The inability to identify and ignore false positives accurately, however, can render smartphone-based accident detection applications useless by wasting emergency responder resources on incident reports that were not real accidents. Section **??** explains how we address this challenge by using device usage context (such as speed) to filter out potential false positives and Section 4.1 provides empirical results evaluating our ability to suppress false positives.

3 Solution Approach

This section describes a prototype smartphone-based client/server application we developed—called "WreckWatch"—to address the challenges presenting in Section 2. WreckWatch provides functionality similar to an accident/event data recorder by recording the path, speed, and forces of acceleration on a vehicle leading up to and during an accident [5]. It can also notify emergency responders of accidents, aggregate images and video uploaded by bystanders at the scene of an accident, and send prerecorded text and/or audio messages to emergency contacts.

3.1 The WreckWatch Client/Server Architecture

WreckWatch is separated into two main components—the WreckWatch server and the WreckWatch client—shown in Figure 3. The WreckWatch client was developed using Google Android. It acts as a mobile sensor, relays accident information to the server via standard HTTP `post` operations, and provides an interface that allows third-party observers to contribute accident report data.

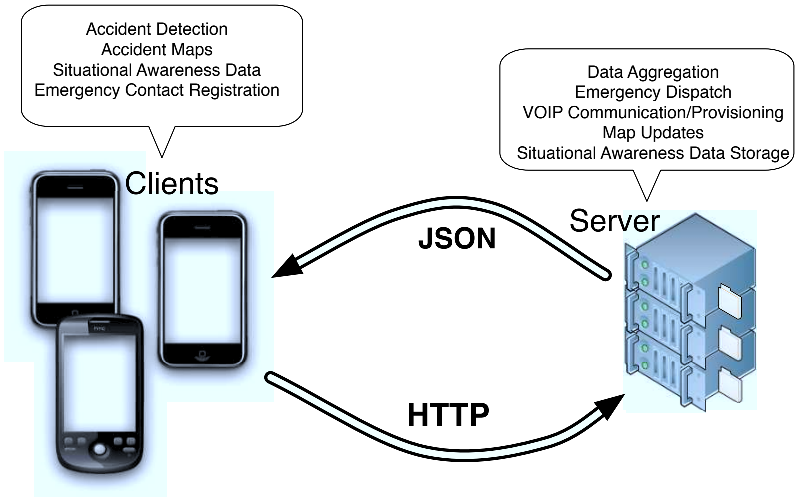


Fig. 3: WreckWatch Architecture Diagram

The WreckWatch Android client is written in Java based on Android 1.5 with Google APIs. It consists of several Android application *Activities*[1] for mapping, testing, and image upload. Background services detect accidents by polling smartphone system sensors, such as the GPS receiver and accelerometers. The polling rate is configurable at compile-time to meet user needs and to provide the appropriate power consumption characteristics. The WreckWatch client can gather data from phone databases (such as an address book) to designate emergency contacts. Communication to the server from the Android client uses standard HTTP `post` operations.

The WreckWatch server was developed using Java/MySQL with Jetty and the Spring Framework. It provides data aggregation and a communication conduit to emergency responders, family, and friends. It also allows clients to submit accident characteristics (such as acceleration, route, and speed) and presents several interfaces, such as a Google Map and XML/JSON web services, for accessing this information.

As accident information becomes available, the WreckWatch server posts location, route and severity information to a Google Map to aid emergency responders, as well as other drivers attempting to navigate the roads near the accident. This map is available over HTTP through a standard web browser and is built with AJAX and HTML, as shown in Figure 4. The remainder of this section presents the formal accident detection model used by WreckWatch and its approach to reducing false positives and then discusses features of the WreckWatch client/server application that supports first responder situational awareness.

[1] Activities are basic building block components for Android applications and can be thought of as a “screen” or “view” that provide a single, focused thing a user can do.

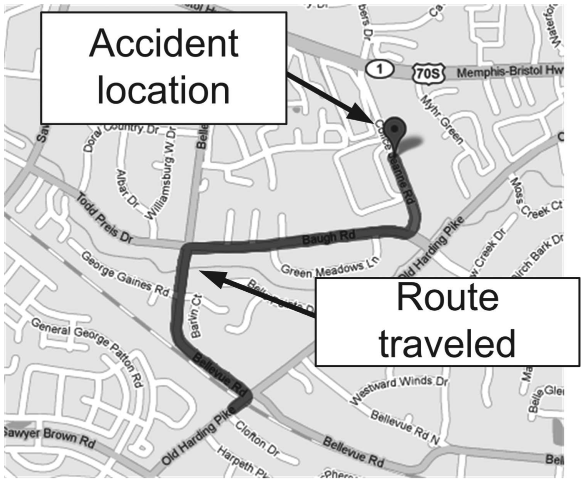


Fig. 4: WreckWatch Accident Map

3.2 The WreckWatch Formal Accident Detection Model

A carefully crafted formal model of accident detection is important to detect traffic accidents accurately. Challenge 1 from Section 2.1 described the problems associated with detecting traffic accidents without direct measurement of impact data from onboard sensors. Challenge 2 from Section 2.3 examined the potential for false positives, which is a key concern with applications that automatically dispatch police or rescue. To address both challenges, WreckWatch uses a soft real-time multi-sensor sampling approach, with threshold-based filtering to predict when an accident occurs. The formal accident prediction framework is based on the following 11-tuple model of the phone state, which is used to extrapolate the state of the vehicle:

$$\gamma =< \phi, T_\phi, \rho, T_\rho, \beta, \epsilon, S_\phi, S_\rho, S_\beta, M_\phi, M_\rho, M_\beta, M_\epsilon > \tag{1}$$

where:

- S_ϕ is the span of time after an acceleration event sets a value for the variable ϕ before the variable is reset.
- ϕ is an acceleration variable that indicates the maximum acceleration experienced in any direction by the phone. The maximum acceleration value is reset after S_ϕ milliseconds have elapsed.
- S_ρ is the span of time after a sound event with a sound pressure level greater than M_ρdBs that the sound event variable, ρ, will remain set to 1.
- ρ is a binary sound event variable that indicates if a sound event greater than M_ρdBs has occurred. The variable has value 1 if a sound event of M_ρdBs or more was experienced by the phone and 0 otherwise. From experimentation and a literature review on air bag deployment [30], we have found that 140dBs is a good value for M_ρ.

- S_β is the span of time after the phone is no longer traveling at least $M_\beta$mph that the speed threshold variable, β, will remain set to 1.
- β is a speed threshold variable with value 1 if the phone has been traveling at greater than $M_\beta$mph.
- ϵ is the distance traveled since the last time the variable β switched from value 1 to 0.
- M_ϕ is the minimum acceleration in Gs required for an acceleration event alone to trigger accident detection.
- M_ρ is the minimum decibels required for an acoustic event to trigger the sound event variable.
- M_β is the minimum speed in miles per hour that the device must be traveling in order to activate the accident detection system when it is inactive.
- M_ϵ is the max distance in feet that the device is permitted to move at a speed lower than the speed threshold, M_β, before the accident detection system is deactivated.

The WreckWatch accident detection algorithm operates on the 11-tuple γ. The accident detection function

$$Ev : \gamma \rightarrow \{0, 1\} \tag{2}$$

evaluates to 1 if an accident is detected and 0 otherwise. An accident detection can be triggered by one of two situations: (1) a high acceleration event and a high decibel sound event are recorded while the vehicle is moving above the threshold speed, M_β or (2) the distance moved since the last time the speed threshold M_β was exceeded is less than M_ϵ feet and an acceleration and sound event occur. More formally, we define these two accident detection conditions as:

$$Ev(\gamma) = \begin{cases} 1 & \text{if } (\frac{\phi}{M_\phi} + \alpha\rho \geq M_{Tr}) \bigwedge (\beta == 1) \qquad (a) \\ 1 & \text{if } (\epsilon < M_\epsilon) \bigwedge (\frac{\phi}{M_\phi} + \alpha\rho \geq M_{Tr}) \qquad (b) \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where:

- α is a adjustable weighting factor applied to the sound event that denotes its importance in the accident detection model. Higher values for α allow collisions at low speed or where the safety systems significantly dampen the impact, which can be detected through a combination of sound and acceleration.
- M_{Tr} is the threshold for accident detection.

The first accident detection scenario is triggered when the smartphone is traveling above a threshold speed associated with being inside a car. In this situation, an accident is detected if the smartphone experiences a violent acceleration event, indicating a probable collision, followed by a high-decibel acoustic event, such as air bag deployment, a horn, or an impact noise. It is also possible to detect an accident solely from an acceleration event, without a sound event, where the acceleration value alone is so large that it exceeds the accident detection threshold

$$\frac{\phi}{M_\phi} \geq M_{Tr}$$

.

The second scenario for accident detection occurs when the smartphone is traveling inside of a vehicle that stops at an intersection, traffic light, or other location. In this scenario, the algorithm attempts to detect if the user has exited the car or is merely waiting for a light or traffic condition to change. The accident detection algorithm uses the M_ϵ distance threshold to keep the detection process active below the threshold speed. As long as the smartphone does not travel more than M_ϵ feet from the last location the speed threshold was exceeded, the detection algorithm assumes that the user is still inside the car. This extra condition allows the algorithm to detect accidents that occur when the user's car is struck by another vehicle while stopped.

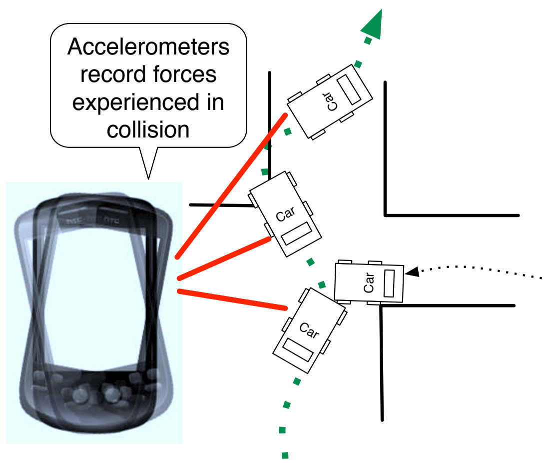


Fig. 5: Device Sensors Provide Acceleration Information

3.3 Using Acceleration Events to Detect Collisions

The accident detection model, γ relies on sampling the accelerometer to detect collisions, as shown in Figure 5. Given a stream of values from the accelerometer, denoted As, where each value As_i is recorded at time T_{As_i}, As_{now} is the most current value, and T_{now} is the current instant in time:

$$\phi = \begin{cases} As_{now} & if \quad As_{now} \geq \phi \\ As_i & if \quad (T_{now} - T_{As_i} \leq S_\phi) \bigwedge (\forall As_j \in As, As_i \geq As_j) \end{cases} \tag{4}$$

The value for ϕ is set to the greatest acceleration event experienced in any direction over the time span S_ϕ. If the current acceleration value is greater than ϕ, then ϕ is updated to the most recent acceleration value.

3.4 Using Acoustic Events to Detect Accidents

Our prior work [8] on accident detection was based solely on acceleration. It was thus potentially susceptible to false positives at low speeds and thus required higher settings for M_ϕ (higher values for M_ϕ, reduce the probability that low speed collisions will be reported). In our accident detection model described in Section 3.2, we added acoustic data analysis to improve lower speed collision detection and reduce the probability of a false positive by listening for high decibel acoustic events, such as impact noise, car horns, and air bag deployment. For example, air bag deployment is accompanied by high-amplitude, short-duration noise that can exceed 170dB at peak amplitude [30].

The WreckWatch formal model for accident detection uses built-in microphones on a smartphone to detect high-decibel acoustic events indicative of an accident. Using a secondary sensor in conjunction with acceleration attempts to lower the probability of false positives. As discussed in Section 4.2, clipping of the audio above 150 decibels and other potential noises (such as shouting) make it hard to use sound alone to detect accidents. It is possible that this limitation could be overcome, but we chose to make acoustic events a secondary filter for accident detection that aids in reducing false positives.

The accident detection model γ relies on sampling the microphone to detect accident noise. Given a stream of sound event decibel values denoted Ks, where each value Ks_i is recorded at T_{Ks_i}, Ks_{now} is the most current value, and T_{now} is the current instant in time:

$$\rho = \begin{cases} 1 & if \quad Ks_{now} \geq M_\rho \\ 1 & if \quad \exists Ks_i \in Ks, (Ks_i \geq M_\rho) \bigwedge (T_{now} - T_{Ks_i} \leq S_\rho) \\ 0 & otherwise \end{cases} \tag{5}$$

During any time span of S_ρ milliseconds, if a decibel value exceeds the M_ρ threshold, then ρ is set to 1. Once ρ is set to 1, it will remain set as long as sound events of greater than M_ρ decibels are experienced every S_ρ milliseconds.

Our future work will investigate dynamically adjusing the weight, α, applied to the sound event during accident detection. For example, if the car radio is set to a high volume level, ρ may remain continually set to 1. In this scenario, high decibel sound is much less indicative of an accident and thus α should be set to a low value.

3.5 Providing Situational Awareness to First Responders

Challenge 3 from Section 2.3 described the importance of replicating the situational awareness capabilities of in-vehicle accident detection and reporting systems. WreckWatch uses a combination of imagery, voice communications, GPS localization, and javascript object notation (JSON) web services to relay situational data to first responders, as described below.

Citizen scientist imagery. In an emergency, WreckWatch allows bystanders and uninjured victims to serve as "citizen scientists" [10] and report critical situational data to first responders. In particular, it allows bystanders and uninjured

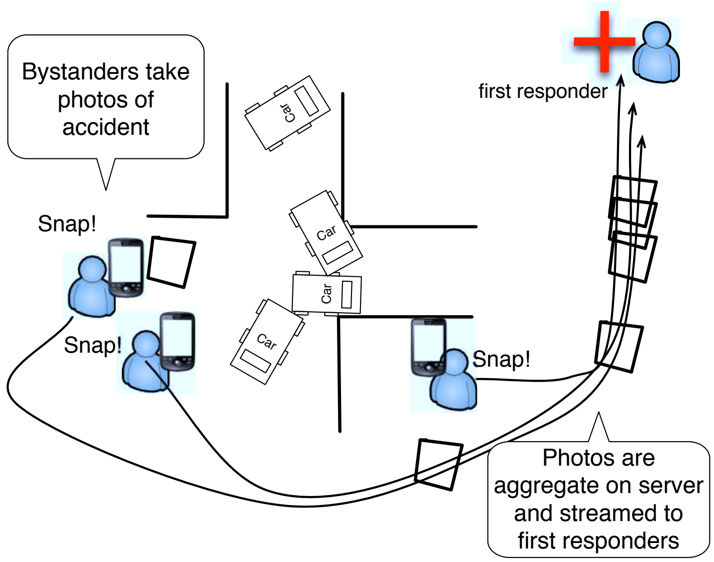


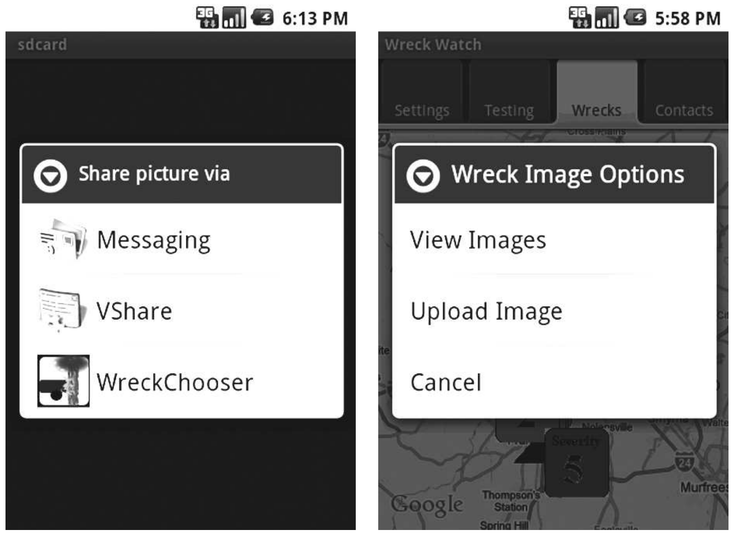
Fig. 6: Accident Image Upload

victims to take pictures using their smartphones and share them with first responders, as shown in Figure 6. Figures 7a and 7b show the client interface for uploading pictures of victim injuries or the accident scene to the WreckWatch server.

Emergency responders can access the uploaded images via mobile devices en route or a standard web browser at an emergency response center. The WreckWatch client provides mapping functionality through Google Maps on the device to ensure that emergency responders can continuously receive information about an accident to prepare them for whatever they encounter at the accident site. This map also allows other motorists to route themselves around an accident, thereby reducing congestion.

VOIP communication channels. The WreckWatch server uses digital portable branch exchange (PBX) functionality to make/receive phone calls and provision phone lines dynamically. It can therefore interact with emergency responders via traditional circuit-switched networks and create accident information hotlines in response to serious accidents via an Asterisk-based digital PBX running Linux. The server can also be configured with emergency contacts to notify via text and/or audio messages in the event of an accident. This data is configured at some time prior to a collision event so the server need not interact with the client to notify family or friends.

The PBX is built on Asterisk and connects to the server through a Java API. The Android client and web client pull information from the server and can be configured based on user needs. Due to the loose coupling and use of open standards between clients and server, additional clients for other platforms (such as other smartphones or desktop applications) can be implemented without the need



(a) Bystanders Can Upload Accident Images

(b) Client Application can View or Upload Accident Images

Fig. 7: Traffic Accident Imaging

to update the server. The WreckWatch server architecture also supports a heterogeneous group of clients, while providing appropriate qualities of service to each device.

JSON emergency web services. The WreckWatch server is a web-based service based entirely on freely-available APIs and open-source software. It is written in Java and built using Jetty atop the Spring Framework. It utilizes a MySQL database to store accident information and image meta-information. The server communicates with the clients via a RESTful architecture over HTTP using custom XML (for the Android application) and JSON (for the web-based application).

All communication between the clients and the server is initiated by clients. The server's operations (such as accident information upload) are performed by individual handlers that can be configured at runtime and are specified by parameters in an HTTP request. This architecture enables the addition of new operations and functionality without any software modifications or the need to recompile. All configuration is handled by an XML file parsed during server startup.

Geolocation and mapping of accidents. When an accident occurs, the WreckWatch client immediately reports certain accident characteristics to the server, including the GPS location of the wreck. Each accident is geo-tagged on the server with its location and entered into a searchable database of accidents. The accident locations are made available to first responders and other motorists through a Google Maps interface.

To further enhance first responders' understanding of the conditions leading up to the accident the route driven by the vehicle in the 30 seconds leading up to the crash is overlayed on top of the map. This route overlay allows first responders to determine the direction of travel and possible cause of the collision. This information allows the system to serve as a "black box" and possibly help to indicate areas where road improvement is needed.

3.6 Potential Advantages of Smartphone-based Accident Detection Systems

Our work with smartphone-based accident detection systems in the context of WreckWatch, we identified the following advantages relative to in-vehicle accident detection systems:

1. Smartphone sensors may measure forces closer to those experienced by victims. In the event of an accident, if the smartphone is in a user's pocket, the smartphone will experience close to the same forces and accelerations experienced by the occupants of the vehicle. Moreover, if the smartphone remains stationary relative to the vehicle during the collision, it is possible to use the data gathered from the smartphone to recreate and model the forces it experienced. In this case, the smartphone can provide data much like that gathered by vehicular ECUs.

Smartphones are often carried in a pocket [17] attached to a person. In these cases, the smartphone would experience the same forces as vehicle occupants, and could thus provide more information than in-vehicle systems by recording the forces experienced by occupants rather than just the vehicle itself. When this directionality and movement is combined with speed and location information from the GPS receiver, it is possible to help reconstruct the accident, including any secondary impacts.

2. The ubiquitousness of smartphones and their relatively low cost may help improve accident detection and notification system use. Many existing accident detection and traffic monitoring systems require an in-and-out of vehicle infrastructure to operate effectively. While some proposed accident detection systems utilize the existing cellular network, they have traditionally focused solely on voice capabilities and have not gained wide adoption. Smartphones allow use of the existing voice and data infrastructure, without the need for additional in-vehcile hardware. Due to customers and manufacturers not having to purchase new hardware, it is possible that the adoption rate of a smartphone-based accident detection systems would be higher than non-smartphone alternatives.

3. Reduced software maintenance complexity via smartphone application upgrade mechanisms. One inherent complexity in traffic monitoring and accident detection systems is the need to upgrade those systems to fix bugs and improve functionality over time. With thousands or millions of in-vehicle accident detection systems, maintenance can rapidly become a very expensive operation. An unfortunate reality is that frequently maintenance often becomes unduly expensive, resulting in the delay of many minor improvements until there is a major improvement that justifies the cost of bringing a vehicle into a service center for an upgrade. It may also be impossible to upgrade some legacy systems and continue servicing them, *e.g.*, OnStar dropped 500,000 of their subscribers due to outdated analog hardware.

Smartphones provide an effective solution for remote software maintenance through their built-in application store upgrade mechanisms, such as the iTunes Store. Moreover, smartphones tend to have a much higher refresh rate than cars, due to their lower costs and appeal as a status symbol. This trend towards constant turnover of hardware offers the potential to lower the average age of the hardware in use for accident detection.

4. Smartphone situational awareness systems can be augmented through cloud-based services. While onboard sensors are excellent for rapid accident detection, they are typically limited in terms of processing and notification capabilities. Since Smartphones are connected to a data network they can access cloud services to elastically extend their computational and/or storage capabilities. Moreover, new data analysis services can be plugged into servers without requiring complex upgrades of clients.

3.7 Potential Disadvantages of Smartphone-based Accident Detection Systems

While smartphones show significant advantages in the fields of accident detection and traffic monitoring, there potential disadvantages that motivate future research and refinement, as discussed below.

Accident detection systems consume a significant amount of battery power. GPS receivers consume a large amount of power and sampling them at the rate necessary to determine speed accurately reduces the battery life of the device to several hours. To overcome this limitation, users can plug smartphones into cigarette lights in vehicles to provide them with power. Requiring users to plug-in smartphones helps establish the context needed to eliminate false positives and also mitigates the power consumption of the GPS receiver.

Low speed traffic may trigger deactivation of WreckWatch. If a driver is stuck in low-speed traffic, their vehicle may travel beneath the M_β speed threshold for significant periods of time. Although WreckWatch uses the smartphone's GPS to determine device and (consequently) vehicle speed it only begins recording accelerometer information and looking for potential accidents above M_β speed threshold. In addition to reducing battery drain, this filter helps eliminate any acceleration events due to significant accidental smartphone drops that might occur outside a vehicle.

In high traffic congestion situations, however, filtering may shut off the accident detection system if the car travels more than M_ϵ feet at low speed, even though the user is still in the vehicle. Future work will explore filtering approaches that better distinguish beteween low-speed vehicle movement and walking. We intend to use the rythmic movement of walking to make this distinction.

Safety systems reduce impact forces. In-vehicle accelerometers are physically mounted to the chassis of the car, so their motion directly mirrors the vehicle and will experience most forces the vehicle experiences. Smartphones, however, are likely to be held in a pocket or holster. Car safety systems are designed to reduce the force on the occupants of the car during an accident and because of this, the forces experienced by the phone may be significantly less than the forces experienced by the accelerometers in the car.

These safety systems accomplish this reduction in force by increasing the time over which the change in velocity occurs. The net change in speed is the same, but the acceleration is less because it occurs over a longer period of time. Direct measurements report much higher accelerations, *e.g.*, the peak accelerations experienced inside a football helmet during play are approximately 29.2 G's [24]. For low-speed accidents there is the potential that the safety systems will reduce the acceleration on the phone below the M_ϕ G-force threshold needed for accident detection. Although low-speed crashes are less life-threatening, they still create a hazard to other motorists and should be reported. In future work, we are investigating other approaches to improve low-speed accident detection.

Destruction of the smartphone may prevent accident notification delivery. To maximize the probability that an accident is reported, it is critical to prioritize data transmission. WreckWatch uses a two-stage process to report accidents. First, the initial accident report is sent to the server using a small message that can be delivered over UDP or HTTP. Any additional information, such as forces of acceleration during the crash, is then transmitted immediately following the transmission of critical data. WreckWatch uses this two-stage protocol to increase the probability that the accident and crash diagnostic data is reported successfully. This two-stage protocol does not completely guarantee that a smartphone will be able to transmit crash data if it is destroyed. We are actively researching future approaches to improving notification success probabilities through the use of ruggedized external cradles for smartphones.

Smartphone OS development companies control the software capabilities of the sensor. For the forseeable future, a smartphone-based accident detection system would run as an application deployed on top of a smartphone operating system (OS). This approach implies that the software must operate within the architectural limitations of the platform. One example is the lack of multi-tasking on initial versions of the iPhone and on the new Windows Phone 7. A smartphone user would likely not be willing to run an accident detection application every time they enter their vehicle. Not only is this an issue for the initial development of such a system, but once the system is developed major changes in the OS application programming interface (API) would have the potential to cripple the entire system. This problem also follows from the current trend of rapid updates to smartphone OS APIs, *i.e.*, if a developed accident detection system was not updated with changes in the smartphone OS API it could become obsolete rapidly.

Production quality testing is hard. A key concern of a smartphone accident detection system is the need to avoid false positives. When this need is combined with the large degrees of freedom (*e.g.*, speed, noise conditions, location of device, etc.) in an accident it is hard to validate a developed smartphone based accident detection system empirically. For this work to reach production quality reliability, methods to test the operational effectiveness of accident detection systems must be created.

4 Empirical Results

This section describes results of tests performed on the WreckWatch application described in Section 3. These results empirically evaluate WreckWatch's ability to prevent false positives and gather information to reconstruct an accident accurately.

4.1 Experiment 1: Evaluating the Possibility of False Positive Acceleration Values

As described in Section 2.3, avoiding false positives is a key challenge when detecting car accidents with smartphones. Although WreckWatch only activates the accident detection system at speed, it is still possible that a driver or passenger could drop their smartphone while the vehicle is in motion. The first experiment was designed to determine if the acceleration component of WreckWatch's accident detection system would be triggered by the phone falling inside the vehicle or by emergency braking that did not result in a crash.

Hypothesis ⇒. Accidental falls or non-emergency braking would produce insufficient acceleration to trigger accident detection. We hypothesized that the acceleration experienced by a smartphone when dropped would be substantially less than a car accident. We believed it would be hard to produce 4Gs of force without dropping the phone from a substantial height (such as from a multi-story building) or from a moving vehicle (such as a car on the highway). We considered both situations to occur rarely enough that they did not warrant experimentation.

Experiment setup. Since WreckWatch's speed filtering only activates the accident detection system when the phone is in motion, our experiments were conducted inside a vehicle. To analyze the potential for false positives from acceleration changes, we conducted two experiments designed to simulate events that generate accelerations whose values could potentially be interpreted as car accidents.

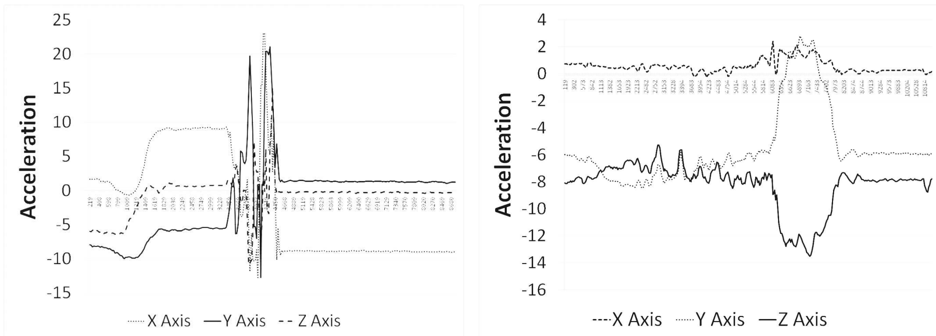
All experiments were performed on a Google ION device running the vendor image of Android 1.5 on a 525 Mhz processor with 288 MB of RAM. The device was factory reset before loading WreckWatch and no additional third-party applications were installed. WreckWatch recorded acceleration on three axes at the highest possible rate and wrote these values to a CSV file on the SD card in the device. This data was then downloaded to a Windows desktop computer for analysis in Excel.

In all graphs, positive z-axis values indicate positive acceleration in the direction from the battery cover toward the screen. Likewise, positive y-axis values indicate positive acceleration in the direction from the USB connector toward the smartphone speaker. Finally, positive x-axis values indicate positive acceleration from left to right when looking at the device with the USB connector closest the observer.

Empirical results. For the first test, the Android device was dropped from ear height in the driver's seat of a car. The device bounced off the seat and wedged between the seat and center console. Figure 8a shows the acceleration on each axis during the collision with the floor.

Using 9.8 m/s as an approximate value for Earth's gravity, the device experienced approximately 2G's in each direction with nearly 3G's on the x-axis before coming to rest. The required acceleration to trigger airbag deployment is 60G's [13, 1]. In addition to being ∼30 times smaller than required to deploy an airbag, this value is well below the 4G's used as a filter. It is therefore unlikely a smartphone could be dropped in a manner that would exceed 4G's. This data supports the use of a filter (presented in Section 3.2) to prevent false positives.

A sudden stop is Another potential scenario that could potentially generate a false positive. This test was performed in a vehicle by reaching a speed of approx-



(a) Acceleration During a Fall (b) Acceleration During a Sudden Stop

Fig. 8: Acceleration During Falls and Sudden Stops

imately 25 mph and engaging in a sudden stop. The test results are approximate as the exact speed was unknown and braking pressure was not exact. Figure 8b shows the acceleration experienced on each axis during the stop. As described in Section 3.6, because the smartphone remained stationary relative to the vehicle, it experienced the same forces as the vehicle. In this instance, the acceleration experienced by the smartphone was actually less than that experienced during the fall.

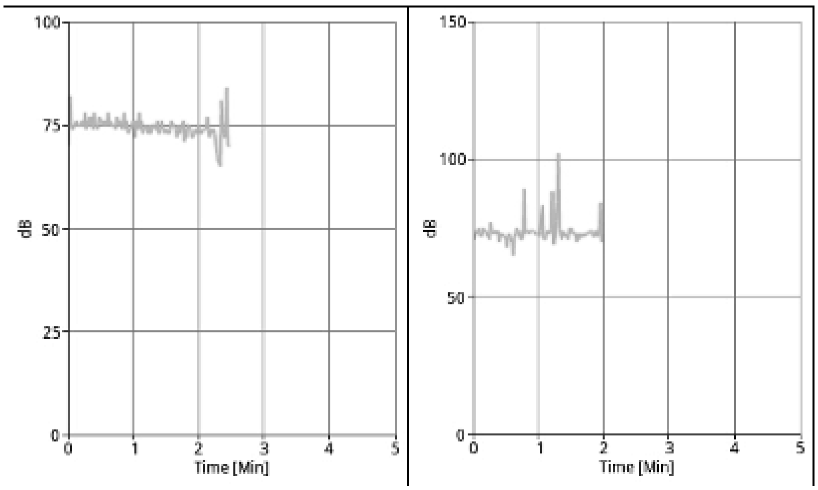
This result is attributed to the fact that although the stop was sudden and forceful, the car (and consequently the smartphone) came to a rest over a period of time that was longer than during the drop test. In other words, the change in velocity was greater but the actual acceleration was less because the change occurred over a longer period of time. Based on this data, it is unlikely for the smartphone to experience 4G's of acceleration simply due to a sudden stop.

4.2 Experiment 2: Evaluating the Possibility of Accoustic False Positives

Smartphone microphones can potentially augment the acelerometer of the phone to detect collisions. Drivers and passengers, however, often inadvertently create an array of loud noises that could potentially be interpreted by the device as the sound of an airbag deploying, leading to false incident reports. We therefore needed to determine whether benign noises associated with normal cell phone use could be mistaken for airbag deployment.

Hypothesis ⇒ Benign noisy acitivities, such as phone drops, shouting, laughing, loud music and driving with windows down would produce insufficient noise levels to trigger accident detection. We hypothesized that none of these noises would reach the 160dB range of an air bag deployment. If this was the case, it would be possible to tune the accident detection model to more heavily rely on accoustic signatures.

Experiment setup. To determine if vehicular or other sounds unrelated to those indicating a collision could trigger accident detection, we recorded the sound pressure in decibels (dB) of a number of potential road sounds that could generate



(a) Highway Noise (b) Dropped Phone

Fig. 9: Potential Noise Levels During Highway Transportation

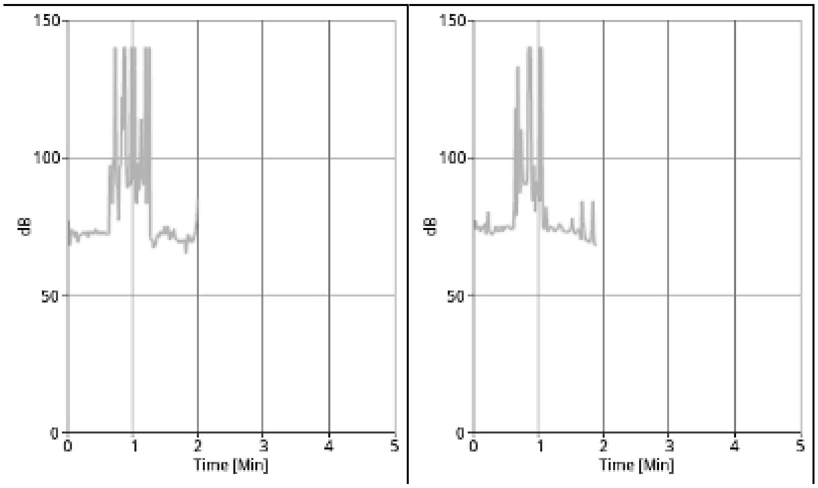
false positives. The decibel measurements for each sound were recorded directly by the phone rather than an external measurement device to directly measure the accoustic inputs that would be received by the accident detection algorithm. The road noises that we analyzed included:

1. Highway noise
2. The phone falling from ear height in a vehicle
3. Loud laughter
4. Shouting in an argument
5. Playing the radio at full volume and
6. Playing the radio at full volume with all windows down

Empirical results. The results of the experiment are shown in Figure 9, Figure 10 and Figure 11. The baseline readings were taken driving a 2006 four door Honda Accord at speeds of 55-70 mph on an interstate highway with the radio playing at 1/3 of maximum volume. As shown in Figure 9 the maximum decibel level reached for the baseline was 81 db.

Noise during transportation can dramatically increase, however, due to several incidents, such as phone drops, laughing, shouting, playing the radio loudly, and rolling down the windows. An effective solution must ensure that the device sound processing capabilities can differentiate between these benign activities and the noises associated with severe collisions, such as airbag deployment. Additional experiments were executed to simulate these events.

First, we recorded the decibel level associated with dropping the device multiple times form ear height. The results can be seen in Figure 9. Phone drops resulted in a maximum decibel level of 103 db, considerably less than the 160-180 db generated by an airbage deploying. We then measured the noise levels associated with two people laughing loudly and two people having a shouting argument. As shown in Figure 10, these activities resulted in a maximum noise level of 145 dBs.



(a) Shouting Argument (b) Loud Laughter

Fig. 10: Human Noise Levels During Highway Transportation

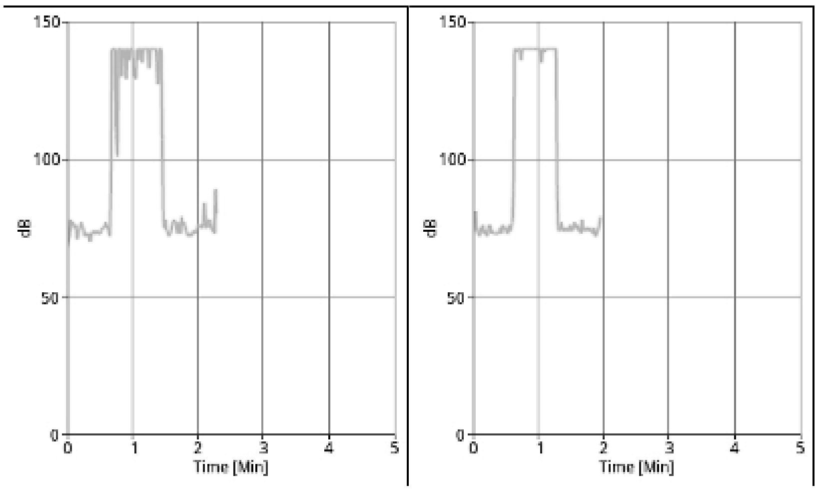
We finally measured the noise generated by playing the radio at maximum volume and driving with all windows down. These activities also generated noise levels of 145 dB as shown in Figure 11.

Based on these experiments, we determined that the ability for the device to detect sound pressure levels greater than 145dB is limited due to signal clipping. Using sound levels alone to determine if an accident has taken place could therefore potentially lead to false positives as a result of normal benign activities. We use this result to tune our accident detection model to rely on the accoustic signature as a secondary indicator of accidents and improve detection at acceleration values below our accelerometer threshold. For example, while the device reporting a noise level of 145 dB could be the result of a shouting match, a reading of 145 db *and* a reading of 3.5G's of force by the accelerometer would likely indicate that an accident occurred.

4.3 Experiment 3: Evaluating Accident Reconstruction Capabilities

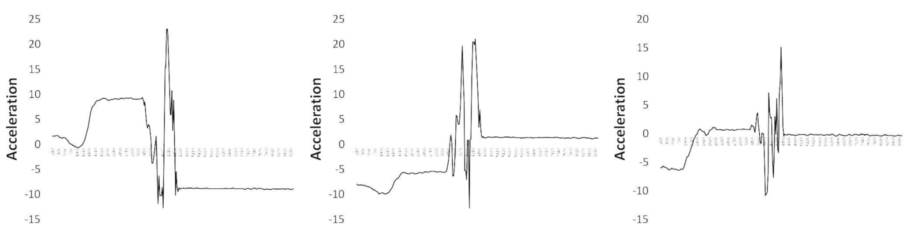
WreckWatch can potentially reconstruct an accident based solely on the data gathered from the smartphone. Due to the smartphone's presence in the vehicle during an accident, the smartphone will usually experience the same forces at the same time as the occupants and the vehicle itself. For example, ~40% of cell phones are carried in some form of pocket [17], in which case the device will likely experience the same forces experienced by the person wearing the pocket.

Hypothesis ⇒ The accelerometer value would provide sufficient information to reconstruct its movement during a crash. Due to the short time period in which a crash takes place, it is possible that a smartphone would have insufficient processing power and sensor sampling rates to capture enough data to accurately model the movement of the phone. We hypothesized that modern



(a) Radio Max Volume (b) Max Volume with Windows Down

Fig. 11: Stereo Noise Levels During Highway Transportation



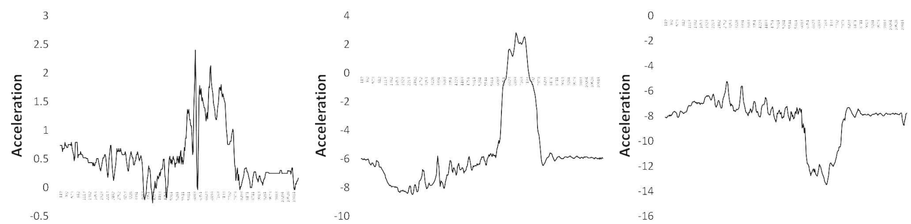
(a) X-Axis Acceleration (b) Y-Axis Acceleration (c) Z-Axis Acceleration

Fig. 12: Acceleration While Dropped in a Car

smartphones have sufficient processing power and sensor sampling rates to aid in accident reconstruction.

Experiment setup. To demonstrate this approach, we analyzed the data from the two experiments conducted in Section 4.1 to determine if we could reconstruct the orientation and movement of the smartphone.

Empirical results. The graph in Figure 12a shows it is possible to determine that the smartphone was initially experiencing zero acceleration along the x-axis indicating that the x-axis was perpendicular to the ground. This orientation is consistent with holding the smartphone to the ear. While falling, the smartphone tilted such the left edge of the smartphone (relative to the screen with the screen facing away from the ground) was the closest edge to the sky and then flipped again such that the left edge was closest to the ground. When Figures 12a, 12b, and 12c are combined it is clear that the bottom of the smartphone made contact first, followed by the left edge, and finally the back of the device.



(a) X-Axis Acceleration (b) Y-Axis Acceleration (c) Z-Axis Acceleration

Fig. 13: Acceleration During a Sudden Stop

The acceleration experienced during the sudden stop was actually less than that experienced during the fall. Given what is known about the event, it is therefore possible to identify the orientation of the smartphone during the event. By examining the graphs in Figure 13 it is possible to determine that the smartphone was resting at an angle such that the top of the smartphone was higher than the bottom of the smartphone. The decrease in acceleration along the z-axis is indicative of the force induced on the device by the seat as the car came to a rest. Graphs of other sudden stop events also have a similar appearance, as long as the device remained stationary relative to the car.

These reconstruction capabilities can help accident investigators identify what was experienced by the occupants of the vehicle and provide them with information that an ADR/EDR simply cannot provide. This information can also be combined with that present in the ADR/EDR to better understand the entire accident rather than simply the forces experienced by the vehicle itself. WreckWatch gives investigators the capability to analyze a real-world accident in a manner similar to the way they would a controlled collision involving crash-test dummies. Although WreckWatch cannot provide investigators with all impact information (*e.g.*, the forces experienced at the ribs [15] or the pressure on the face [22]), it can provide them with specific information about the overall force on the body and how effectively the restraints protected the passenger.

5 Related Work

This section compares WreckWatch with related work on accident and traffic detection systems. Our comparison with related work is organized as follows: (1) intelligent transportation systems, (2) traffic monitoring with cell phones, (3) mayday systems, and (4) traffic and road monitoring monitoring sensor networks.

Intelligent transportation systems. The US Federal Communications Commission (FCC) requires cell phones to provide emergency personnel with their location. This mandate increases their viability as an accident-detection system by ensuring that position data is not limited to advanced smartphones. Vehicle localization [35] and rapid data acquisition are important to an Intelligent Transportation System (ITS), which utilizes sensor networks to monitor traffic conditions and make adjustments to increase safety and reduce congestion on transportation networks [33]. These systems count cars to determine speed and congestion, as well as detect ice build-up and other hazards [19]. An ITS is not limited to highway traffic monitoring [4]. One major advantage of WreckWatch is that it could be utilized as a subsystem to an ITS.

Traffic monitoring with cell phones. Related work has used cell phones to construct a wireless mobile network for traffic-related applications. Traffic conditions are often measured via loop detectors that count vehicles and determine their speed. Since these loop detectors are typically embedded in the pavement there is a high cost associated with their installation and maintenance [18]. Moreover, loop detectors are often installed in main highways, limiting available information [28].

Cell phones have been tapped as a potential solution to both of these issues, because they provide a substantially larger amount of information, and because cell phone tracking could be available on most roads without installing specialized detection hardware. WreckWatch is a step towards showing that cell phones are an effective medium towards a wireless sensor network focused on automobile and traffic information.

The European National Institute for Transport and Safety Research conducted a study that used the volume of cell phones in range of a given cellular tower to identify potential areas of congestion or accidents [18]. This work is similar to WreckWatch in that it utilizes the cellular radios for the communication of information. WreckWatch is unique in that it utilizes the Android platform's sensor APIs to detect wrecks on a vehicle by vehicle basis, rather than using aggregate metrics. WrechWatch's execution directly on the smartphone allows it to access and utilize significantly more information about the device and user.

Mayday systems. Mayday systems provide voice connection to an emergency assistance while automatically providing user location. Additional items that mayday systems provide include remote door unlocking, remote engine diagnosis, theft detection and tracking, automatic route guidance, travel information, and various hands-free operations. Previous work [35] outlines the implications of location awareness on cellular devices, and the effect that this awareness would have on mayday systems.

The WreckWatch system could be extended to provide immediate voice capabilities via integration with the Asterisk digital PBX. Given WreckWatch's current integration with the Asterisk PBX, this extension is not technically hard to prototype. While remote diagnosis seems far-fetched, advances in automobile ECU interfaces will likely make this possible in the future. With the increase in wireless keys, remote door unlocking could be accomplished. If the phone has a wireless chip at the correct frequency then it can simply broadcast the door (or engine) key combination. If not, add-on smartphone sensor interfaces can be built to provide such capability. Route guidance, travel information, and hands-free operations could be easily added to the WreckWatch system by utilizing various Android APIs.

Other work [34] focuses on using the cellular features of OnStar together with accident detection functionality to investigate potential correlations between hands-free phone calls and car accidents. This work analyzed the proximity of calls to the OnStar system to an airbag deployment notification. WreckWatch could be extended to provide this information, and even more information by analyzing behavior (such as texting, voice calls, Internet browsing or even gaming) prior to an accident. Work to analyze the impact of distractions due to information systems (such as cell phones [31, 14]) has relied on imprecise analysis that could be improved through the use of a system like WreckWatch that can not only detect accidents, but is also aware of potentially distracting actions, such as answering calls or checking emails.

Traffic and road monitoring monitoring sensor networks. Other related work has implemented sensor networks in construction zones to monitor traffic flow and congestion /citebathula2009sensor. These networks must be long-lived, inexpensive, rapidly deployable, and require minimal maintenance. WreckWatch provides all these capabilities at a significantly lowered cost to developers. Moreover, related work has not focused on the increased danger due to construction zones occasionally introducing unfamiliar roads in an area where drivers feel familiarity and comfort. WreckWatch can include not only passive monitoring, but also active alerting and notification.

Monitoring road and traffic conditions using smartphones has been evaluated in past research. Prior work has focused on the sensing component of detecting various contextual items, such as honk detection and physical bump/brake/pothole detection [23]. WreckWatch extends these concepts (*e.g.*, adding airbag deployment detection), capitalizes upon the advantages of utilizing the underlying smartphone cellular infrastructure, provides automated interaction with emergency responders, and automatically notifies emergency contacts, such as family members.

6 Concluding Remarks

Reducing the time between when an accident takes place and when it is detected can reduce mortality rates by 6% [12]. Conventional in-vehicle accident detection and notification systems, such as OnStar, are effective in reducing the time gap before first responders are sent to the scene. These systems, however, are expensive and not available in all vehicles.

To further increase the usage of automatic accident detection and notification systems, smartphones can be used to indirectly detection accidents through their onboard sensors, such as accelerometers. Many challenges must be overcome, however, particularly the potential for false positives from accidentally dropped phones. Due to the large volume of "phantom" (accidental) calls to emergency services, reducing the false positive rate of smartphone accident detection is important.

Using a combination of context data, such as determining when a user is inside a vehicle, sensor data, such as accelerometer and acoustic information, and intelligent sensor data filtering, accident detection systems can be created that are resistant to false positives. For example, air bag deployment is only triggered at over 60G's of acceleration. As shown by experiments in Section 4, accelerations above 4Gs are unlikely for dropped phones.

In developing and evaluating our prototype accident detection and notification system, WreckWatch, we learned the following lessons:

• **Accidents exert extreme forces on a phone that are unlikely to occur when dropping it.** The forces experienced during a car collision are extreme and highly unlikely to occur in any other event other than a high-speed collision. These events are therefore easier to identify and categorize accordingly. Moreover, by combining the accident detection process with contextual information to determine when the user is in a vehicle, false positives are less likely.

• **Smartphones can offer novel situational awareness capabilities.** Uninjured motorists and bystanders can serve as citizen scientists and provide multiple streams of voice and imagery data from the scene of the accident. This information can aid first responders in determining the serverity of the accident, the victims involved, and the urgency of medical care. Moreover, smartphones can provide data about the identify of the victims and automatically alert emergency contacts, such as family members.

• **Smartphones application stores significantly aid in decreasing the cost and complexity of software maintenance.** The built-in application upgrade mechanisms and communication channels on a smartphone make it possible to push updates to thousands or millions of clients and roll back if installation fails. We have found that this capability is quite helpful in maintaining/evolving the software in accident detection and notification systems.

• **It may not be possible to detect all accidents with smartphones.** Due to the filters utilized to prevent false positives, it may be possible to experience a low speed "fender-bender" without the application detecting it. More work is needed to enhance the filtering mechanisms to account for these types of collisions. In particular, WreckWatch's filtering algorithm could be enhanced to determine whether the user is in a vehicle or not utilizing history information. For example, users often travel similar routes to work and WreckWatch could learn where stops or reductions in speed are common by analysis of trends (*e.g.* if a person usually travels through an area at 40mph but occasionally slows to a stop indicating a potential traffic jam). Likewise, WreckWatch could use known intersections to identify potential stops and anticipate them or download traffic information to predict the location of traffic jams resulting from long-duration reductions in speed.

• **Acoustic data is not sufficient for detecting traffic accidents.** Our empirical results show that some smartphone microphones and signal processing infrastructure suffers from signal clipping above 140dBs. This clippling makes it hard to differentiate sounds, such as shouting, from air bag deployment. It is possible that this limitation can be overcome, but it will require additional work.

WreckWatch is an open-source Android application that is freely available from `code.google.com/p/vtnetapps`.

Acknowledgements This work was supported by a grant from the National Science Foundation, RAPID:Collaborative Research:Cloud Environmental Analysis and Relief, CNS# 1047753.

References

1. National Highway Traffic Safety Administration. Federal Motor Vehicle Safety Standards: Occupant Crash Protection - Supplemental Notice of Proposed Rulemaking, 1999.
2. National Highway Transportation Safety Administration. 2007 Traffic Safety Annual Assessment - Highlights. 2008.
3. M. Alsliety. How does SDR fit the telematics model?
4. M. AP TAYLOR. INTELLIGENT TRANSPORT SYSTEMS. *Handbook of transport systems and traffic control*, page 461, 2001.
5. A. Askland. Double Edged Sword That Is the Event Data Recorder, The. *Temp. J. Sci. Tech. & Envtl. L.*, 25:1, 2006.
6. A. Blandford and BL William Wong. Situation awareness in emergency medical dispatch. *International Journal of Human-Computer Studies*, 61(4):421–452, 2004.

7. H.R. Champion, J. Augenstein, A.J. Blatt, B. Cushing, K. Digges, J.H. Siegel, and M.C. Flanigan. Automatic crash notification and the URGENCY algorithm: Its history, value, and use. *Advanced Emergency Nursing Journal*, 26(2):143, 2004.
8. Chris Thompson and Jules White and Brian Dougherty and Adam Albright and Douglas C. Schmidt. Using Smartphones and Wireless Mobile Networks to Detect Car Accidents and Provide Situational Awareness to Emergency Responders. In *Third International ICST Conference on MOBILe Wireless MiddleWARE, Operating Systems, and Applications (Mobilware 2010), June 30-July 2, 2010*, Chicago, IL.
9. A. Cohen and L. Einav. The effects of mandatory seat belt laws on driving behavior and traffic fatalities. *Review of Economics and Statistics*, 85(4):828–843, 2003.
10. J.P. Cohn. Citizen science: can volunteers do real research? *BioScience*, 58(3):192–197, 2008.
11. M.R. Endsley. Toward a theory of situation awareness in dynamic systems. *Human Factors: The Journal of the Human Factors and Ergonomics Society*, 37(1):32–64, 1995.
12. W. Evanco. The Impact of Rapid Incident Detection on Freeway Accident Fatalities. *Mitretek Systems, Inc., WN96W0000071*, 1996.
13. B. Fildes, S. Newstead, JS Barnes, and AP Morris. Airbag effectiveness in real world crashes, 2001.
14. P. Green. Crashes induced by driver information systems and what can be done to reduce them. In *SAE CONFERENCE PROCEEDINGS P*, pages 27–36. SAE; 1999, 2000.
15. L. Groesch, G. Netzer, and L. Kassing. Dummy for car crash testing, October 20 1987. US Patent 4,701,132.
16. J. Harrald and T. Jefferson. Shared situational awareness in emergency management mitigation and response. In *System Sciences, 2007. HICSS 2007. 40th Annual Hawaii International Conference on*, page 23. IEEE, 2007.
17. F. Ichikawa, J. Chipchase, and R. Grignani. Where's the phone? a study of mobile phone location in public spaces. In *Proc. IEE Mobility Conference 2005*. Citeseer, 2005.
18. W. D. Jones. Forecasting Traffic Flow. *IEEE Spectrum*, 2001.
19. A.N. Knaian. *A wireless sensor network for smart roadbeds and intelligent transportation systems*. PhD thesis, Citeseer, 2000.
20. A. Lenhart. *Teens and mobile phones over the past five years: Pew Internet looks back*. Pew Internet & American Life Project, 2009.
21. A. Lie, C. Tingvall, M. Krafft, and A. Kullgren. The effectiveness of electronic stability control (ESC) in reducing real life crashes and injuries. *Traffic Injury Prevention*, 7(1):38–43, 2006.
22. H. Mellander, S. Nilsson, C.Y. Warner, M.G. Wille, and M. Koch. Load-sensing faceform for crash dummy instrumentation, September 8 1987. US Patent 4,691,556.
23. P. Mohan, V.N. Padmanabhan, and R. Ramjee. Nericell: Rich monitoring of road and traffic conditions using mobile smartphones. In *Proceedings of the 6th ACM conference on Embedded network sensor systems*, pages 323–336. ACM New York, NY, USA, 2008.
24. R.S. Naunheim, J. Standeven, C. Richter, and L.M. Lewis. Comparison of impact data in hockey, football, and soccer. *The Journal of Trauma*, 48(5):938, 2000.
25. W.I.S. Query. Reporting System (WISQARS). *National Center for Injury Prevention and Control, Centers for Disease Control and Prevention (producer). Available from: URL: www. cdc. gov/ncipc/wisqars*.
26. S. Rauscher, G. Messner, P. Baur, J. Augenstein, K. Digges, E. Perdeck, G. Bahouth, and O. Pieske. Enhanced Automatic Collision Notification System- Improved Rescue Care Due To Injury Prediction- First Field Experience, 2009.
27. Mikael Rickns. Idg news service. `http://www.infoworld.com/d/mobilize/android-big-winner-smartphone-sales-increase-50-percent-710`, August 2010.
28. G. Rose. Mobile phones as traffic probes: practices, prospects and issues. *Transport Reviews*, 26(3):275–291, 2006.
29. R. Sampson. Misuse and Abuse of 911, 2000.
30. J.E. Saunders, W.H. Slattery III, and W.M. Luxford. Automobile airbag impulse noise: Otologic symptoms in six patients*. *Otolaryngology-Head and Neck Surgery*, 118(2):228–234, 1998.
31. P. Trbovich and J.L. Harbluk. Cell phone communication and driver visual behavior: The impact of cognitive distraction. In *CHI'03 extended abstracts on Human factors in computing systems*, page 729. ACM, 2003.
32. M. Verma, R. Lange, and D. McGarry. A Study Of US Crash Statistics From Automated Crash Notification Data, 2007.

33. RJ Weiland and LB Purser. Intelligent Transportation Systems. *Transportation Research*, 1:40AM, 2009.
34. R.A. Young. Association between embedded cellular phone calls and vehicle crashes involving airbag deployment. In *Proceedings of the 1st International Driving Symposium on Human Factors in Driver Assessment, Training, and Vehicle Design, Aspen, CO*, pages 390–400, 2001.
35. Y. Zhao. Mobile phone location determination and its impact on intelligent transportation systems. *IEEE Transactions on Intelligent Transportation Systems*, 1(1):55, 2000.

=2