October 15, 2019

DETECTING A VEHICLE CRASH USING A MOBILE COMPUTING DEVICE

Marc Stogaitis

DETECTING A VEHICLE CRASH USING A MOBILE COMPUTING DEVICE

Car accidents are one of the leading causes of deaths in the US, causing over 100 fatalities daily. In 2007 alone, more than 43,100 deaths resulted from 10.6 million traffic accidents. Each minute that an injured crash victim does not receive emergency medical care can make a large difference in their survival rate. Analyses show that reducing accident response time by 1 minute correlates to a 6% increase in the number of lives saved. Therefore, reducing the time it takes from when an accident occurs to when the accident is reported is critical to reducing traffic fatalities.

Given the popularity of smart-phones, the large amount of sensors they carry, and their connection to cell-networks, a smart-phone is a suitable device to automatically detect a vehicle crash and provide assistance after the crash. Some phones already process sensor data to detect whether a user is in a vehicle, is running, or has left their phone on a table, and similar force-analysis techniques can be used to detect a car accident. If a mobile device detects that a car accident has occurred, the mobile device can present a user interface that provides assistance to the user, such as by making it easy to call 911 or by automatically text messaging emergency contacts.

DETECTION ALGORITHM

Both precision (avoiding false-positives) and recall (when an accident happens, ensuring that the phone actually determines that an accident has occurred) are important. When an accident happens, it is important that the phone identifies that the accident has occurred because users may come to rely on this functionality (high recall is needed). However, incorrectly

detecting an accident that has not happened (e.g., when someone drops their phone) also has a high cost, because the phone could then unnecessarily text message an emergency contact and/or call an emergency service, depending on the way the phone implements the technology described in this document. This is why a high accuracy algorithm is important.

A phone can use its accelerometer as a primary sensor to detect accidents. Here are some statistics relevant to creating an algorithm that detects accidents:

- A light accident in which a vehicle is rear-ended at 5mph at an angle generates around 30Gs of acceleration.
- A vehicle rear-ended at 15 mph generates over 50Gs of acceleration.
- A vehicle rear-ended at 45 mph generates around 60-90Gs of acceleration.
- A car hitting a stationary object at 55 mph generates around 100-120Gs of acceleration.
- A drag racer going from 0 to 330 mph in 4.5 seconds generates 3.34Gs of acceleration.
- A McLaren F1 hitting the brakes dropping from 60 mph to 0 in 2.8 s generates 1G.
- Dropping a phone is believed to create around < 4Gs of acceleration.
- Deploying an airbag creates a short-duration noise that can exceed 170dB (although some phones clip audio above 145dB).
- Airbags deploy at 60Gs.
- Shouting, loud laughter, and radio at max volume can exceed 145dB.
- 40% of phones are carried in a pocket.

Some of these numbers may change based on the location of the phone (e.g., whether the phone is in a user's pocket), but the numbers indicate that the forces experienced during a car crash are not encountered in normal circumstances. As such, a simple accident-detection algorithm may involve two steps: (1) Detect whether a mobile device is located in a vehicle using existing

activity recognition processes; and (2) Monitor accelerometer data to identify when movement of the mobile device exceeds 5Gs, 6Gs, 7Gs, or 10Gs, for example.

IDENTIFYING FALSE POSITIVES

A false positive may result when a user brakes hard and their phone drops to the floor, slips and hits the edge of a cup holder, or is flung forward and hits the dashboard. There are various ways to address such events:

- The activity recognition process may be able to determine whether a phone is in a pocket, which can affect crash-detection thresholds, because a phone in a pocket may be much less likely to be flung forward during hard braking.
- A user interface may include ways to handle false-positives (e.g., upon detecting an accident, a phone may present a user-selectable button that states “I’m fine, false alarm”, with a 30 second countdown timer before a message is sent to an emergency contact).
- A vehicle crash may have force data that is different than a phone dropping, at least with regard to the duration of the forces (e.g., 60ms vs. 10ms). As such, a threshold length for high-force data could be set, and if the high-force data does not last longer than the threshold, then the event could be discarded as a false positive.
- Another option is to integrate recorded forces over a time period. For instance, both a vehicle accident and other high-force events (like a phone drop) result in high forces, but car accidents result in the high forces lasting longer. As such, a device could integrate forces using a sliding window of length X. If abs(result) > Y, then the phone may determine that an accident has occurred. The integration can use a simple trapezoidal

rule, X can be determined using data (e.g., likely in the 0.5 to 1 second range), and Y can be determined using data.

- The mobile device may lower its threshold for determining that an accident has occurred if the mobile device has been damaged during a high-force event. For example, if a screen is determined to not work after a high-force event, the mobile device may designate that an accident has occurred.
- If a user agrees to on-device audio analysis, the phone may analyze audio to identify if audio present at the same time as the high forces correspond to a crash. There are a few different audio-analysis techniques that can help correlate high forces with a crash. (1) Accidents will typically be accompanied by a significant amount of noise and a simple "loudness" threshold can be set, and any high-force event that is not accompanied by significant loudness can be discarded as a false positive. (2) A car crash audio classifier may be trained using audio taken from actual car crashes (e.g., videos of car crashes that are posted online). Similarly, the car crash audio classifier may be trained at least partially on the sound of an airbag deploying (e.g., because some research states that the sound of an airbag deploying is over 170dB and has a unique audio signature). (3) The timing of audio and acceleration data can be used to detect a crash versus an event that is not a crash. Dropping a phone can result in a loud sound (especially if the phone is dropped on the microphone). Still, the timing of the sounds and the acceleration forces is likely to be different from a car crash, because the sound of a car crash is likely to get to the phone before the high G's of the crash are felt, while a phone drop is likely to result in a roughly 1:1 timing with audio. This third option may require synchronizing audio

and acceleration timestamps, because different chips in a single mobile device sometimes record the audio and acceleration forces.

- Pressure readings from a barometer may be used as a signal to help distinguish false positives from true positives. Many mobile computing devices include barometers, and a pressure spike that occurs during an accident may be greater and/or different in characteristics than a pressure spike that occurs during other forms of high-force events (e.g., dropping a phone to the ground).

USER INTERFACE

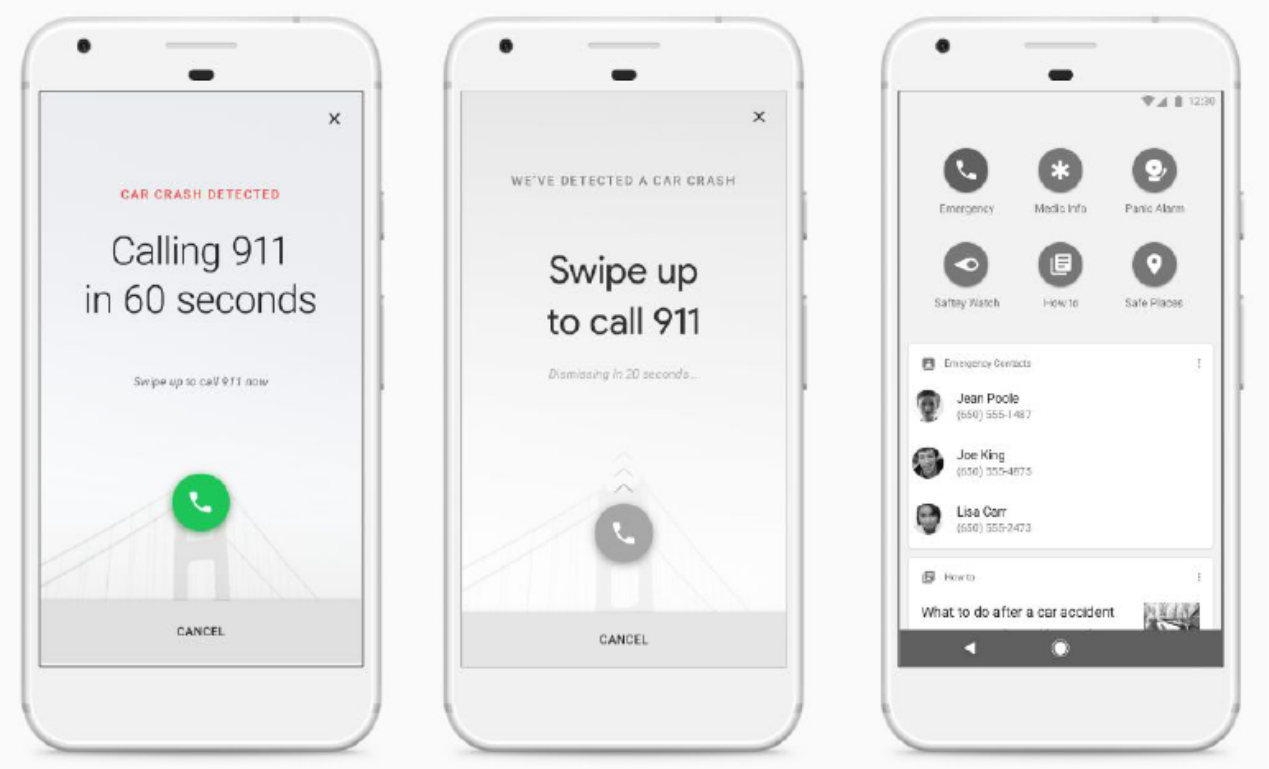
The user experience flow could account for the possibility of false positives. There are various different options for responding once an accident is detected, such as the following (from less automated to most automated):

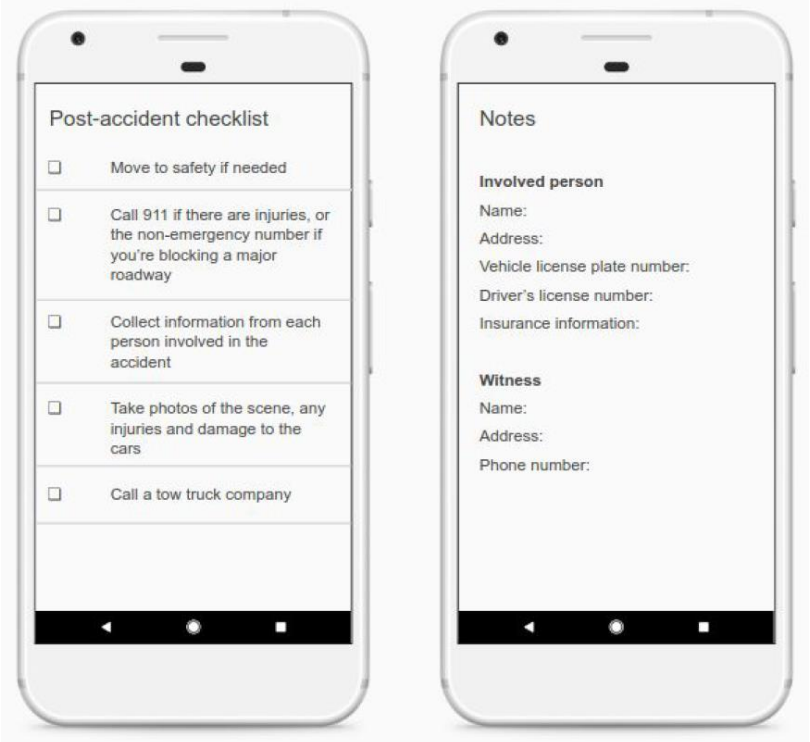
- Show a notification which states "In an accident?". Tapping the notification brings the user to a user interface that includes a one-click button to call 911 or another local emergency number. That same user interface may also include a check-list and other information to assist a user that has been involved in an accident (e.g., present insurance information and a phone number for the user's insurance agent, tell the user to take pictures of each car, get names of parties involved and exchange information with those individuals, and request a police report).
- Show a user interface that includes a countdown that starts at 30 seconds, upon the termination of which the phone may text message emergency contacts that a user designates when opting-in to the accident-reporting process described in this document.

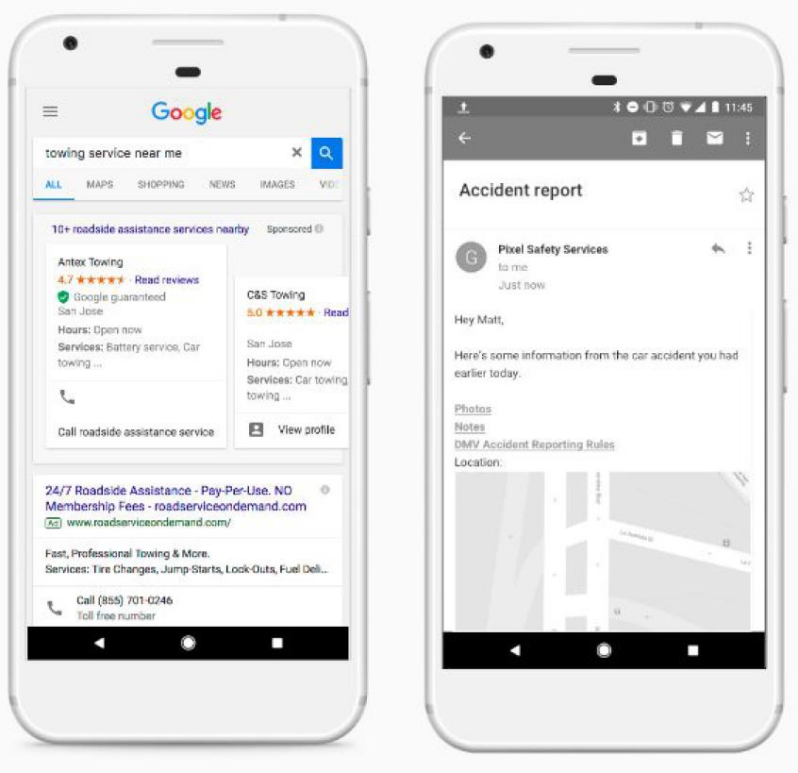
The user interface may also include an “I’m OK” or “No accident occurred” button to cancel the text message, and a button that makes it easy to dial 911.

- Automatically call 911 or another local emergency number. In some examples, the phone (or a remote system that is involved in generating content for the call on behalf of the phone) may automatically announce relevant details with 911 or the other local emergency number. For example, even though the user may not participate in the telephone call, a computer-generated voice may state the nature of the emergency (e.g., a car crash), the location of the telephone, identifying details if available (e.g., car make/model/color), and helpful user medical information (e.g., blood type and allergies). This information would only be provided if the user had previously provided that information and approved its automated dissemination during an accident. Such information may also be automatically provided with a text message to 911 or another local emergency number.

Example user interface screens that the phone may automatically present upon detecting an accident are illustrated below.


CAR CRASH DETECTED
Calling 911
in 60 seconds
Swipe up to call 911 now
CANCEL
WE'VE DETECTED A CAR CRASH
Swipe up
to call 911
Dismissing in 20 seconds...
CANCEL
Emergency
Medis Info
Panic Alarm
Safety Watch
How to
Safe Places
Emergency Contacts
Jean Poole
Joe King
Lisa Carr
How to
What to do after a car accident


Post-accident checklist
Move to safety if needed
Call 911 if there are injuries, or the non-emergency number if you're blocking a major roadway
Collect information from each person involved in the accident
Take photos of the scene, any injuries and damage to the cars
Call a tow truck company
Notes
Involved person
Name:
Address:
Vehicle license plate number:
Driver's license number:
Insurance information:
Witness
Name:
Address:
Phone number:



EVALUATING ALGORITHM

Acceleration data that corresponds to false positives and that could be used to evaluate the accuracy of an accident-detection algorithm could be obtained by dropping phones or analyzing previously-recorded acceleration data in which phones were dropped or used in non-car-accident situations.

One option to obtain data that corresponds to accidents is to partner with organizations that perform vehicle accident testing, and include test phones in the cars to record acceleration forces during the accidents. Example organizations include the Transportation Research Center, the National Highway Traffic Safety Administration (NHTSA), and the Insurance Institute for Highway Safety (IIHS). Another option is to obtain data from organizations that drive vehicles on streets to take 360-degree pictures of the surrounding areas. Some of those picture-taking

vehicles include accelerometers, and there are likely several crashes a year from which acceleration data could be obtained. Another option is to analyze literature that discusses characteristics of accidents to develop appropriate thresholds. Those thresholds and an appropriate algorithm may be incorporated into an application that users could install or activate as part of a study to test the accuracy of the algorithm. Users could specify if an accident occurs and/or a system could analyze the data to look for true positives (e.g., driving followed by large acceleration events, followed by a significant stoppage afterwards). Testing may want to identify optimal sampling rates (e.g., is a 50Hz sampling rate sufficient?).

IMPLEMENTATION DETAILS

To avoid excessive power usage, a low-power processor can continuously monitor accelerometer data and can wake the main processor (e.g., an application processor) when high acceleration forces are detected. In devices that do not have a low-power processor available to continuously monitor accelerometer data, the accident-detection algorithm may be run by the main processor when the additional power cost would be negligible. For example, the phone may run the accident-detection algorithm when a mapping application is running in the foreground (e.g., providing driving directions), because the main processor will already be operating. The accident-detection algorithm may also run anytime the screen is on and the user is determined to be in a vehicle. Another possibility is to run the accident-detection process on mobile devices that support at least 1 minute of acceleration batching.

ABSTRACT

A mobile device such as a smartphone may analyze acceleration data to determine when the mobile device, and therefore the user of the mobile device, has been in a vehicle accident. Detecting whether a vehicle accident has occurred may account for an intensity of acceleration forces, a duration of acceleration forces, and concurrent non-force activities detected by the mobile device (e.g., loud audio or screen breakage). If the mobile device has determined that an accident has occurred, the mobile device can present a user interface that provides access to functions that are helpful following an accident. For example, the mobile device may provide one-touch dialing or texting to 911 or emergency contacts. In some examples, the mobile device may automatically dial or text 911 or emergency contacts upon an automatically-initiated countdown timer completing without user interruption.