Use of Phone Sensors to Enhance Distracted Pedestrians' Safety

Yu-Chih Tung and Kang G. Shin, *Life Fellow, IEEE*

Abstract—Studies have shown that using smartphones while walking—called *distracted walking*—significantly increases the risk of pedestrians colliding with dangerous objects. In this paper, we explore how to mitigate this problem by exploiting the phone's built-in sensors only and developing an application called `BumpAlert`. `BumpAlert` provides a generic solution without requiring any prior knowledge of the user's surroundings by estimating distances to nearby objects using the phone's speakers and microphones. This process is enhanced further by using the images acquired from the phone's rear camera, when necessary. We have evaluated `BumpAlert` under a variety of settings ranging from aisle to outdoor environments with walls, pillars, signboards, dustbins, and people, etc., that are common in our daily surroundings. Our evaluation has shown an average accuracy of `BumpAlert` to be higher than 95 percent with a less than 2 percent false-positive rate to detect frontal objects 2–4m away, which suffices for the user to react and avoid collision. Even though `BumpAlert` is unable to detect all dangerous situations, most participants of our user study feel safer when they walk with `BumpAlert` enabled. Integrating our current design of `BumpAlert` with other safety systems can provide a practical solution for protecting distracted pedestrians.

Index Terms—Mobile sensing and computing, pervasive computing, smartphones, distracted pedestrians

1 Introduction

The risk of injury is reported to increase significantly when pedestrians are distracted by their use of smartphones while walking, i.e., *distracted walking*. Pedestrians are known to notice 50 percent less environmental changes when they text on their phone while walking [1]. According to the number of emergency room visits reported in the United States in 2010, the rate of accidents due to pedestrians' use of smartphone has grown 10x in 5 years [2]. This accident rate is likely to increase sharply with the increase of distracted pedestrians. Such accidents can also be severe; for example, people may walk distracted into the middle of the road and get knocked down by an oncoming car, or may bump into trees or utility poles causing head injuries. Recognizing this growing risk of cellphone users, in Chongqing, a sprawling city in central China, authorities have even set up a "cellphone lane" where people using their phones can stroll without running into anyone or object not holding/using a phone [3]. Also, Taiwan government is about to establish a law to fine distracted pedestrians $10 to reduce the accident rate.

Reducing this risk by using the phone itself without requiring any additional sensors or infrastructural support has been drawing significant attention from both research and industry communities, but has not yet produced a satisfactory solution. Some systems can identify cars by building an image classifier with the images of frontal cars, but cannot detect any object beyond the cars [4]. Some others focus on preventing people from losing steps when they walk through the transitions between pathway and road [5]. While existing approaches address various specific aspects, their reliance on strong assumptions, like the shape or color of objects, prevent them from detecting general obstacles in the user's path. To fill this gap, we propose `BumpAlert`, which addresses an important but unexplored problem: "can commodity phones determine if the user is walking toward (dangerous) obstacles without assuming any prior knowledge of the objects?" Guaranteeing the elimination of all dangerous incidents is the ultimate goal of all safety systems but very hard, if not impossible, to achieve. Like most existing approaches, `BumpAlert` is an add-on phone function to enhance the safety of distracted pedestrians that aims to reduce the accident rate as much as possible at reasonable cost/overhead.

It is challenging to detect obstacles by utilizing only the built-in sensors in commodity phones. To achieve high detection accuracy at low computation and energy costs, we exploit several phone sensors. `BumpAlert` uses the phone's speakers and microphones to estimate the distance between the user and nearby objects, and also uses the phone's rear camera to validate the detected objects, only when necessary. Several novel algorithms are developed and implemented by exploiting these sensor inputs. For example, the false detections caused by omnidirectional phone speakers/microphones are removed by a novel *motion filter* that tracks the user's trajectory using inertial sensors. Also, the distances to obstacles can be estimated by a single camera without depth perception since the phone's height has already been determined by the `BumpAlert`'s acoustic detector. This paper makes several contributions in that `BumpAlert`

• *The authors are with the University of Michigan–Ann Arbor, Ann Arbor, MI 48109. E-mail: {yctung, kgshin}@umich.edu.*

Manuscript received 14 Aug. 2016; revised 10 Sept. 2017; accepted 2 Oct. 2017. Date of publication 23 Oct. 2017; date of current version 3 May 2018. (Corresponding author: Yu-Chih Tung.)
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TMC.2017.2764909

- is the first phone app to "actively" monitor the environment and alert distracted walkers in real time;
- relies only on sensors available in commodity smartphones, without requiring any specialized sensors;
- does not rely on any *a priori* knowledge of obstacles, thus offering a generic solution applicable to a broad range of situations/environments; and
- consumes only a small fraction of resources, thus unaffecting users' experience in using their phones.

`BumpAlert` is implemented on the Android platform as an app using the OpenCV library and the Java Native Interface. Our evaluation results show its capability to detect objects with higher than 95 percent accuracy in typical outdoor/indoor environments and consume around 8 percent of battery power per hour while running as a mobile app.

We have conducted a user study of `BumpAlert` in a controlled environment. Although `BumpAlert` does not guarantee safety for all possible dangerous scenarios that distracted walkers might encounter, our user study shows that 71 percent of the participants agree that `BumpAlert`'s detection accuracy is useful and 86 percent of them are willing to accept `BumpAlert`'s energy cost for detecting dangerous obstacles with a high probability. A user-interface study based on Microsoft Kinect [6] also corroborates that a system displaying frontal obstacles can make distracted walkers feel safer and more confident. Moreover, 43 percent of the participants in our study have experienced bumping into objects during distracted walking, and 86 percent of them have heard others collided with obstacles. These results are consistent with other studies, indicating the real danger of distracted walking. A demo video of `BumpAlert` can be found from [7].

The remainder of this paper is organized as follows. Section 2 summarizes the related work in accident prevention systems. Section 3 gives an overview of `BumpAlert` and Section 4 describes the implementation details. Sections 5 and 6 provide our experimental evaluation and user study, respectively. The paper concludes with Section 9.

2 Related Work

Obstacle detection and avoidance have been an active area of research [8], [9], [10], [11] in the field of intelligent vehicles and robotics. Of particular interest is the active safety systems deployed in cars to protect pedestrians. However, most of these systems require expensive devices such as RADAR, LIDAR, SONAR, and multiple cameras to detect pedestrians and predict their movement. These solutions are not easily wearable by people as they are usually heavy or require advanced sensors, but they can be used as a basis for signal processing, especially for camera imaging and SONAR processing. Note that some robots might use cheap sensors to detect obstacles, but these sensors are still specially designed for this purpose. For example, sonars used in robots are directional while phone speaker/microphone are not. Another direction of study focuses on detection of pedestrian(s) with the help of infrastructure, such as pre-deployed cameras at intersections [12]. However, the same cannot be assumed in mobile phone environments.

Instead of using advanced/expensive sensors, one can find and exploit various built-in sensors of smartphones. These include accelerometers which sense the phone's movement, gyroscopes which detect the phone's orientation, cameras and microphones which capture images and record sound in the surrounding environment. These sensors have been utilized to develop various apps, such as indoor phone localization [13], [14], context-aware computing [15], [16], and human–computer interfaces [17], [18].

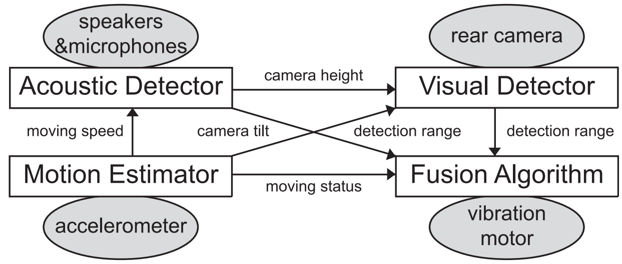


Fig. 1. *System blocks of BumpAlert.* Multiple sensing components are utilized to optimize the detection performance.

Although there exist a myriad of apps that exploit sensors to perform various functions on the phone, little has been done on distracted walkers' safety, despite its rapidly growing importance. A passive approach using the phone's rear camera was proposed in [19], [20] to take and display the frontal image as the background of apps. Since it is a passive solution, the user still has to be responsible for identifying and avoiding the obstacles shown on the screen of his phone. However, users usually focus on the task (e.g., playing a game) at hand and may not pay attention to the changes in the background of the app they are running. Moreover, there are also apps, such as games, that do not allow the change of background.

There are also other mobile apps that sense environments and provide active feedbacks. WalkSafe [4] is able to identify the frontal view of an (approaching) vehicle by using the phone's rear camera when pedestrians are making telephone calls while crossing the road/street. LookUp [5] monitors the road transitions, such as the height change from a sidewalk to a street, by connecting inertial sensors mounted in shoes. Both apps target the scenarios parallel to `BumpAlert`, and it is possible to integrate `BumpAlert` with them to enhance pedestrians' safety. CrashAlert [6] targets the same scenario as ours, detecting obstacles when users are distracted-walking. However, it mainly focuses on the design of walking user interface (WUI). The functionality of obstacle detection in CrashAlert is delegated to Microsoft Kinect, which is not available in commodity phones. In this paper, we explore how to detect and avoid objects in front of a distracted walker by using only the phone's built-in sensors and building and evaluating a mobile application called `BumpAlert`. Even though `BumpAlert` is unable to detect all dangerous situations (see Section 7), it has been shown to be able to detect most dangerous objects for distracted pedestrians, ranging from glass doors, sign boards, to a small parapet wall.

3 BumpAlert

As shown in Fig. 1, `BumpAlert` consists of four main components that interact with each other: (1) *acoustic detector* that uses sound to estimate the distances between the user and nearby objects; (2) *visual detector* that determines the presence of dangerous objects using the rear camera; (3) *motion estimator* that determines the user's walking speed; and (4) *fusion algorithm* that combines information from all the other components and generates an alert for the user when a dangerous object is detected nearby.

3.1 Acoustic Detector

The acoustic detector borrows ideas from sonar sensors for object detection. The speaker sends 10 periods of a sine wave at a frequency of 11,025 Hz and picks up its reflections through the phone's microphones. In order to make `BumpAlert` compatible with most commodity smartphones, the signal sent is sampled at 44.1 kHz and two consecutive signals are transmitted with a 100 ms separation to differentiate their reflections at the microphones/receivers. Note that this setting is designed to be widely supported by commodity phones and can be adapted as phone hardware improves. For example, Section 8 describes the extended setting designed for Galaxy Note4, which provides reasonably good detection accuracy with *inaudible* sound.

Algorithm 1. Acoustic Detection

Input: acoustic signal array at the nth detection: S_n, peak window: win_{peak}, walking speed: w, threshold coefficient: α
Output: detection result: D_{succ}

$S \leftarrow$ matched_filter(bandpass_filter(S_n))
$noise \leftarrow$ estimate_noise(S) & $thr \leftarrow \alpha(noise_{mean}+noise_{std})$
$P_n, D_n \leftarrow \phi$ & $peakMax, peakOffset \leftarrow 0$
for i from $win_{peak}/2$ to $len(S) - win_{peak}/2 - 1$ **do**
  $isPeak \leftarrow True$
  for j from $i - win_{peak}/2$ to $i + win_{peak}/2$ **do**
    if $S[j] > S[i]$ **then**
      $isPeak \leftarrow False$
      break
  if $isPeak$ and $S[i] > thr$ **then**
    $P_n \leftarrow P_n \cup i$
    if $S[i] > peakMax$ **then**
      $peakOffset \leftarrow i$
      $peakMax \leftarrow S[i]$
for $p \in P_n$ **do**
  $d = speed_{sound}(p - peakOffset)/(2rate_{sample})$
  $D_n \leftarrow D_n \cup d$
$D_{succ} \leftarrow$ motion_filter($D_n, D_{n-1}, \ldots, D_{n-\delta}, \delta, w$)
return D_{succ}

To identify the signals reflected from objects, the recorded signal is first passed through a bandpass FIR (Finite Impulse Response) filter and then through the corresponding matched filter as shown in Algorithm 1. At the nth record, the highest-amplitude samples within a moving window are marked as peaks, P_n, if the signal's amplitude exceeds a threshold, thr. Due to the automatic gain control (AGC) in microphones and the different levels of environmental noise, thr is adjusted to the received noise level. The noise is observed from 600 samples before the sent signal is received, with the threshold set to α (mean($noise$)+std ($noise$)) where α is set to 4 in `BumpAlert`. The width of the moving window, win_{peak}, is set to 40 samples, which is equal to the number of samples in the transmitted signal. The maximum resolution that can be discerned with these chosen parameters is about 15 cm, which equals the product of the signal's duration and the speed of sound, so objects within 15 cm of each other will be classified as a single object. Fig. 2 shows an instance of acoustic detection. The first peak indicates the sound sent out of the speaker, while the second peak is the reflection from the human body 28 cm away, and the third peak is the reflection from the floor 142 cm below the speaker. According to the ground truth, the error is less than 5 cm in this case.

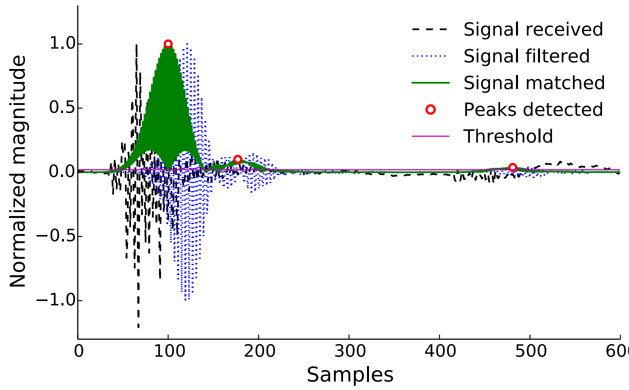
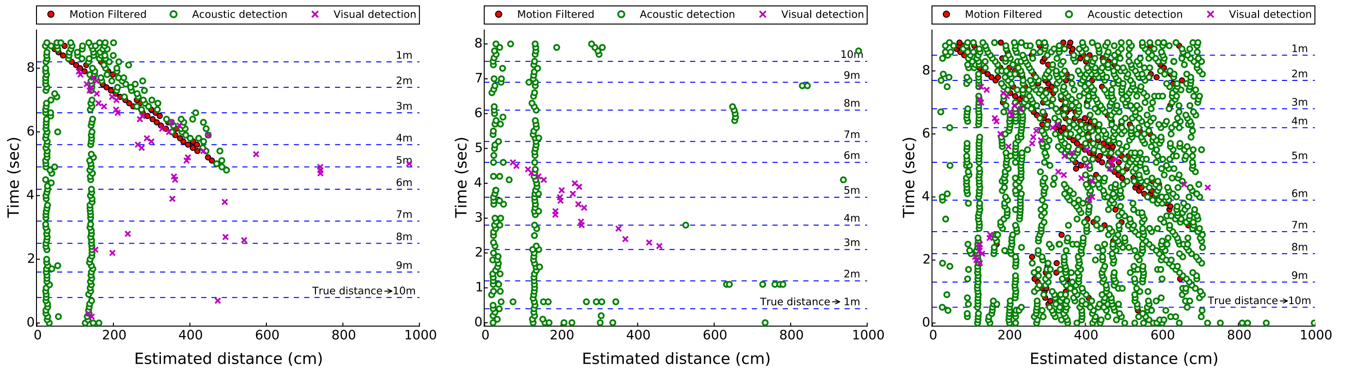


Fig. 2. *Example measurement of acoustic detection.* Peaks of signals passing the matched filter indicate the reception of reflections from objects. The first and strongest peak represents the sound directly transmitted from phone speakers.

The signal used should be lower than a half of the sampling frequency for its accurate recovery. Ideally, a higher frequency is preferred because the sound of such a frequency will be less audible (hence less annoying) to the user, but the sent and reflected signals also degrade more at higher frequencies. On the other hand, decreasing the signal frequency will increase the time necessary to send a sufficient number of periods of the signal, which will lower the detection resolution. Note that there is no need to use a lower frequency signal. A lower frequency signal might incur less decay during its propagation, and can thus receive the reflections from farther-away objects. However, it also increases the time to wait for all reflections before sending the next sensing signal. There are also more environmental noises in the lower frequency band. According to our experimental results, the signal frequency of 11,025 Hz suffices to capture reflections within 2–4 m, and reflections from objects more than 10m away are too weak to be detected for most devices we tested. This is what `BumpAlert` needs, enabling detection of nearby obstacles, and ensuring that all significant reflections are received within 100 ms. Note that the current design of `BumpAlert` does not cope with the interference caused by multiple nearby devices. However, this problem can be avoided by utilizing existing multiple wireless access protocols. For example, different devices can emit different frequencies of sound (FDMA) or different kinds of sound (CDMA) to ensure the emitted signals to have minimal correlation with each other. Testing such advanced settings is part of our future work.

The distance between the user and each object is computed as 1/2 the traveling time of the signal reflected from the object×the speed of sound (331 m/s). The performance of this scheme depends strongly on the ability to accurately record the time when signals are sent and their reflections are received. Errors of a few milliseconds will cause an estimation error of several meters due to the high speed of sound. Thus, any error between timestamps caused by non-real-time phone operating systems is unacceptable. We circumvent this problem by recording the time when the (reflected) signal is sent (received) and then computing the



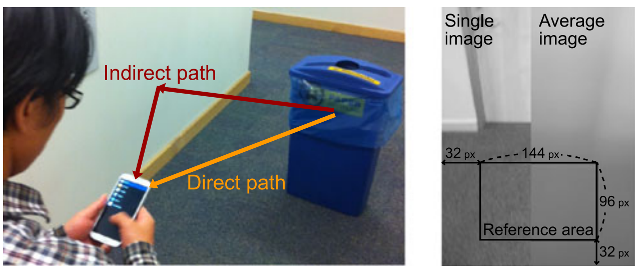
(a) Walking towards a wall (b) Walking away from a wall (c) Walking towards a wall in an aisle

Fig. 3. *Distances to objects estimated by acoustic/visual detectors when a user walks toward/from a wall.* A user is guided to walk towards (away) a wall from a location 10m away (1m behind). The circles and crosses represent the estimated distances to objects (including the wall) detected by our acoustic/visual detectors. The dotted lines represent the real distance to the target wall, which is collected via timestamped traces when users walk through pre-installed tags on the ground.

time difference between the sent and the reflected signals in terms of the number of consecutive samples [21]. As shown in Fig. 2, the signal identified with the largest magnitude, *peakMax*, is regarded as the sent signal, i.e., the signal directly going to the microphone. In Algorithm 1, this is used as the reference, *peakOffset*, for computing the time difference between the sent and the reflected signals. As the detection results shown in Fig. 3a, when a user walks toward a wall (obstacle) from a 10 m-away position, our acoustic detector is able to identify the reflection (as the diagonal green hollow circles) from the wall when the users are 5 m away (as marked with the dotted line) at time 5. In this figure, the constantly appearing objects (two prominent vertical green hollow circles) estimated to be 30 cm and 150 cm away are, respectively, the human body and the floor.

One limitation of acoustic detection is that phone speakers and microphones are *omni-directional*, and hence the direction of the obstacle cannot be resolved. Another related problem is reception of multi-path reflections. The signal received by a microphone is actually a combination of the sent signal and multiple reflections of the same signal. Thus, an object actually 50 cm away may cause a false detection as 150 cm away due to multi-path reflections as shown in Fig. 4a. This effect is severe, especially in an indoor environment where objects like walls and pillars cause multi-path reflections.

However, these two problems are greatly reduced by the `BumpAlert`'s need to detect only the closest object, i.e., the shortest-path reflection. Most reflections from objects behind the user are also absorbed/blocked by the user's body which is akin to the property of WiFi signals being blocked by the mobile users [22], [23]. As shown in Fig. 3b where the user is walking away from a wall, the acoustic detector does not detect any prominent peaks of reflections even if the wall is just 1 m behind the user at time 0. This feature helps the acoustic detector prevent from making wrong estimations when the object is actually behind the user. However, the detection results will still be affected by reflections from the side objects, such as walls and pillars. As shown in Fig. 3c, the same experiment of a user walking toward a wall from 10 m away is repeated but in a narrow (5 m-wide) aisle. False detections are made due to side walls (vertical green hollow circles within a 2–6 m range), making it difficult for `BumpAlert` to identify the real obstacle, i.e., the wall in front of the user.



(a) Example posture and multipath reflections (b) Average of taken images

Fig. 4. *Assumed holding posture and its effect on detections.* A wrong acoustic detection with a longer estimated distance happens due to multipath reflections. The marked area of images taken in this posture includes the ground texture with a high probability.

Algorithm 2. Motion Filter

Input: detection distance of the nth detection: D_n, previous δ detections $D_{n-1}, \ldots, D_{n-\delta}$, depth: δ, walking speed: w
Output: results passing motion filter: D_{succ}

$D_{succ} \leftarrow \phi$
for $d \in D_n$ **do**
  $r \leftarrow 0$
  for i from 1 to δ **do**
    $d_{est} \leftarrow d + i \times w \times period_{detection}$
    for $d_{history} \in D_{n-i}$ **do**
      if $|d_{est} - d_{history}| < win_{error}$ **then**
        $r \leftarrow r + 1/\delta$
  if $r > r_{succ}$ **then**
    $D_{succ} \leftarrow D \cup d$
return D_{succ}

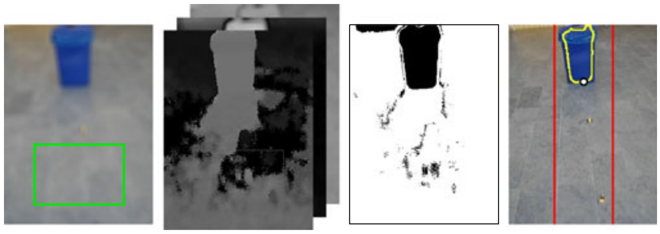
To improve the detection results further, we introduce a *motion filter* that eliminates the detected reflections with 0 relative speed to the user. This filter is inspired by the results shown in Fig. 3, where all detected objects showing a constant distance to the users over time (*vertically* aligned circles) are unnecessary for the functionality of `BumpAlert` since it is impossible for the users to bump into those objects without any relative speed to them. The user's walking speed is estimated by the phone's accelerometer as described later. The high-level goal of this motion filter is to remove detections with relative speeds unmatched with the user's walking speed. Thus, given the user's walking speed, w, and a history of δ previous detection results, $D_{n-1} \sim D_{n-\delta}$, only those reflections from objects moving at similar walking speeds are classified as the true obstacles toward which the user is walking.

As shown in Algorithm 2, the current detection, $d \in D_n$, is projected backward based on the user's walking speed, $i \times w \times period_{detection}$, yielding d_{est}. This is compared with the previously detected position of the object, $d_{history}$, and the probability of the presence of an object increases if the history matches the projection, i.e., $|d_{est} - d_{history}| < wind_{error}$. Yielding a probability, r, higher than a given ratio, r_{succ}, is said to *pass* the motion filter and identified as a positive detection. With this additional filtering, the reflections caused by objects without any matching relative speed, such as floor or side walls, can be filtered out as shown in Fig. 3a. In this figure, detections passing a motion filter (marked as red solid circles) represent only the signals from target obstacles (i.e., the wall users walking towards) while the detections caused by human body, floor, and multi-path reflections inside the wall are excluded. A similar effect can also be found in Fig. 3c, where most reflections from side walls are also filtered out. However, the noisy detections in a cluttered environment cannot be completely eliminated by the motion filter. As shown in the same figure, more than 10 false detections caused by side objects pass our motion filter since those objects are too close to each other, resulting in a significant number of false positives which might annoy users. These false positives are reduced/removed by using the visual detector to ensure the detected object being in front of the user.

3.2 Visual Detector

To overcome the inherent limitation of acoustic detection, an additional sensing layer is added using the phone's rear camera. This removes the false positives and provides information of the object's direction. `BumpAlert` assumes that users will hold their phones in a typical position as shown in Fig. 4, and the rear camera's line of sight will be clear to capture objects in front of the user. `BumpAlert` can send the users texts or generate vibrations to maintain their phone tilt in its operational range. We have conducted a detailed survey of users' willingness to maintain their average phone tilt required for the functionality of `BumpAlert`. See the details of this user study in Sections 5 and 6.

There are two main challenges to detect objects in the rear camera view. The first is to determine the presence of objects and the second is to determine the distance between the user and the objects due to the lack of depth perception in the images taken by only a single camera. `BumpAlert` does not use any *a priori* information, such as the shape and color, to identify the presence of objects. Having no prior knowledge makes `BumpAlert` more general, enabling detection of any type of dangerous objects and preventing collision with them. Detecting objects without any prior knowledge is difficult, though. The goal of `BumpAlert` is, however, not to identify every object in the scene but to know if there is *any* nearby object in front of the user. Specifically, `BumpAlert` adopts the back-projection technique in [24], [25] to identify the objects that are different from the



(a) Blurred image (b) HSV transformation (c) Backprojection (d) Detection result

Fig. 5. *Visual detection.* Images take by phone rear cameras are smoothed, transferred to HSV color scheme, back projected, and then filtered out blob detections.

ground/floor. Its idea is to use the texture of the ground surface where the user is walking on and to compare it with the rest of the image, looking for textural similarities and dissimilarities. As shown in Fig. 5, a 10×10 blurring filter is used first to reduce the noise from the image, and the image is then transformed into the HSV space. The back-projection algorithm is applied to determine which parts of the image are not related to the ground/floor texture. The last step is to apply an erosion filter to remove any residual error from the back-projection algorithm. After completing these steps, blobs with areas larger than a predefined threshold are identified as obstacles and the point closest to the bottom of the image is returned as the closest obstacle.

Some astute readers might observe that the key assumption in the back projection is the knowledge of ground/floor texture. In case an object is erroneously included in the region as a reference of ground/floor, the object will not be seen by our visual detector because the back projection classifies the object as a part of ground/floor. Identifying the ground/floor in an arbitrary image is difficult, but does not cause problems to `BumpAlert`. Images are only taken when users are walking and using their phones at an assumed position as shown in Fig. 4. Under this assumption, we can ensure that a specific area in the image can represent the information of the ground/floor with a high probability. In order to determine this area, we conducted an experiment with 10 participants. They were requested to take pictures while using their phones at a comfortable position 2 m away from a door. The average of all the pictures taken is shown in Fig. 4b, where the dark area indicates the area consisting of ground/floor. The size of that area we choose is 96 pixels × 144 pixels located 32 pixels above the bottom of a 240 × 320 image. The area is chosen above the bottom of the image since it is possible to include the user's feet in the bottom area.

After the closest point of objects in the image is identified, a pixel difference from the detection point to the bottom of the image is defined as p, and the pixel-to-real-world distance transform is computed as $d = \mathrm{pixel\_to\_distance}(p, h_p, t_p)$, where d is the real-world distance to the detected object, h_p and t_p represent the height and the tilt of the user's phone with respect to the ground. A detailed derivation of this transform based on a camera projection model can be found in [26]. This computation is possible only if the height and tilt of the phone are known. As these two parameters are not fixed when people are walking, a method is needed to estimate them online. The phone's tilt can be directly acquired from the accelerometers as $t_p = cos^{-1}(acc_z/acc_{mag})$, where acc_z is the acceleration orthogonal to the phone's surface and acc_{mag} is the magnitude of overall acceleration caused by the

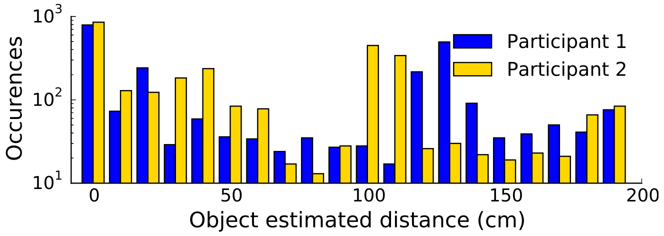


Fig. 6. *Height estimation by the acoustic detector.* The phone's height can be estimated by the sound reflections from the ground.

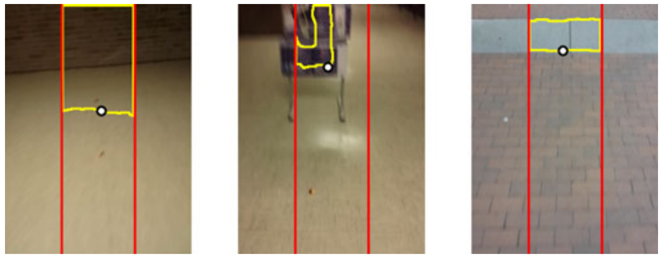
user's motion or the earth gravity. In contrast to derivation of the tilt from the accelerometer readings, the phone's height is unknown when the user is walking. BumpAlert *utilizes the results of the acoustic detector to estimate the phone's height*. This design is novel since existing image-based detections simply assume the height of a camera is known. This parameter might be easy to acquire in certain scenarios, such as installing the camera at a fixed location inside a car, but not in our scenario, since the height of a phone vary with users depending on their height and ways to hold the phone.

The histogram of objects detected by the acoustic detector with different estimated distances are plotted in Fig. 6. This data were collected for two participants of different heights. The maximum peak at distance 0 is the receipt of the transmitted signal. Detections within [10, 60] cm are reflections from the human body. There are also relatively fewer detections in the region [70, 180] cm. The main reason for this phenomenon is that people need a space in front to move forward, resulting in a low probability that there is an object in this range while people are walking. Thus, the highest peaks within this range are actually the reflections off from the floor. As shown in the figure, this is approximately 120 to 140cm for participant 1 and ranges from 100 to 120 cm for participant 2. By tracking the distance in this range, we can estimate the phone's height with an error of less than 20 cm.

Although the visual detector can determine both the direction and distance of frontal objects, it is undesirable for constant/frequent use for the following reasons:

- computational cost of image processing is much higher than acoustic detection, thus consuming more battery;
- distance measured is less accurate than acoustic detection due to the changing tilt and height estimations;
- back projection may be inaccurate for complex floor patterns; and
- falsely identifying pattern transitions on the ground/floor as obstacles.

From our experiments, we found the false positives of visual detection caused by the following three factors as shown in Fig. 7. First, shadows cast on the ground will cause the color of the ground/floor to be different from its surroundings, hence flagging as a different texture area. Second, overhanging obstacles cause the estimated distance to be farther away than the actual position because their bodies are not fully connected to the floor. Third, changing patterns of the ground/floor also cause false detections and are mistaken as an obstacle as they are different from the identified ground/floor texture. A representative error pattern of visual detection can also be found as the purple crosses shown in Fig. 3. For example, there is a burst of false positives between



(a) Shadow (b) Overhang (c) Patterns

Fig. 7. *False detection by the visual detector.* The visual detector might over/under-estimate the distance to objects in different scenarios.

3 and 5 seconds in Fig. 3b, even though there were not any objects ahead. The detection is also less accurate than the acoustic detector. As shown in Fig. 3a, the estimation errors of the visual detectors between 5 and 8 seconds are about 10–100 cm while the acoustic detector has errors less than 15 cm. BumpAlert overcame the above challenges by *combining* the acoustic and visual detectors as described in Section 3.4.

3.3 Motion Estimator

As mentioned in the previous section, the tilt of a phone's camera is directly related to its accelerometer. Similarly, the acoustic detector needs feedback from the phones' sensors that provide information about the user's walking speed to improve the detection accuracy. Using the accelerometer readings, the steps that a person takes can be detected as there exist periods of high and low accelerations. Each peak-to-peak cycle indicates if a step has been taken and the walking speed can be estimated as the product of step frequency and average step size. In BumpAlert, the step size can be either entered by the user or set to the default average step size. This coarse estimation of walking speed is adopted in various applications, such as dead-reckoning systems [27]. The acceleration can also allow the system to determine if the user is walking or stationary when its variance exceeds a predefined threshold.

3.4 Fusion Algorithm

A combination of the above algorithms is used to improve accuracy and lower the false detection rate. We also reduce power consumption by deactivating components that would not improve the detection accuracy. Fig. 8 shows the logical flow of when to run which component based on outputs from other components. First, the detection algorithm need not be run when the user is stationary. We trigger the the detection algorithm only when the user is walking, and switch it off when there is no movement. Second, the low-cost acoustic detector is triggered before the high-cost visual detector. That is, the visual detector is triggered to double-check the acoustic detection result only when the latter is not convincing enough. When the visual detector is enabled, a warning message is issued only when the both detectors find the same object within a 2–4 m range. The acoustic detector is good at detecting the objects around the user within a certain range but is less effective in dealing with side objects (i.e., in cluttered environments). In contrast, the visual detector is free from the side object problem since it focuses on the user's front view. BumpAlert therefore uses a combination of acoustic and visual detections, especially in cluttered environments.

To identify cluttered environments, the motion filter in Algorithm 2 is also used to estimate the number of

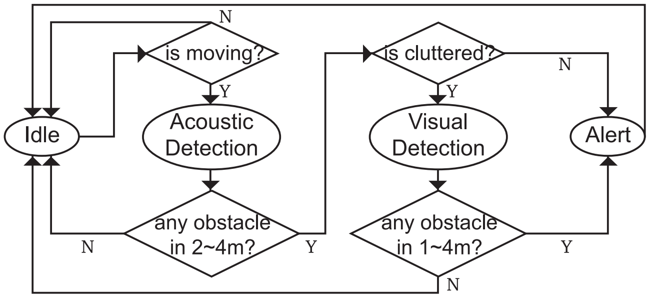
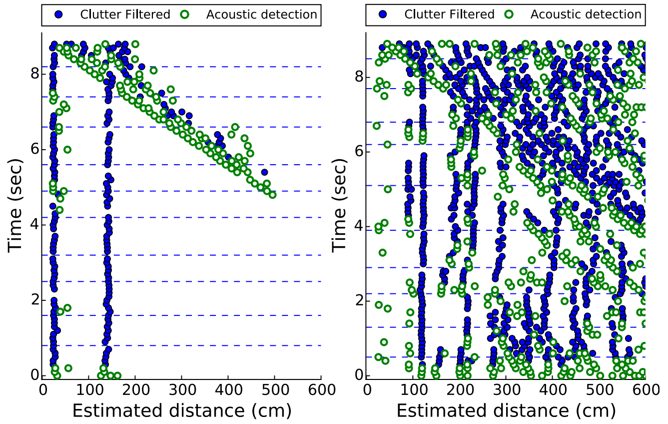


Fig. 8. *Fusion algorithm* if necessary, the visual detector can be enabled to check if the objects found by the acoustic detector actually exist.



(a) Outdoor (b) Aisle

Fig. 9. *Objects identified by the clutter filter.* The clutter filter is a special case of the proposed motion filter for finding objects with 0 relative speed to users. It provides a hint for the fusion algorithm to trigger the visual detector, when necessary.

stationary objects when we set the relative speed to 0. This new application of the motion filter with 0 relative speed is called the *clutter filter*, and its effectiveness is shown in Fig. 9. It can detect those objects that do not have any relative speed to the user. The outdoor environment does not leave many objects after applying the clutter filter, while the aisle environment leaves many objects. Thus, the aisle environment can be identified as cluttered since the number of objects passing the filter exceeds a predefined threshold. Reuse of the motion filter for identifying a cluttered environment is also a novelty of `BumpAlert`, which provides a hint to the fusion algorithm for triggering the visual detector. Existing approaches based on light and geomagnetism changes can only determine if users are located inside a building or not [28], which is not sufficient since disabling the visual detector in a lobby area (indoor) is found to provide better results, but the visual detector is necessary in a cluttered (indoor) aisle.

In the fusion algorithm, different detectors complement each other in different situations. In a cluttered aisle, side walls will be falsely classified by the acoustic detector as obstacles, but are filtered out by the visual detector since such side objects are not captured by the rear camera. On the other hand, crossing from a cement floor to a grassy area is falsely classified as obstacles by the visual detector but is filtered out by the acoustic detector because no reflections are received from the grassy area. By integrating these detectors, `BumpAlert` can therefore discover dangerous objects with high accuracy and a low false positive rate. Note that our current design aims to prevent users from bumping into static objects, like walls, signboards, or pillars. See Section 7 for the discussion of detecting moving objects.

4 Implementation

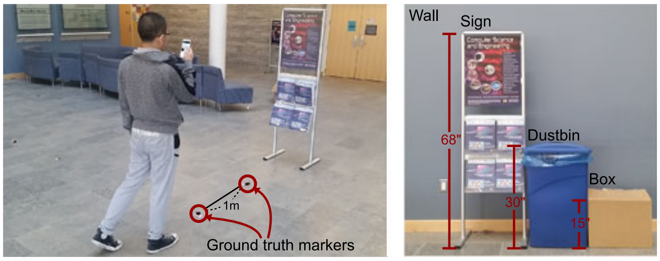
We implemented `BumpAlert` as an Android app on the Galaxy S4. As `BumpAlert` relies only on the phone's built-in sensors, it can be easily ported to different platforms, such as iOS and Windows. For `BumpAlert` to be computationally efficient, the signal processing, such as bandpass and matched filters, are implemented in C and interfaced through the Java Native Interface (JNI), which yielded shorter execution times. The control logics shown in Fig. 8 are implemented in Java due to its low computation requirement. As a result, each iteration of the acoustic/visual detector can be completed within 25/80 ms while its period is set to 100 ms.

We choose the rate to trigger acoustic/visual detectors to be 10 Hz and the sensing range to be 2–4 m in order to balance between detection accuracy and processing cost. According to the results in [29], [30], the average human walking speed is about 1.5 m/s and the reaction time to an auditory alert is about 150 ms. This reaction time is similar to using a vibration alert. Thus, a sensing period of 100 ms with a distance range of 2–4 m is sufficient to alert the user, and the choice of these parameters works well as shown in Section 5.

To run `BumpAlert` with other apps simultaneously, we may choose to implement `BumpAlert` as a system service. However, in the latest version of the Android API, the camera is not allowed to be used in a background service due to privacy issues. Likewise, in `BumpAlert`, images are not saved but only processed for object detection. In future, we will implement `BumpAlert` as an open-source library so that app developers may easily include our modules to enable this functionality to protect their users.

5 Experimental Evaluation

We have conducted a series of experiments to assess the performance of `BumpAlert` in real-world settings. Since the goal of these experiments is to capture and evaluate the performance of `BumpAlert`, we manually selected objects of different sizes and asked participants to walk toward those objects multiple times under different representative scenarios. The benefit of this setting is to collect ground truth and quantize the detection of `BumpAlert` accurately. This information is important for us to infer the performance of `BumpAlert` in the real world but difficult to obtain if participants are allowed to walk toward random obstacles in a single long route. Moreover, as shown in our experiments, the performance of `BumpAlert` depends on the objects and scenarios, so the objects seen in a long route create a significant bias in the final result. For example, a path consisting of 10 walls and 5 dustbins can get a better result than the one of 5 walls and 10 dustbins because wall is an easy target to detect. To avoid this bias, we chose to provide the accuracy of `BumpAlert` against each object in different scenarios rather than the aggregated accuracy in a single long route. The usability of `BumpAlert` is evaluated further in Section 6 via a users study that collects feedbacks from 21 participants who used `BumpAlert` for 15 min. In future, we plan to evaluate `BumpAlert` with more participants over a longer period of time after its deployment.



(a) Test environment (b) Target obstacles

Fig. 10. *Test Setting.* Ground truth markers are used to collect the real distances to the test targets. The selected test targets are ordered by their size, which is related to detection accuracy.

In each experiment, 7 participants are instructed to walk towards various objects, such as walls and signboards in both indoor and outdoor environments. Each of these experiments is repeated 10 times to average the errors due to the differences in each user's walking pattern and path. The participants are instructed to press a specified button when they walk through a marker placed on the ground as shown in Fig. 10a. This serves two purposes. First, it simulates the users being pre-occupied with a task that they would have to accomplish by looking at, or typing on their phones. Second, the ground truth can be collected as the markers are placed at a 1m interval. In this evaluation, we define a positive detection as the obstacle detected within a 2–4 m range. Any alert when the user is 2–4 m away from the target object is classified as a successful alert and the ratio of these alerts is called the *true positive* (TP) rate. On the other hand, any alert occurring when the user is actually 4 m or farther away from the target object is classified as a false alert, and the corresponding rate is calculated as the *false positive* (FP) rate. The average delay is defined as the time from a participant walking through the 4 m marker to the time an alert is triggered.

5.1 Accuracy in Different Environments

In this experiment, a set of 4 objects shown in Fig. 10b are used as obstacles in 3 different environments. They are wall, signboard, dustbin and cardboard box, which are ordered by their relative size. These objects are selected to represent different types of objects in the real world. The difficulty of detection is due mainly to the size of objects. For example, we get similar results for the detection of a glass door and a wall. Moreover, since these objects can easily be found/moved in both indoor and outdoor environments, the performance degradation caused by different environments can be accurately measured in this setting. Other objects, such as pillars and cars, are also tested and shown to have similar characteristics but those results are omitted due to space limitation. The three test environments we used are an open outdoor area, a building lobby, and a (5 m–wide) cluttered aisle. Each participant repeats each experiment 10 times and the 10 Hz raw data of both acoustic and visual detectors are logged to evaluate the detection rate of individual experiment offline by the same detection program. This is to allow for comparison of each individual component based on the same data set, which consists of more than 12 km walking traces. We conducted experiments in the presence of environmental noises, such as students' chatting, but found those noises did not affect `BumpAlert`'s performance much since the frequencies of most noises associated with human activities are below 11 kHz [31] and `BumpAlert` adjusts its detection based on the noise level. The only problem we found is participant 7's outdoor trace collected on a very windy day (more than 24 mph). In this case, the signal received at the phone's microphone was saturated by the wind sound alone, and hence, we postponed the experiment to the next day.

From the results in Table 1, one can see that the acoustic detector outperforms the visual detector in TP rate because the sensing range and sensitivity of the former is longer and better than the latter. The overall TP rate of acoustic detection is higher than 95 percent which is sufficient to identify most dangerous objects. The average delay in all cases is shorter than 650 ms for both visual and acoustic detections. This low delay of `BumpAlert` provides the users walking at 1.5 m/s with more than 2s to react and avoid the obstacles, which is much longer than the human's reaction time [29].

The aisle scenario shows a high FP rate for the acoustic detection due to its cluttered environment. In contrast, the visual detection is not affected by this scenario due to the directional nature of image taken by the phone's rear camera. Therefore, the average FP rate of visual detection in this scenario is even lower than the FP rate of acoustic detection. We exploit this complementary nature of acoustic and visual detectors by using a fusion algorithm to ensure a high TP rate in outdoor environments while significantly reducing the FP rate in indoor environments as shown in Table 1. The fusion algorithm also lowers the FP rate in outdoor environments which are due mainly to a strong wind blowing into the phone's microphones. Actually, many *false*

TABLE 1
Comparison of Performance in Different Environments

	Wall			Signboard			Bin			Box		
	TP (%)	FP (%)	Delay (ms)	TP (%)	FP (%)	Delay (ms)	TP (%)	FP (%)	Delay (ms)	TP (%)	FP (%)	Delay (ms)
Outdoor-Acoustic	100	0.6	320	98.3	5.6	516	96.7	2.8	567	91.7	3.5	470
Outdoor-Visual	63.3	9.8	247	85.0	27.6	265	85.0	13.9	251	75.0	19.9	485
Outdoor-Fusion	100	0.5	433	98.3	2.2	610	95.0	1.7	572	90.0	1.8	508
Lobby-Acoustic	98.3	1.3	108	93.3	1.2	318	96.7	0.6	278	93.3	1.3	321
Lobby-Visual	78.3	11.9	297	61.7	4.5	323	86.7	12.7	496	71.7	7.5	711
Lobby-Fusion	98.3	1.3	111	93.3	1.0	367	96.7	0.6	290	93.3	1.3	325
Aisle-Acoustic	100	32.1	105	100	28.2	193	100	29.1	203	98.3	28.0	245
Aisle-Visual	98.3	28.4	45	100	27.5	598	100	22.2	417	100	26.8	465
Aisle-Fusion	91.7	9.8	330	100	6.0	547	95.0	6.1	447	91.7	6.3	566

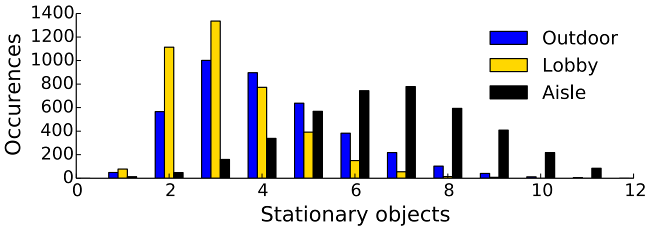


Fig. 11. *Stationary objects passing the clutter filter.* Cluttered areas can be identified by monitoring the number of stationary objects.

detections in the 5 m-wide aisle are not incorrect since there exist objects, e.g., the water fountain on the side and the emergency sign at the ceiling, which is *in front of* users within 2–4 m. If the detection range is shrunk to 1–2 m, the FP rate of acoustic detector is reduced from 28 to 5 percent and it is reduced further to 2 percent when a combination of acoustic and visual detection is applied. Note that there is a trade-off between false detection and detection range. One possible resolution is to alarm users when they are located in a cluttered environment where the detection range of `BumpAlert` is shrunk, and hence they need to pay closer attention to their walking. This is part of our future work.

As stated in Section 3, the key component of the fusion algorithm to work properly is its ability to estimate the number of stationary objects through the clutter filter. In real-world experiments, we set the threshold of stationary objects to classify environments as cluttered as 5 (i.e., turning off the visual detection when there are < 5 stationary objects). The distribution of stationary objects in different scenarios is plotted in Fig. 11. Our experimental results also validate the effectiveness of the clutter filter in enabling the visual detector under a proper condition.

Of all the objects considered, the wall is found the easiest to detect due to its large size, and the box is the hardest in terms of TP ratio and delay. Moreover, the TP rate of signboard detected by the visual detector is lower than that of other objects, which is due to the signboard overhanging above the floor as shown in Fig. 7b. Although the visual detection for the signboard is above 80 percent in outdoor and aisle environments, their high TP rates are also accompanied by a high FP rate. This implies that the detector was guessing most of the time, leading to the high TP and FP rates and not a true representation of accurately detecting the object.

Many other objects have also been tested but the results are not reported here due to space limit. One interesting finding is that acoustic detection of a human is harder than a box even when the human is much larger than the box. This is because the human body absorbs most sound signal instead of reflecting it. We found that the acoustic detector can only detect humans within a 1–3 m range under the current setting, which is shorter than the other objects we tested. Nevertheless, `BumpAlert` can still detect humans with a TP ratio higher than 82 percent. Moreover, the chance of bumping into a person is less likely than other stationary objects because people usually try to avoid distracted walkers. An alternative solution to this problem is to continuously monitor objects with an additional signal of different (low) frequency which is easy to be reflected by the human body.

TABLE 2
Individual Detection Rate of the Trace in Lobby

id	acoustic TP(%)	acoustic FP(%)	visual TP(%)	visual FP(%)	$\bar{h_p}$ (m)	$\bar{t_p}$ (°)
p1	97.5	5.4	97.5	36.4	1.3	52
p2	100.0	1.8	100.0	11.1	1.1	54
p3	95.0	2.5	87.5	21.3	1.3	53
p4	100.0	3.2	90.0	17.6	1.1	39
p5	90.0	0.2	12.5	2.7	1.0	31
p6	100.0	1.7	100.0	32.2	1.2	65
p7	100.0	2.6	100.0	17.6	1.2	56

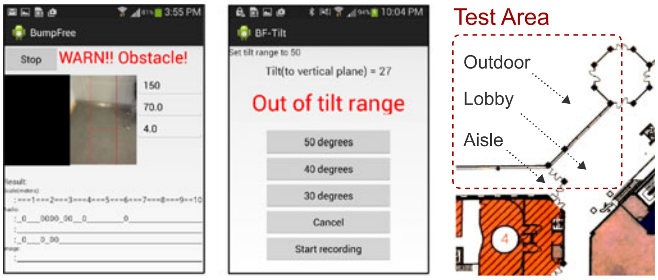
Even though the current version of `BumpAlert` does not handle moving objects, it is general enough to detect a variety of objects in real time. The issue of detecting moving objects like humans or cars will be part of our future work, and it might be addressable by using other complementary approaches such as those in [4], [6].

5.2 Accuracy among Different Participants

To study the effects of different participants with different phone-holding positions and walking patterns, the above results are separated based on individual participants. The phone tilts/heights and the corresponding detection results are summarized in Table 2. According to our experiments, the tilt of phones, t_p, varies from 31° to 65° among different participants; so does the phone height h_p vary from 1 to 1.3 m. These parameters for the same user did not vary much over time.

An interesting finding is that the acoustic detection accuracy is slightly different among participants. We have repeated several tests with different holding positions and found that the variation is affected by the way the phone is held and the AGC of the phones' microphones. For example, when the speakers are being blocked by fingers, the received signal strength is low due to the obstruction. On the other hand, if the phone is held tightly, the magnitude of the received signal sent directly from the phone is increased. This signal may be strong enough to saturate the range of the microphones, and the reflected signals are usually weaker due to the lower gain adapted by AGC. However, with the adaptive threshold mechanism as described in Section 3, `BumpAlert` can accurately estimate the noise level and detect reflections effectively.

The extreme low visual detection ratio of participant p5 was caused by his way of holding the phone, 30° with respect to the horizontal plane. The detection results we collected from participant p5 show that only those images close to (within 1 m of) the obstacles can yield a sufficient area for detection because of the low tilt of the phone, implying that our visual detection is not applicable to certain postures of holding a phone. We also recruited two additional participants who hold their phones with a posture similar to participant p5's to repeat the above experiments. Our results indicate that the visual detector is unable to function with tilt lower than 30° for identifying 2 m-away objects. However, the high probability of successful visual detection by the other users also implies that visual detection works with a broad range of tilts from 40° to 65°. One potential way to address this issue is to warn the users, when they enable `BumpAlert` but hold their phones with



(a) Demo app (b) Tilt survey app (c) Test location

Fig. 12. *Survey settings.* The demo version of `BumpAlert` processes the acoustic/visual detectors in real time. The separate tilt survey app records phone tilt when participants walk and provide feedback if the phone tilt is not in the selected range.

the tilt less than 40°. According to the users study in Section 6, most users feel comfortable with this operating range of `BumpAlert`.

5.3 Processing Cost and Energy Consumption

Our final experiment is to evaluate `BumpAlert` for its real-time performance and resource consumption. Under its four different configurations, we ran `BumpAlert` for an extended period of time in typical environments. The CPU usage of `BumpAlert` is logged via the *top* command at an interval of 10 seconds. A 1-hour trace is averaged to obtain CPU usage as well as power consumption. Four different scenarios are tested: idle (with backlight on), acoustic detection only, visual detection only, and trace. The idle case is used as a baseline which mainly represents the power consumed by the backlight. In the case of acoustic or visual detection only, each algorithm is run independently at 10 Hz with backlight on. Since the energy consumption depends on how often `BumpAlert` turns on/off the visual detector, we also include a real-world trace from participant 1 where the visual detector was enabled only when necessary. This trace is collected when the participant is walking between his home and work. We chose to display participant 1's result because his on-foot travel time is longer than the other participants.

The CPU usages when the app is Idle, in Acoustic only, and Visual only are 3.08, 8.92 and 17.80 percent, respectively. One can see that the CPU usage of Visual detector is approximately twice the value of Acoustic detector. As the high CPU usage, the power consumption of the visual detector is also observed to be much higher than the acoustic detector's. For example, the acoustic detector only consumes one-fourth more energy than the idle baseline (with backlight on) but the visual detector consumes twice more energy. In our experiments, most of the energy is consumed by the microphone/speaker/camera hardware, not by the computation [32]. Thus, the capability of reducing the energy consumption in software is limited. Note that the percentage of additional energy consumed by `BumpAlert` will be reduced further when users turn on WiFi/4G or play mobile games. In the actual usage as the trace of participant 1, the S4 battery only has an additional 8 percent drop after one hour usage.

6 Users Study

We randomly selected 21 passers-by (10 females and 11 males) in our campus without prior knowledge of `BumpAlert` to evaluate its usefulness and practicality. The users were asked to try out `BumpAlert` for 15 minutes and fill out a survey form. Users tried a demo version of `BumpAlert` as shown in Fig. 12a at locations shown in Fig. 12c. The results are summarized in Table 3.

TABLE 3
Survey Results (%)

Questions	Disagree	No option	Agree
I can play my phone around 40° (at walking for detecting obstacles)	10	0	90
I can play my phone around 50°	10	18	72
I can play my phone around 60°	80	5	15
Detection accuracy is helpful	14	14	72
Detection range is acceptable	28	0	72
False alarm is bothering	39	32	29

The first section of our survey attempts to analyze the prevalence of distracted walking. Our result indicates that 81 percent of the participants use their phones while walking and 43 percent of them had run into obstacles due to distracted walking. Even though a half of the participants did not bump into any obstacle before, 76 percent of them were afraid of running into obstacles when they use their phones while walking. The percentage of people colliding with obstacles increases to 86 percent if their friends who had bumped into objects are included.

The second section of the survey attempts to know the tilt when the users hold their phones and check if people are willing to hold phones in a specific tilt range for the benefit of obstacle detection and warning. A separate Android survey app shown in Fig. 12b was used to record and inform the participants of the tilt in holding their phones. They were first asked to walk with `BumpAlert` enabled to record tilts when they hold their phones in the most comfortable position. Then, we selected several different angles that allow the survey app to monitor the tilt of phones and provide a feedback (via vibration and a red text) when the user does not hold the phone in the selected angle within a $\pm 10°$ range.

The phone tilt has been studied extensively in [33] by continuously recording the tilt via published Android widgets. However, the users' state (e.g., walking or sitting) when the tilt is recorded was not reported there. In our users study which records the phone tilt when users are walking, most participants hold their phones at approximately 35° relative to the ground. This result matches the average phone tilt when Google Maps is run as reported [33]. This tilt distribution is not optimal for `BumpAlert` as shown in Section 5. However, after having experience in holding phones with different angles and being told about our purpose, 90 percent of participants were willing to hold their phones between 40° and 50°, which is proven good for `BumpAlert`. Thus, it is reasonable to provide similar feedback when `BumpAlert` is enabled but the tilt of phones is not in the operation range.

The third section of the survey asks participants to evaluate the usefulness of `BumpAlert` after a 15 min trial in three scenarios as shown in Fig. 12c. The three criteria we used are the detection accuracy, detection range and false-alarm rate. About 72 percent of the participants agree that the detection accuracy and range are adequate, allowing them enough time to react to imminent obstacles. Some participants have commented that they would be able to avoid obstacles at even a shorter distance, such as 1.5–3 m. This feedback was useful for `BumpAlert` to reduce the false positive ratio. 29 percent of the participants did not want to have any false alarm. We found some of participants react even to the correct detection of a wall 4 m away as a false alarm. Based on the performance of `BumpAlert` we were able to satisfy most participants with low false positive rate and good detection ratio.

The last section of the survey addresses the issue of power consumption. Only 14 percent of the participants want the power consumption to be below 4 percent per hour. The power consumption of `BumpAlert` varies from user to user, depending on the users' activities. In our initial experiment, power consumption is approximately 8 percent per hour, which meets the criteria of 86 percent of people who are willing to use the application.

Even though the study of 21 users is somewhat limited, it did help us understand what the users need. For example, besides the quantitative results mentioned earlier, during the user study, we also noticed that the users' satisfaction with `BumpAlert` is strongly dependent on the user interface (UI). For example, in a crowded area, users are more comfortable when the UI shows a detailed notification like "*Crowded area detected. Don't use your phone while walking*" rather than a message like "*BumpAlert is off*". Many of the feedbacks we received actually made us adjust our design as shown in Section 8. Crafting a proper UI and building a large-scale user study are parts of our future work.

7 Limitations and Discussion

Based on our evaluation and users study, `BumpAlert` has been proven able to prevent distracted walkers from colliding with various obstacles, ranging from glass doors to small dustbins. However, the current version of `BumpAlert` has a few limitations. A test deployment via PhoneLab [34] or Amazon Mechanical Turk [35] might be the next step to evaluate how `BumpAlert` works for different devices, obstacles, user heights, walking patterns, or phone-holding postures. Also, discussed below are possible venues to detect moving objects, minimize the liability of missed detections, and avoid the audible sound interference.

7.1 Detection of Moving Objects

In addition to the various static objects we have already tested for the evaluation of `BumpAlert`, its current version cannot detect moving objects since they have the unmatched relative speeds and are thus filtered out by the motion filter. There are several potential solutions to address this issue. For example, instead of just matching the pedestrian's walking speed with the speeds of objects moving toward the user, a more sophisticated machine learning algorithm might be able to distinguish the detections caused by different objects, and then track their moving trajectories. However, this type of complex algorithm might consume more energy/computation resources, and generate additional false detections. Finding a balance between detection capability and computation cost is part of our future work.

7.2 Liability of Missed Detections

As mentioned earlier, `BumpAlert` is unable to warn users of "all" dangerous objects, and it is also not the purpose of

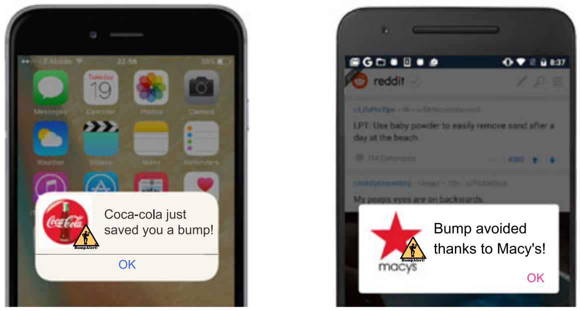


Fig. 13. *An example user interface for the business of app developers.* `BumpAlert` executes in the background with no disturbance to users and the warning with third-party advertisements is shown only when dangerous obstacles are detected.

`BumpAlert`. Some objects might be detected by integrating `BumpAlert` with other existing systems while others may not. For example, distracted pedestrians might fall by stepping through the gap from sidewalks to streets, but `BumpAlert` will not be able to detect this gap since there is nothing in the gap to reflect the audio signals. This situation can be prevented by incorporating an existing system designed specifically for recognizing the street gaps [5]. The same principle can also be applied to the detection of moving vehicles [4]. However, no matter how the system is integrated and designed, there will always be possible missed detections. That is, all warning systems including `BumpAlert` are to enhance, but not to guarantee, distracted walkers' safety.

We argue that even an expensive system relying on many specialized sensors still experiences miss detections, e.g., the recent tragic accident of the latest Tesla autopilot driving model [36]. The main goal of `BumpAlert` is to provide distracted pedestrians additional safety protection with only minimal resources. So, users should not expect to navigate based solely on `BumpAlert` but exploit the `BumpAlert`-provided warning for their safety. Fig. 13 shows an example user interface for developing `BumpAlert` as a freemium which lowers the users' expectation of 100 percent detection rate. App developers still get paid via advertisements in the alert view when obstacles are detected correctly.

7.3 Annoyance Caused by Audible Sound

As mentioned before, `BumpAlert` relies on a 11 kHz beep to sense environments. Although only a short (i.e., 40 samples) sequence of sound is emitted, imperfect speaker design makes the beginning and end of this sound louder than expected. This audible noise is due mainly to the hardware limitation of commodity phones.

The authors of [37] have shown that a 22 kHz sound can be used to send data at a low bit-rate with proper signal processing. However, the purpose of `BumpAlert` is different from theirs in that the emitted sound should be strong enough to generate reflections from obstacles rather than sending data in a best-effort manner. Moreover, their results did not account for the limitation of speaker hardware either, since a special speaker (unavailable in commodity phones) was used in their evaluation. In our experiments with Galaxy S4/5, inaudible sound of 22 kHz is unable to detect objects within 2–4 m. This result is also consistent with their hardware study; the signals captured by certain commodity phones at 22 kHz are 30 dB weaker than those

TABLE 4
Audible Sound Survey (%)

Questions	Disagree	No option	Agree
I can tolerate 11 kHz sound beep (on the purpose to detect obstacles)	42	10	48
I can tolerate 4 kHz sound beep	48	10	42
I can tolerate 441 Hz sound beep	42	29	29
I can tolerate 11–22 kHz chirp	95	0	5
I can tolerate Music fused beep	39	29	32

in the audible range. The responses to this audible sound among 21 participants are summarized in Table 4.

The participants were asked to answer the questions after trying `BumpAlert` and based on the assumption that it can help them avoid collision with obstacles during distracted walking. As shown in this table, even with prior knowledge of `BumpAlert`'s purpose, only 48 percent of them support the sound emitted by the current version of `BumpAlert`. Other lower frequency sounds received even less support. The use of a wide-band chirp, which can further enhance the accuracy via pulse compression, was rejected by 95 percent of the participants. An interesting candidate to *hide* the audible beep is to fuse the signals into a music. For example, an instrumental music is selected and the music signal of 10–12 kHz is filtered out and replaced with our sound signals, and the emitted beeps can thus be played stealthily. However, even fewer users support this idea since some think playing music while walking actually gets more attention from other people. But only 10 percent of the participants chose not to support any of these sound candidates. Thus, `BumpAlert` may provide multiple sound signals for each user to choose based on his preference. Utilizing different sound signals can also enable multiple users to run `BumpAlert` simultaneously, where the received signals from different users can be differentiated by the corresponding filters. `BumpAlert` can also use inaudible sounds to detect objects with newer mobile devices that are equipped with better-fidelity microphones/speakers, thus causing no disturbance to users. Next, we present this light modification of `BumpAlert` based on our evaluation results and user feedbacks.

8 BumpAlert+

From the participants' feedback after using `BumpAlert`, we found most users favoring less user interference, such as running the detection in background, no audible noise, and low false detection over high detection accuracy. For example, they prefer to turn off the object-detection function in a high false positive (e.g., crowded) area rather than getting many false and correct detections. Moreover, while most of the participants in our study liked the benefits of `BumpAlert`, only 48 percent of them were happy with the sound signal (of 11 kHz). `BumpAlert` relies on 11 kHz beeps to sense environments because it provides the best sensing capability among the mobile devices we tested. Inaudible sound of 22 kHz with Galaxy S4/5 is unable to detect objects within a 2–4m range, because the signals captured at 22 kHz are significantly weaker than those in the audible range [37]. To preserve the safety of distracted walkers without annoyance, we design and implement an extended system called `BumpAlert+` which provides reasonable detection accuracy with nearly zero user annoyance. `BumpAlert+` is designed as a background system service which uses only an inaudible sound to sense environments. In a crowed area, `BumpAlert+` will not check the image taken by rear camera but pop up a warning message asking users to take care by themselves, and temporarily turns off the detection. The detection range is shrunk to 3 m since many participants in our study regarded the detection of objects 3 m away as false detections. Currently, `BumpAlert+` can only be executed on Galaxy Note 4 as it provides the highest sensing capability of inaudible signals among the devices we tested. We believe that the design of both `BumpAlert` and `BumpAlert+` can be improved and generalized for devices that will likely emerge in the near future.

The main modification employed in `BumpAlert+` is to use 25 ms-long 18 kHz–24 kHz chirps sampled at 48 kHz to sense the environmental reflections. We choose the chirp signal instead of a pure tone since we need to boost the SNR of received signals in this inaudible band. To make this sound inaudible to humans, `BumpAlert+` also applies similar fade-in/out windowing at the beginning and the end of each chirp as shown in [37]. Our experimental results show the signal to noise ratio (SNR) of this particular sound design on Galaxy Note 4 provides sufficient signal strength to detect nearby objects. Porting `BumpAlert+` to other devices with compatible hardware settings is part of our future work. In `BumpAlert+`, each chirp sensing period is decreased from 100 ms to 50 ms since the detection range is set smaller and the audio frequency is higher (so the reflections from far objects decay more quickly). This chirp signal setting also provides less estimation errors and finer granularity due to the property known as *pulse compression* [38]. It is worth noting that the matched signal strength of both the 11 kHz tones and inaudible chirps will degrade when users are moving due to a known effect called *doppler shifts*. However, considering the normal case where users are walking at 1.5 m/s (i.e., a doppler factor of 1.009), this degradation is negligible. This degradation can be mitigated further by adopting the doppler-invariant sensing signals, such as the hyperbolic frequency-modulated waveforms [39].

Based on this new audio setting, the acoustic detector is modified as follows. First, instead of estimating distance by using the highest correlation peak, a one-time calibration is done by sending 10 repetitions of a wide-band pilot signal before using `BumpAlert+`. This calibration process compares the received and the sent pilots, and ends when the microphone/speaker sample offset is tuned to less than 5. After getting the matched filter results as shown in Section 3, a time-varying gain is applied to compensate for the decay of the signals reflected from far objects. This is accomplished by multiplying a dynamic gain, i.e., $gain(x) = x^{1.65}$ where x is the audio sample offset. An example of this new detection when the user is walking toward the corner of an aisle is plotted in Fig. 14. This figure can be regarded as a higher-resolution version of Fig. 3c, where the bright areas represent the likelihood of an object detection. As shown in this figure, we reuse the clutter filter to remove objects with speed 0 relative to the user (such as the ceiling or side walls) before applying the motion filter. After removing those objects, we set the threshold to 0.12 in order to alarm users if the median of motion filtered area exceeds this threshold.

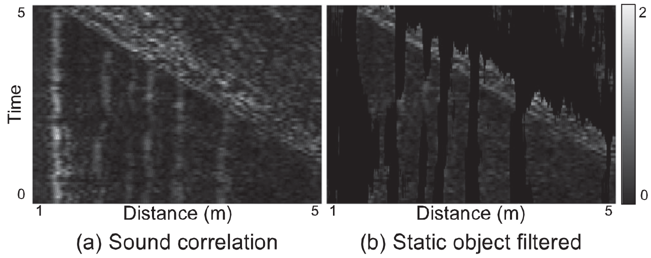


Fig. 14. *Acoustic detection in BumpAlert+.* Bright areas indicate the possible existence of detected obstacles.

Our measurements show that BumpAlert+ yields comparable results as BumpAlert in identifying objects for the scenario shown earlier. We also tested many other objects in both open and crowded areas, and plotted the results in Fig. 15. Thin objects like flat poles are invisible to BumpAlert+ and aisles with width less than 3 m are just marked as consistent warning. These results can be further improved by setting a more aggressive threshold. For example, setting the threshold to 0.08 can make the round pillar and the phone station detectable with 98 percent accuracy with only 4 percent false positive rate. However, as mentioned earlier, BumpAlert+ is designed to remove/mitigate users' annoyance, and hence the parameter setting is tuned to ensure a low false positive rate with high priority. This result shows that BumpAlert+ serves this design purpose, providing reasonable detection accuracy with nearly zero user disturbance. A demo video of BumpAlert+ can be found from [7].

As mentioned earlier, this *inaudible* optimization is tuned mainly based on Note 4, and different devices might have varying results of using the same setting. Fig. 16 shows the device capability of using BumpAlert+ to detect a 1.5 m-high parapet wall when it is 2 or 3m away from users. The peak detect energy ratio is used to characterize its capability of detecting objects. For example, when the wall is 3 m away from users, we first calculate the peak of the reflected signal strength between 2.8 m and 3.8 m and then divide this value by the peak detection energy in the same range of a reference data collected without any obstacle. This metric represents the signal strength of the acoustic reflections to be captured by the device hardware. As shown in Fig. 16, Note 4 can receive more than 19 dB peak detect energy ratio from the inaudible reflections when the object is 3 m away, while S4 only captures less than 5 dB even when the object is 2 m

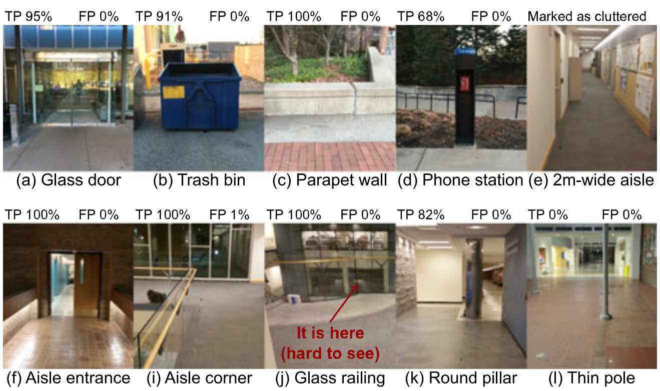


Fig. 15. *Performance of BumpAlert+.* Various scenarios have been tested by walking toward the obstacles from a position 10m away.

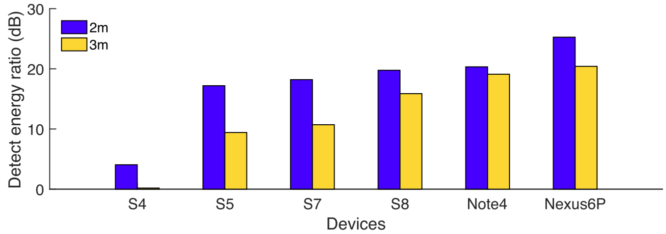


Fig. 16. *Device Compatibility.* Detect energy ratios are measured as the peak received acoustic energy when the target is present versus the peak energy in an open area without obstacles.

away. Among the devices we tested, Nexus 6P can provide the best result with BumpAlert+. We also notice that the detection capability of Samsung Galaxy S-series devices has improved over time, i.e., S8 $>$ S7 $>$ S5 $>$ S4. Based on our testing results, the current setting of BumpAlert+ can be applied to S8 and Nexus 6P easily. Repeating our previous tests on different devices, like detecting different objects when users are moving, is part of our future work.

9 Conclusion

We have explored how to reduce the accident rate of distracted walking by using only phone sensors. A prototype called BumpAlert has been designed, implemented and evaluated as a mobile app to warn distracted pedestrians of imminent collision with obstacles. Since BumpAlert relies only on built-in sensors of commodity phones, it can be easily deployed on different platforms. BumpAlert detects obstacles by fusing several sensor inputs with minimal computation and energy overheads. In the current implementation of BumpAlert, the accuracy of detecting objects in front of the user is higher than 95 percent in both outdoor and indoor environments. This high detection rate of BumpAlert is achievable in a wide spectrum of real-life environments, ranging from glass doors to small dustbins, since it does not depend on any *a priori* knowledge of detected objects. Our users study has shown BumpAlert to be acceptable to the general public and a lightweight version called BumpAlert+ is also proposed based on the users' feedback on BumpAlert. We expect BumpAlert and /or BumpAlert+ will reduce accidents caused by distracted walking.

Acknowledgments

The work reported in this paper was supported in part by the US National Science Foundation under Grants CNS-1505785 and 1646130.

References

[1] J. Lim, A. Amado, L. Sheehan, and R. E. Van Emmerik, "Dual task interference during walking: The effects of texting on situational awareness and gait stability," *Elsevier Gait Posture*, vol. 42, no. 4, pp. 466–471.
[2] J. L. Nasar and D. Troyer, "Pedestrian injuries due to mobile phone use in public places," *Accident Anal. Prevention*, vol. 57, pp. 91–95, 2013.
[3] Chinese City Creates a Cell Phone Lane for Walkers. (2017). [Online]. Available: https://www.newsweek.com/chinese-city-creates-cell-phone-lane-walkers-271102
[4] T. Wang, G. Cardone, A. Corradi, L. Torresani, and A. T. Campbell, "WalkSafe: A pedestrian safety app for mobile phone users who walk and talk while crossing roads," in *Proc. ACM Int. Workshop Mobile Comput. Syst. Appl.*, 2012, pp. 5:1–5:6.

[5] S. Jain, C. Borgiattino, Y. Ren, M. Gruteser, Y. Chen, and C. F. Chiasserini, "LookUp: Enabling pedestrian safety services via shoe sensing," in *Proc. ACM 1st Int. Conf. Mobile Syst. Appl Serv.*, 2015, pp. 257–271.

[6] J. D. Hincapié-Ramos and P. Irani, "Crashalert: Enhancing peripheral alertness for eyes-busy mobile interaction while walking," in *Proc. ACM SIGCHI Conf. Human Factors Comput. Syst.*, 2013, pp. 3385–3388.

[7] BumpAlert Demo Video. (2017). [Online]. Available: https://kabru.eecs.umich.edu/?page_id=987

[8] J. Borenstein and Y. Koren, "The vector field histogram-fast obstacle avoidance for mobile robots," *IEEE Trans. Robot. Autom.*, vol. 7, no. 3, pp. 278–288, Jun. 1991.

[9] J. Minguez, "The obstacle-restriction method for robot obstacle avoidance in difficult environments," in *Proc. IEEE Int. Conf. Intell. Robots Syst.*, 2005, pp. 2284–2290.

[10] V. Philomin, R. Duraiswami, and L. Davis, "Pedestrian tracking from a moving vehicle," in *Proc. IEEE Intell. Vehicles Symp.*, 2000, pp. 350–355.

[11] H. Shuldiner, "Volvo stops for pedestrians," *Ward's Dealer Business*, vol. 43, no. 12, pp. 9–12, 2009.

[12] H. Veeraraghavan, O. Masoud, and N. Papanikolopoulos, "Computer vision algorithms for intersection monitoring," *IEEE Trans. Intell. Transp. Syst.*, vol. 4, no. 2, pp. 78–89, Jun. 2003.

[13] K. Liu, X. Liu, and X. Li, "Guoguo: Enabling fine-grained indoor localization via smartphone," in *Proc. ACM 1st Int. Conf. Mobile Syst. Appl Serv.*, 2013, pp. 235–248.

[14] S. P. Tarzia, P. A. Dinda, R. P. Dick, and G. Memik, "Indoor localization without infrastructure using the acoustic background spectrum," in *Proc. ACM 1st Int. Conf. Mobile Syst. Appl Serv.*, 2011, pp. 155–168.

[15] S. Nirjon, et al., "Auditeur: A mobile-cloud service platform for acoustic event detection on smartphones," in *Proc. ACM 1st Int. Conf. Mobile Syst. Appl Serv.*, 2013, pp. 403–416.

[16] Y.-C. Tung and K. G. Shin, "EchoTag: Accurate infrastructure-free indoor location tagging with smartphones," in *Proc. ACM 9th Annu. Int. Conf. Mobile Comput. Netw.*, 2015, pp. 525–536.

[17] J. Wang, K. Zhao, X. Zhang, and C. Peng, "Ubiquitous keyboard for small mobile devices: Harnessing multipath fading for fine-grained keystroke localization," in *Proc. ACM Proc. 1st Int. Conf. Mobile Syst. Appl Serv.*, 2014, pp. 14–27.

[18] Y.-C. Tung and K. G. Shin, "Expansion of human-phone interface by sensing structure-borne sound propagation," in *Proc. ACM 1st Int. Conf. Mobile Syst. Appl Serv.*, 2016, pp. 277–289.

[19] AndroidWalk N Text. (2017). [Online]. Available: https://play.google.com/store/apps/details?id=com.incorporateapps.walktext

[20] Talk When U Walk. (2014). [Online]. Available: https://play.google.com/store/apps/details?id=com.a3logics.talkwyw

[21] C. Peng, G. Shen, Y. Zhang, Y. Li, and K. Tan, "BeepBeep: A high accuracy acoustic ranging system using cots mobile devices," in *Proc. ACM 1st Int. Conf. Mobile Syst. Appl Serv.*, 2007, pp. 1–14.

[22] Z. Zhang, et al., "I am the antenna: Accurate outdoor ap location using smartphones," in *Proc. ACM 9th Annu. Int. Conf. Mobile Comput. Netw.*, 2011, pp. 109–120.

[23] S. Sen, R. R. Choudhury, and S. Nelakuditi, "SpinLoc: Spin once to know your location," in *Proc. 17th Int. Workshop Mobile Comput. Syst. Appl.*, 2012, pp. 12:1–12:6.

[24] A. Derhgawen and D. Ghose, "Vision based obstacle detection using 3D HSV histograms," in *Proc. Annu. IEEE India Conf.*, 2011, pp. 1–4.

[25] I. Ulrich and I. R. Nourbakhsh, "Appearance-based obstacle detection with monocular color vision," in *Proc. 26th AAAI Conf. Artif. Intell.*, 2000, pp. 866–871.

[26] J. Fernandes and J. Neves, "Angle invariance for distance measurements using a single camera," in *Proc. IEEE Int. Symp. Ind. Electron.*, 2006, vol. 1, pp. 676–680.

[27] X. Zhu, Q. Li, and G. Chen, "APT: Accurate outdoor pedestrian tracking with smartphones," in *Proc. IEEE Conf. Inf. Comput. Commun.*, 2013, pp. 2508–2516.

[28] P. Zhou, Y. Zheng, Z. Li, M. Li, and G. Shen, "IoDetector: A generic service for indoor outdoor detection," in *Proc. ACM Conf. Embedded Netw. Sensor Syst.*, 2012, pp. 113–126.

[29] J. Brebner, "Reaction time in personality theory," in *Reaction Times*. London, New York. Toronto: Academic Press, pp. 309–320, 1980.

[30] R. C. Browning, E. A. Baker, J. A. Herron, and R. Kram, "Effects of obesity and sex on the energetic cost and preferred speed of walking," *J. Appl. Physiology*, vol. 100, no. 2, pp. 390–398, 2006.

[31] S. P. Tarzia, P. A. Dinda, R. P. Dick, and G. Memik, "Indoor localization without infrastructure using the acoustic background spectrum," in *Proc. ACM 1st Int. Conf. Mobile Syst. Appl Serv.*, 2011, pp. 155–168.

[32] F. Ben Abdesslem, A. Phillips, and T. Henderson, "Less is more: Energy-efficient mobile sensing with senseless," in *Proc. ACM Workshop Netw. Sys. Appl. Mobile Handhelds*, 2009, pp. 61–62.

[33] A. S. Shirazi, N. Henze, T. Dingler, K. Kunze, and A. Schmidt, "Upright or sideways?: Analysis of smartphone postures in the wild," in *Proc. ACM 5th Int. Conf. Human-Comput. Interaction Mobile Devices Serv.*, 2013, pp. 362–371.

[34] PhoneLab: A Smartphone Platform Testbed. (2017). [Online]. Available: https://www.phone-lab.org/

[35] Amazon Mechanical Turk. (2017). [Online]. Available: https://www.mturk.com/mturk/welcome

[36] Tesla driver killed in crash with Autopilot active, NHTSA investigating. (2017). [Online]. Available: http://www.theverge.com/2016/6/30/12072408/tesla-autopilot-car-crash-death-autonomous-model-s

[37] P. Lazik and A. Rowe, "Indoor pseudo-ranging of mobile devices using ultrasonic chirps," in *Proc. ACM 10th ACM Conf. Embedded Netw. Sensor Syst.*, 2012, pp. 391–392.

[38] S. Salemian, M. Jamshihi, and A. Rafiee, "Radar pulse compression techniques," in *Proc. IEEE Int. Symp. Microw. Antenna Propag. EMC Technol. Wireless Commun.*, pp. 203–209.

[39] J. Yang and T. Sarkar, "Doppler-invariant property of hyperbolic frequency modulated waveforms," *Microw. Optical Technol. Lett*, vol. 48, no. 6, pp. 1174–1179, 2006.

Yu-Chih Tung received the MS degree from the National Taiwan University, in 2012. He is working toward the PhD degree in computer science & engineering with the University of Michigan–Ann Arbor, advised by Prof. Kang G. Shin. His research interests include building mobile applications and security systems based on physical-layer signal processing. He formerly worked as a research assistant (intern) at HP Labs with their Mobility team, at CSAIL MIT with Prof. Dina Katabi, and at Academia Sinica with Dr. Kate Ching-Ju Lin. He has presented his research at top-ranked conferences since 2011, including ACM MobiCom, Mobisys, and CCS. Various applications based on his research have been implemented and are in the process of commercialization.

Kang G. Shin received the BS degree in electronics engineering from Seoul National University, Seoul, Korea, and the MS and the PhD degrees in electrical engineering from Cornell University, Ithaca, New York, in 1970, 1976, and 1978, respectively. He is the Kevin and Nancy O'Connor Professor of Computer Science and founding director of the Real-Time Computing Laboratory in the Department of Electrical Engineering and Computer Science, The University of Michigan, Ann Arbor, Michigan. At Michigan, he has supervised the completion of 80 PhDs and also chaired the Computer Science and Engineering Division for three years starting in 1991. From 1978 to 1982 he was on the faculty of the Rensselaer Polytechnic Institute, Troy, New York. His current research focuses on QoS-sensitive computing and networks as well as on embedded real-time and cyber-physical systems. He has authored/coauthored more than 900 technical articles (more than 330 of which are published in archival journals) and more than 40 patents or invention disclosures. He has also received numerous institutional awards and best paper awards. He is a life fellow of the IEEE.