9-2014

CrossNavi: Enabling real-time crossroad navigation for the blind with commodity phones

Longfei SHANGGUAN

Zheng YANG

Zimu ZHOU
Singapore Management University, zimuzhou@smu.edu.sg

Xiaolong ZHENG

Chenshu WU

See next page for additional authors

Author

Longfei SHANGGUAN, Zheng YANG, Zimu ZHOU, Xiaolong ZHENG, Chenshu WU, and Yunhao LIU

CrossNavi: Enabling Real-time Crossroad Navigation for the Blind with Commodity Phones

Longfei Shangguan‡, Zheng Yang†, Zimu Zhou‡, Xiaole. . . shu W. . . iu†

‡CSE Department, Hong Kong University of Science and Technology, Hong Kong
†School of Software and TNList, Tsinghua University, China

ABSTRACT

Crossroad is among the most dangerous parts outside for the visually impaired people. Numerous studies have exploited navigating systems for the visually impaired community, providing services ranging from block detection, route planning to realtime localization. However, none of them have addressed the safety issue in crossroad and integrated three key factors necessary for a practical crossroad navigation system: detecting the crossroad, locating zebra patterns, and guiding the user within zebra crossing when passing the road. Our *CrossNavi* application responds to these needs, providing an integrated crossroad navigation service that incorporates all the essential functionalities mentioned above. The overall service is fulfilled by the collaboration of built-in sensors on commodity phones, and requires minimal human participation. We describe the technical aspects of its design, implementation, interface, and further improvements to make the system practical on a wider basis. Experimental results from three visually impaired volunteers show that the system exhibits promising behavior in both urban and rural areas.

Author Keywords

Smartphone; Crossroad Navigation; Sensors

ACM Classification Keywords

H.3.4. Information Storage and Retrieval: Systems and Software

INTRODUCTION

Over 285 million people are estimated to be visually impaired worldwide [2]. These people cannot walk about unaided, especially when safely crossing roads in urban areas. Usually they use a white cane to detect the tactile landmarks (e.g. lowered curbs in the sidewalks Fig. 1 (a)) and align themselves with zebra crossings. However, many crossroads are still not equipped with proper tactile paving facilities (Fig. 1 (b)). And situations are even worse in developing countries, where 90% of the world's visually impaired live. It is common to see these assistant facilities damaged (Fig. 1 (c)) or blocked by improper infrastructure design (Fig. 1 (d)). Moreover, it is still challenging for the blind to cross the road even if they have been aligned to the zebra crossings. The crosswalk is
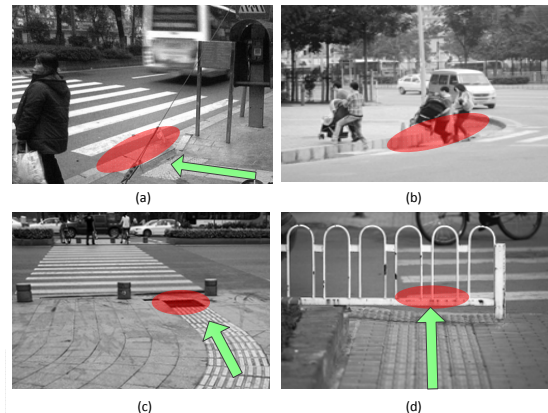
Figure 1. The assistant facilities are damaged or blocked by improper infrastructure design.

marked by zebra patterns which are visually identifiable only. Therefore, blind people may easily veer off the zebra crossings and get stuck in the middle of a busy street.

A crossroad navigation system is critical yet challenging. It should be able to detect crossroads, locate zebra patterns, and ensure the user within the area of zebra crossings when passing the road. Traditional ways such as white canes and guide dogs provide inadequate information or may receive limited acceptability, thus rendering them infeasible or pernicious in practice. Recent advance in wireless and embedded technology has fostered the flourish of electronic travel aids (ETAs) market. Representative systems include wearable navigating devices [13, 20], intelligent white cane [1], and anti-veering systems [8, 17]. Unfortunately, such systems are mostly single-purposed (e.g., detecting obstacles, planning travel paths, or rectifying veering) and relatively expensive, thus not accepted worldwide.

The confluence of advanced sensor technology and widely available smartphones broadens the possibilities for ultra-portable, low cost navigating approaches. Today's smartphones possess powerful computation capabilities and integrate multi-functional sensors. These advances lay solid foundations of ambient sensing for crossroad navigation. In this context, researchers have leveraged commodity smartphones with computer vision techniques to locate crosswalks [3, 7, 12]. However, the resulting systems require users to manually take photos with their phones, which is cumbersome and even infeasible for the visually impaired. Despite intrusiveness issue aforementioned, these systems fail to bring all of essential services, including crossroad detection, zebra crossing localization, and veering angle rectification, together into a practical working application.
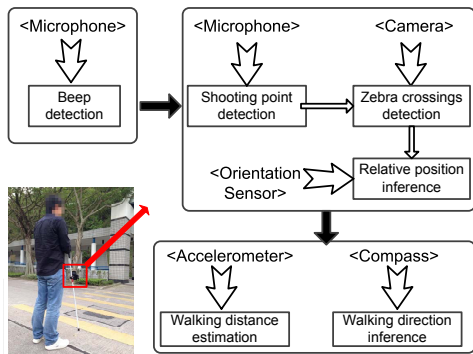
Figure 2. Work-flow of *CrossNavi*. We purposely blur the face of volunteer for double-visually impaireded review.

In this paper, we propose a new smartphone application, *CrossNavi*. Rather than a simple zebra crossing localizer, *CrossNavi* provides an integrated crossroad navigation service that incorporates all the necessary functionalities aforementioned. The overall service is fulfilled by the collaboration of built-in sensors on commodity phones, and requires minimal human participation. Specifically, *CrossNavi* first exploits the unique acoustic features of traffic buzzers for crossroad detection. It then automatically triggers camera to shoot images in front and leverages advanced Computer Vision (CV) techniques, along with orientation data to locate zebra crossings. After guiding users to align to zebra crossings, *CrossNavi* records accelerometer and compass readings to recover the user trail, based on which it provides auditory hints to rectify unintentional veering during user's movement, thereby protecting user from walking outside the crosswalk.

Despite the convenience of *CrossNavi*, to upgrade it to an applicable level, the challenges are threefold. First, the system should be *non-intrusive*, scheduling sensing tasks automatically without explicit human intervention; Second, *CrossNavi* should execute in a *realtime* manner, providing navigating service promptly; Third, as an user-end application running on energy starving devices, *CrossNavi* should be *lightweight*, incurring minimum, or at least ignorable overhead to mobile devices (e.g., energy, storage, and CPU workload). We address these key issues by introducing a module driven structure and a set of efficient algorithms. For example, instead of applying computational-intensive stereo vision techniques for crosswalk localization, we combine low-power sensors with lightweight CV techniques, and achieve satisfying localization accuracy with reduced latency.

The only external input that *CrossNavi* depends on is a set of road parameters of interest, which is easily obtained via mainstream positioning service providers such as Google Map and AutoNavi. Although *CrossNavi* is specially designed for the visually impaired, the implications of this system may extend beyond. For example, the crossroad detection module can label landmarks (crossroad), and further enhance inertial sensor based outdoor navigation systems. Our current prototype of *CrossNavi* provides auditory feedbacks to assist user to cross the road. The performance evaluation shows that the system exhibits promising behavior. In summary, the key contributions of this paper are:

Figure 3. examples of audible units around crossroads.

1. We identify the opportunity of navigating the visually impaired to cross roads with smartphones without user intervention. Our approach leverages the rich sensing capabilities of smartphones with human movement patterns to enable an automatic, non-intrusive crossroad navigation scheme. To our best knowledge, this is the first work that provides an integrated crossroad navigation service for the visually impaired with smartphones.

2. To improve system scalability, for example, latency and energy, we design a module-driven architecture and put forward efficient algorithms for crossroad navigation. Such design uses modularity to separate each functional unit, and thus is flexible for individual module upgrading.

3. We fully implemented *CrossNavi* on Android platforms and conducted extensive experiments in various scenarios. The evaluation shows that system exhibits promising performance.

The rest of the paper clarifies each of these contributions, beginning with a system overview, followed by design, implementation and evaluation of *CrossNavi*. Finally, we summarize the limitations and point out potential future work.

SYSTEM OVERVIEW

We build the framework of *CrossNavi* in awareness of three design objectives: non-intrusive, real-time, and lightweight. Fig. 2 shows the system architecture. Specifically, the framework consists of three modules, crossroad detection, zebra crossings localization, and user veering rectification.

1) Crossroad detection: Initially, *CrossNavi* works in low-power listening mode. It keeps the microphone open to collect acoustic signals. Once the standard deviation of the acoustic samples exceeds a pre-calibrated threshold, *CrossNavi* performs the autocorrelation operation on these acoustic samples to ascertain the user is approaching the crossroad. The rationale behind is that beep signals of traffic light around crossroads show a repetitive pattern which serves as a robust indicator for crossroad approaching. It is worth noting that all the other sensors keep in silence in this state. The successful detection of crossroad then triggers zebra crossings localization module.

2) Zebra crossings Localization: *CrossNavi* uses camera and orientation sensor to characterize the spatial relationship between the user and zebra crossings. To achieve a non-intrusive localization scheme, the phone is required to bound on the white cane for automatic image shooting. However, images captured during fast swing can be blurred. The trick in *CrossNavi* is to capture the short stable period during each swing cycle (tip tends to rest on the ground for a while before swing backwards), and trigger camera accordingly to
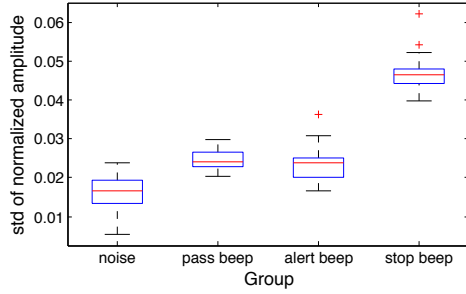
Figure 4. distribution of std **values of acoustic traces collected on different sites and states.**

take clear photos. If zebra crossings are successfully detected, *CrossNavi* estimates its location via the calibration of orientation sensor and camera.

3) User veering rectification: Based on the spatial information of zebra crossings, *CrossNavi* then guides users to align to zebra crossings via voice reminders. Once *CrossNavi* detects that the zebra crossing is in front of the user, it automatically triggers the veer rectification module to sense user headings and walking distance, based on which it estimates the safe veering (based on geometry analysis) and reminds users to rectify their headings properly. This process continues until users safely cross the road.

SYSTEM DESIGN

Approaching the Crossroad?

CrossNavi utilizes image processing techniques to detect zebra crossings. However, most image processing techniques are computationally intensive. To avoid unnecessary computation and reduce energy overhead of image processing, it is crucial to launch the image-based zebra crossing detection scheme only when the user approaches a crossroad. An intuitive way for this is using GPS. However, empirical results demonstrated localization errors up to 50 meters, especially in urban areas, thus making it inapplicable for fine-grained crossroad navigation. In this section, we explore the microphone on smart phones to perceive the unique ambient context of crossroad, and accurately identify the moment when the user is approaching the crossroad.

Characteristics of Beep signals
Audio signals are widely adopted to indicate the status of crossroads (e.g., Scotland, Hong Kong, Singapore). Fig. 3 shows audible units (buzzers) fixed along with stoplights at public crossroads in Hong Kong. They periodically send a short and rapid beep audio to indicate the *Go* status, while a slow and repeating pattern for the *Stop* status. Therefore, we take advantage of the beep signals to infer whether the user is approaching a crossroad[1].

Despite complicated and noisy acoustic environment at busy streets, the beep signals can serve as a robust indicator for crossroad approaching detection for two reasons. First, the beep signals lead to radical changes of acoustic intensity (e.g.,

---

[1]*CrossNavi* works only on the crossroad where the buzzer works around-the-clock (without human triggering). We suggest viewers to learn more on crossroad navigation for the blind in [11].
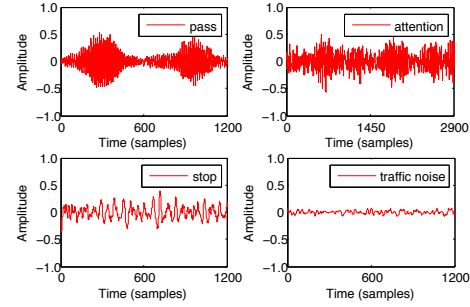

Figure 5. Normalized auto-correlation over different acoustic profiles.

the standard deviation of acoustic samples will increase.) which serve as a good indicator for the presence of crossroads. Second, such beeps exhibit repetitive patterns, which can be robustly detected via autocorrelation.

Fig. 4 illustrates the distribution of standard deviation values (std for short) over different traffic states, with 20000 acoustic samples each. As it demonstrates, the std values of acoustic traces collected near the crossroad are significantly higher than those of noises collected on the street. In addition, more than 95% of the std values of acoustic traces collected on the street are below 0.02, while all of those values collected near the crossroad are larger than 0.02. Thus, in *CrossNavi*, we set a threshold ϵ as 0.02 (which is empirically optimal) for crossroad detection.

On Street vs. Around Crossroad

While beep signals lead to large std values of acoustic profile, the inverse is not always true. For example, the unexpected car horns may also lead to a large std value. Therefore, to improve the crossroad detection accuracy, *CrossNavi* double-checks the detection result by executing the autocorrelation on the acoustic profile whose std is greater than ϵ. Specifically, given a sequence of acoustic samples S, *CrossNavi* computes the normalized auto-correlation for lag τ at the $m - th$ sample as:

$$\chi(m,\tau) = \frac{\sum_{k=0}^{k=\tau-1}[s_{m+k} - \mu(m,\tau)][s_{m+k+\tau} - \mu(m+\tau,\tau)]}{\tau \cdot \sigma(m,\tau) \cdot \sigma(m+\tau,\tau)}$$

(1)

Where $\mu(k,\tau)$ and $\sigma(k,\tau)$ are the mean and standard deviation of the normalized sequence of acoustic samples $< s_k, s_{k+1}, ..., s_{k+\tau-1} >$, respectively. For ease of presentation, we denote the maximum value of $\chi(m,\tau)$ as χ_{max}.

Fig. 5 depicts the distribution of $\chi(m,\tau)$ computed on different acoustic traces. One was recorded on the street, and the other one was sampled in the vicinity of the crossroad. As seen from Fig. 5, when χ is 0.35 or higher, the probability that the person is not approaching the crossroad is extremely low. Thus, we set the threshold λ as 0.35 which is empirically optimal. *CrossNavi* then recognizes the crossroad as follows:

- if $std \leq \epsilon$, then $place = street$;

- if $std > \epsilon$ and $\chi_{max} \geq \lambda$, then $place = crossroad$;

Figure 6. (a): the phone coordinate system; (b): the decomposition of cane swing cycle; (c): analytical acceleration profile along the Z axis during a cane swing cycle; (d): acceleration readings along the Z axis during a cane swing cycle;

Figure 7. An illustration of the motion trail of the white cane.

Type	Optic angle (α)	Detection range (r)
Galaxy S III	$\approx 63.44°$	$\approx 104.64°$
Iphone 5	$\approx 59.54°$	$\approx 100.74°$
Iphone 4S	$\approx 59.54°$	$\approx 100.74°$
Galaxy Note II	$\approx 59.54°$	$\approx 100.74°$
Iphone 4	$\approx 59.54°$	$\approx 100.74°$

Table 1. comparison of optic angle of best-selling smart phone in 2013.

Where's the Zebra?

To guide the visually impaired to go across roads, *CrossNavi* should recognize zebra crossings and perceive its location. The advance of computer vision (CV) ripens the real-time zebra crossing recognition techniques. However, all of these works (e.g. [3, 7, 12]) assume that the images (containing zebra crossings) are available and ready for processing or require the visually impaired to shoot images manually. In practice, as the visually impaired are insensitive to graphic marks of zebra crossings, it is difficult and often infeasible for them to shoot images manually. Besides, due to the hardware limitation, the camera on commodity phones has limited optic angle. Therefore, to capture zebra crossings, we are often required to shoot in different directions, which put extra pressure to the visually impaired.

Automatic image shooting mechanism (AIS)
Principle: *CrossNavi* aims to capture zebra crossings without human interventions. We use camera motion to emulate a wide-angle lens. The intuition behind is that when the person is walking, the white cane in hand will experience a *Zigzag* motion trail. By mounting the smart phone on this white cane, it is possible to expand the optic angle of camera by shooting multiple images (via in-built camera) in front along different directions (as shown in Fig. 7).

Feasibility: We introduce detection range (r) to characterize the expanded optic angle of in-built camera. Fig. 8 illustrates the schematic plot of camera in motion. Without loss of generality, we denote the optic angle of a given camera as α. Let β and γ be the rotation angle of the white cane to the left and right, respectively. By elementary geometry, we have $r \approx \alpha + \beta + \gamma$.

Figure 8. The wide-angle lens of camera.

Figure 9. Fig. 9(a) shows the distribution of rotation angle of the white cane during movement. Fig. 9(b) shows the cane swing traces of another visually impaired volunteer.

Given the focal length (f) and the CMOS size $(a \times b)$ of the camera, we can compute the optic angle α based on the image-forming equation $\alpha = 2 \cdot \arctan(\frac{b}{2 \cdot f})$. On the other hand, the rotation angle of canes varies from person to person yet converges for a specific user. As Fig. 9(a) shows, it maintains in a relative stable value (around $25°$) for a random chosen visually impaired volunteer. Therefore, one could combine the equations aforementioned to compute the extended optic range r.

As an example, we pick the best-selling smartphones in 2013 and compute their optical angles α based on the image-forming equation. Using the orientation reading of the rotation angle β and γ in Fig. 9(b), we then compute the detection range r and list it in Tab. 1. Compared with the standard optic angle of each smartphone, it is clear that the covering range of built-in camera has been significantly expanded.

Shooting point detection
As demonstrated in the previous section, the swing of a white cane significantly extends the optical angle of the phone camera. However, it entails subtle challenges to obtain effective photos for efficient zebra crossing recognition. On one hand, images captured during fast swing can be blurred which may render the zebra crossing almost unrecognizable. On the other hand, as cameras are more power-hungry than other sensory modalities such as inertial sensors and microphones,
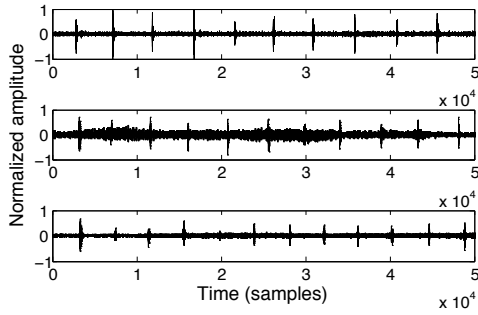
Figure 10. Acoustic profile of bumps subjects to blocks, visually impaired tracks and bituminous roads in the noisy environment.
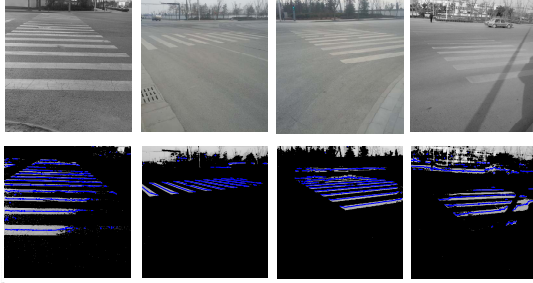

Figure 11. Examples of parallel lines extracted from the images.

it would quickly drain phone batteries to take photos continuously. Therefore, it is important to trigger photo shooting only when the camera remains stable.

To identify such stable moments (denoted as shooting points), we analyse the typical swing cycle of a white cane during walking, and exploit the built-in accelerometer to detect the stable moments. Fig. 6(c) portrays the acceleration profile along the Z axis during a cane swing cycle. As is shown, the user slightly raises the cane from one side (position a), swings across the middle (position b) with sharp acceleration, and starts to decelerate before the tip reaches the other side (position c). The cane tip is usually protected by a cortex rubber sleeve. Thus, when the tip contacts the solid ground, the ground repulses the tip gradually (the bumping phase). Moreover, since the tip tends to rest on the ground for a while before swinging backwards (possibly due to change of swing direction), it creates a short (yet sufficiently long) period when the camera stays still to take clear photos. Therefore, we choose these bump points as the shooting points.

The next challenge is to detect bump point and triggers the camera for image capturing in realtime. As the smart phone sways with the white cane, the acceleration readings (along the Z axis) will decrease first and then increase, after which it keeps stable during the bumping period. Thus, an intuitive way is to identify the stable period of acceleration readings for bump detection. However, due to the cushioning process and sensor noise, the acceleration readings during these periods may jump and fluctuate continuously (show in Fig. 6 (d)) which makes stableness-based method error-prone. Another possible way is to predict the bump period based on the repetitive pattern of swing process. However, as Fig. 6 (d) indicates, the period of each swing action varies significantly for the same people. As a result, it is difficult or even infeasible to capture the bumping event via prediction methods.
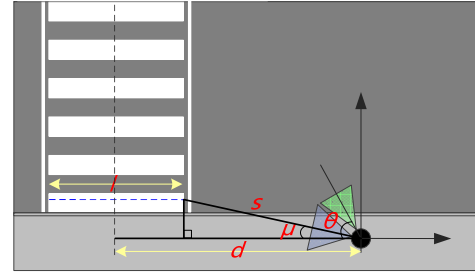

Figure 12. An illustration of rotation angle and lateral shift computation.

In *CrossNavi*, we leverage the sound effect of bumps for bump detection. The rationale behind is when the tip contacts the ground, it will generate whomps which potentially serves as an indicator for bump event. To demonstrate it, we record acoustic samples during white cane's movement with a smartphone in a noisy environment. The sampling rate is set to 8KHz. Fig. 10 portrays the acoustic profiles of bumps subjects to different kinds of roads, including blocks, tracks for the visually impaired and bituminous roads. As this figure reveals, all of these acoustic profiles show a remarkable peak when the tip contacts with the solid ground, indicating a bump happening. In this paper, we use a threshold-based method to detect bumps. When the normalized amplitude of acoustic sample is larger than 0.4 (empirically optimal in our experiments), we ascertain a bump happens, otherwise we delete it and go through the next sample.

Zebra crossing recognition

CrossNavi relies on computer vision techniques for zebra crossing recognition. There are plenty of algorithms [3, 12, 21, 22] that can successfully detect zebra crossings and we borrow the idea from [21] with little modification. In their method, the algorithm first executes Hough transformation on the input image to detect straight lines. As zebra signs are regular stripes and parallel to each other, the algorithm then picks out groups of nearly parallel lines and checks their concurrency as hypotheses for zebra crossings.

However, performing Hough transformation on the raw image may result in abundant straight lines which potentially expands the parallel line searching space and increases computational complexity. Instead of searching straight lines globally, *CrossNavi* separates the foreground from the background and executes Hough transformation on the foreground only. The intuition behind is that the contrast ratio of white stripes of zebra crossings is significantly higher than that of other objects in view, making zebra crossings outstanding and prone to be recognized as foreground. Fig. 11 shows running results of our algorithm. As the result shows, most of the white stripes are successfully categorized as foreground, which are then recognized as zebra crossings with our algorithm. Although our system is primarily designed to detect zebra patterns, it can be easily configured to recognize other mark patterns of crosswalks by other algorithms.

Relative position Inference

Once a zebra crossing is recognized, *CrossNavi* then leverages the sensor readings to infer its relative position with re-
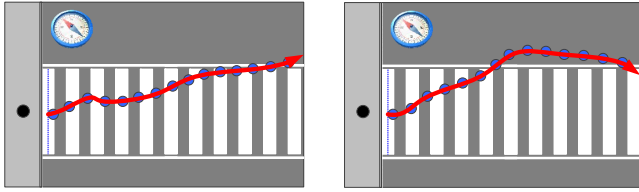
Figure 13. An illustration of motion trails of two visually impaired volunteers.

Figure 14. An illustration of user headings rectification estimation.

spect to the user. The position is characterized by a tuple: $< rotation\ angle(\theta), lateral\ shift(d) >$. Fig 12 illustrates the relative position between the user and zebra crossing.

The *rotation angle* is the angle that a user needs to rotate in order to be orthogonal to zebra crossings. Let μ be the slope of the stripe closest to the observer with respect to the horizon line of the image. Recall that photos are taken at the bump point. Taking the rotation angle of camera into consideration, and then based on the elementary geometry, we have:

$$\theta = \begin{cases} \mu + \beta, & \textit{if shoot on the left side of the body}; \\ \mu + \gamma, & \textit{if shoot on the right side of the body}; \end{cases}$$

The *lateral shift* is the displacement that is required by the visually impaired people to be at the center of the zebra stripe that is closest to the sidewalk. Let s be the distance between the user and the line segment closest to the user in pixel units. Given the calibration information of the camera (i.e., focal length f), as well as the orientation of the camera (captured already, i.e., α and β) and height from the ground, it is easy to compute the ground distance d in meters. Due to the page limitation, we omit the computation details and refer interested readers to [3]. Given this value, we further estimate the *lateral shift* d by the equation: $d = s \cdot cos(\mu) + \frac{l}{2}$, where l is the length of the zebra strip. In our experiments, we find that the localization accuracy of our methods is insensitive to the image size. Therefore, we choose to compress the image to a size of 486 x 648, which saves image processing latency to a great extent.

Safely Crossing it?
Once the visually impaired people are aligned to zebra crossings, they can safely enter the zebra patterns for road crossing. However, without the ability to refer to environmental cues such as the sun and tangible signs, the visually impaired people are prone to veer and walk in circles [23]. To demonstrate this, we recruited two visually impaired volunteers and asked them to walk inside zebra crossings (with size of 3.5m x 12.3m). Then, we recorded their motion trails with a video camera and plot them in Fig. 13. As this figure indicates, even under the guidance of white canes, both of these two volunteers veered intermittently along their movements. What is worse, we find that both of them surpassed the boarder of zebra crossings and nearly collided with stopping cars.

To navigate users to walk within the zebra crossing, a possible way is to use GPS-based navigating service. However, GPS service provides trivial information in maintaining headings, e.g., at crossroad, and it may simply state "go straight" or
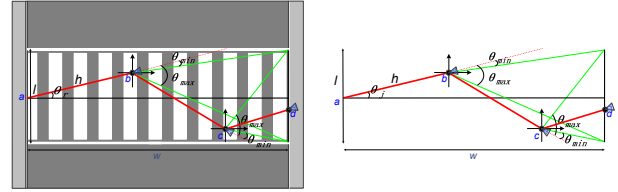
"turn left" without any further details. Moreover, the remarkable localization errors of GPS (around 5m for civil users and increases significantly in urban areas [2].) also blocks the widespread applicability of GPS in fine-grained road crossing navigation service. In *CrossNavi*, we explore combinations of multiple sensing dimensions, along with observations on human behavior to identify the walking headings of users, infer their movement trails, and rectify veering promptly.

Walking heading inference
The principle of walking headings inference module is based on such an observation: *when the visually impaired are walking along zebra crossings, the front face headings of their phones (bound on the canes) are always aligned with user headings, serving as a good indicator of their walking headings.* *CrossNavi* employs built-in compass to capture the front face heading of the smartphone and infers user headings as follows. It first partitions the user motion trial into sub trails T_j, with each lasting k swing cycles. In each swing cycle i ($1 \le i \le k$), *CrossNavi* detects two bumps (the cane contacts with the ground) and records compass readings o_i^l and o_i^r accordingly. The walking headings in each swing cycle i, therefore, approximates to $\frac{o_i^l + o_i^r}{2}$. After getting k continuous walking headings in sub trail T_j, *CrossNavi* computes the walking headings θ_j within this sub trail by the formula: $\theta_j = \frac{1}{k} \sum_{i=1}^{k} \frac{o_i^l + o_i^r}{2}$.

Basically, the parameter k is critical for the effectiveness of crossroad navigation. If k is too large, then the user may walk too long to surpass the boarder of zebra crossings. While if k is too small, it will take significant computation resources and drain the battery quickly. In the experiment part, we study the relationship between k and system performance.

Mitigating the sheering
Let χ be the headings of zebra crossings. Ideally, if $\theta_j = \chi$ for each j, we can ascertain that the user is walking straightly along zebra crossings. However, as aforementioned, the inability of recognizing visual signs renders the visually impaired veer intermittently during movement. Thus, *CrossNavi* should perceive the user motion trail and provide rectification warnings for safety.

In *CrossNavi*, we design an Incremental Walking headings Rectification scheme (IWR) based on geometry analysis. As Fig. 14 shows, suppose the user departs from a and veers in the first sub trail and reaches the point b. The walking distance h is easily computed by multiplying the average moving distance p (empirically set to 30cm) in each swing cycle and the number of swing cycles k. As cane movement has

---

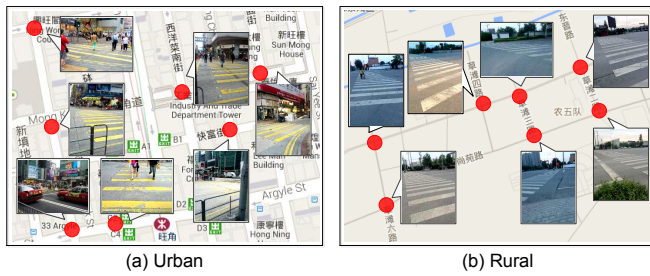[2]http://www.gps.gov/systems/gps/

Figure 15. An illustration of experimental field. The red dot denotes the locations where our experiments are conducted. We purposely blur the images for double-visually impaireded review.

similar repetitive pattern as human walking, similar to [18], here we also employ an auto-correlation based step counting method for moving distance computation. Given the moving distance h and user heading θ_j, *CrossNavi* computes the minimum and maximum angles θ_{min}, θ_{max} that the user should rectify by the following equations:

$$\theta_{min} = \arctan(\frac{\frac{l}{2} - h \cdot \sin(\theta_j)}{w - a \cdot \cos(\theta_j)}) \qquad (2)$$

$$\theta_{max} = \theta_j + \arctan(\frac{\frac{l}{2} + h \cdot \sin(\theta_j)}{w - a \cdot \cos(\theta_j)}) \qquad (3)$$

In these equations, l and w are the length and width of zebra crossings which are open sources and accessible in public transportation administration web sites. θ_{min} and θ_{max} indicate the necessary rectification angles for users to follow. After computing θ_{min} and θ_{max}, *CrossNavi* averages these two values and conservatively guide user to rectify his/her walking headings with the angle of $\frac{\theta_{min}+\theta_{max}}{2}$.

In practice, as human brain is incapable of guiding the body to turn certain angles, it is impossible to follow the rectification angles manually without assistance. In the current version of *CrossNavi*, we simply ask users to rotate their canes and leverage compass readings to measure rectification angle. For example, in our implementation, *CrossNavi* broadcasts short-term beeps to indicate the proper rotation angle of the cane. Users then rectify their headings with this angle.

IWR continuously captures the readings of compass, microphone and acceleromater, and infers user headings in each sub trail. It computes user's walking distance and rectifies veering until the user reaches the other side of the road.

USER INTERFACE
We implement *CrossNavi* on Android 4.3 Jelly Bean operating system. The current version consists of about 1100 lines of code and provides auditory feedbacks to assist the user to cross the road[3]. The phone running *CrossNavi* can be easily

---

[3]Once a blind person approaches the crossroad, he can hear the beeper signals, and thus will know the crossroad is not far away. Hence it is unnecessary to give acoustic indication to the user when it is about to approach the zebra crossing. A friendly reminder from the beeper detection module may help remind the user to get prepared and concentrate on crossing the road, and we will consider it in future versions of our system.

mounted on the cane by using a phone holder which is easily accessible on the market. When *CrossNavi* locates zebra crossings or when the user deviates from the desired heading, feedback is produced to navigate/rectify users. With this simple interface, the users can determine the location of zebra crossings, align themselves to it, and rectify veers for safety. In the current version, the users are required to launch and shut down *CrossNavi* manually, which may be difficult for the visually impaired. However, we believe, with the advance of Speech Recognition technique such as *Siri*[4] and *S Voice*[5], it is possible to operate *CrossNavi* via voice control message, which will alleviate user's overhead.

EXPERIMENTAL RESULTS

Experimental Setup
We implement *CrossNavi* APK on three types of smartphones – Samsung Nexus I9250, Motorola MT788, and Xiaomi 2S. All types of phones are equipped with necessary sensors. The Samsung Nexus I9250 is equipped with 1GB RAM and dual-core 1.5GHz processor; the Motorola MT788 has 1GM RAM and single-core 2.0 GHz processor; the Xiaomi 2S has 2GB RAM and quad-core 1.2 Ghz processor.

To give a comprehensive evaluation of *CrossNavi*, we recruit three visually impaired volunteers and ask them to use our application for crossroad navigation. The dataset for evaluation is collected over 200 times of road crossings from 14 crossroads (spanning from noisy urban areas to relatively quiet rural areas) over totally 2.1km walking distances (shown in Fig. 15). Each experiment is conducted multiple times and the results are averaged for final evaluation. Although the visually impaired would be more familiar with the spatial layouts after crossing a specific crosswalk multiple times, we noticed that some participants walked more relaxed after several times of crossing at certain simple spatial layouts, yet still occasionally tended to veer off. Hence our system can still be useful and timely remind the relaxed users even if they may have been familiar with the spatial layouts.

Micro Benchmarks
Performance of crossroad detection
In this trail of experiments, we collect acoustic samples in 14 different sites and examine the effectiveness of the crossroad detection module. Fig. 16 illustrates the relationship between the user-to-crossroad distance and detection success rate. As it indicates, the detection success rate increases when user approaches the crossroad in both rural and urban areas. In particular, when the user is far away from the crossroad, e.g., more than 10 meters away from the crossroad, the beep signal attenuates to an indistinguishable level for crossroad detection. However, as the user approaches the crossroad, the detection success rate boosts rapidly and finally climbs to 80% and 93% in urban and rural areas, respectively. It clearly demonstrates that *CrossNavi* is insensitive to acoustic noises and is able to detect crossroad promptly.

Further, we investigate the detection success rate under different traffic conditions. The results are shown in Fig. 17.
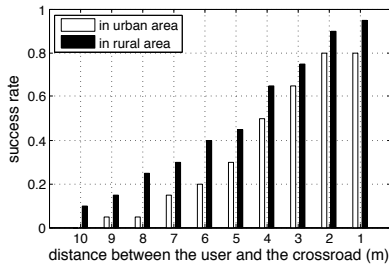
---

[4]http://www.apple.com/hk/en/ios/siri/
[5]http://www.samsung.com/global/galaxys3/svoice.html

Figure 16. distance between the user and the cross road vs. detection success rate.
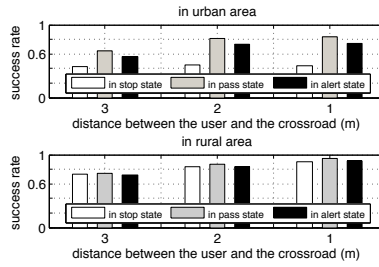

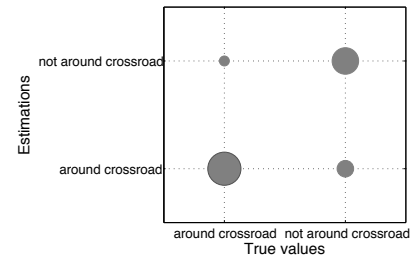**Figure 17. detection success rate under different traffic conditions.**
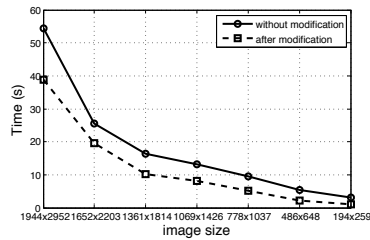

Figure 18. crossroad detection confusion matrix.

Figure 19. latency vs. image size.

Figure 20. distance between the user and the cross road vs. angular offset.

Figure 21. distance between the user and the cross road vs. ranging errors.

As these bar charts indicate, in urban areas the detection success rate varies significantly under different traffic conditions. Nearly 80% crossroads are identified in pass state (green light) when the user-to-crossroad distance is within 2m. However, this figure declines by approximately 5% in alert state and 15% in stop state at every measurement point. This is coherent with our illustrative experiment (Fig. 5) that the beep signals of stop state are relative indistinguishable from ambient noises. Nevertheless, *CrossNavi* still achieves 65% success rate in this state. As for rural cases, the quiet environment renders *CrossNavi* much superior than in urban areas. Specifically, when the user is 1 meter away from the crossroad, *CrossNavi* achieves as high as 92% success rate in pass state, 90% in alert state, and 89% in stop state.

Fig. 18 presents the confusion matrix showing how false positive/negative crossroad detection is distributed. For each intended subject along the X-axis, the size of the circle reflects the proportion of detection results in the corresponding Y-axis. On the whole, *CrossNavi* achieves competitive detection results, with an accuracy of over 90% true positive success. 13% are processing failures, in which cases *CrossNavi* mistakes the street as crossroad and invokes the zebra crossing localization module. Nevertheless, the false positive rate is relative small (below 10%), which demonstrates the effectiveness of the acoustic-based crossroad detection module.

Performance of zebra crossing localization
To demonstrate the effectiveness of the zebra crossing localization scheme, we collect images under different weather conditions, including a bright sunny day and a cloudy rainy day. We first examine the effectiveness of the zebra crossing detection scheme by examining the image processing latency before and after we distinguish the foreground under different image size settings. All the running time is calculated by averaging the processing latency on three kinds of smartphones. As Fig. 19 indicates, the image processing latency is proportional to the image size for both these cases, yet our method
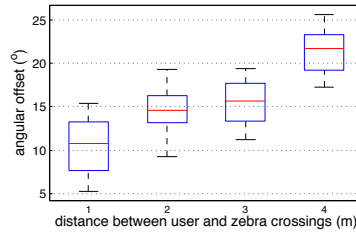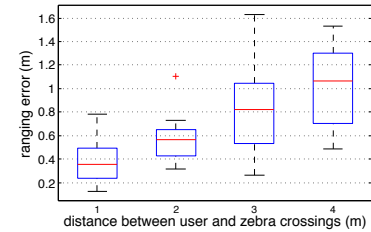
considerably surpasses the unmodified CV technique, requiring less than $\frac{2}{3}$ of time for processing relative large images. Although the gap between these two methods decreases as the image becomes smaller, our method still saves about 1.2s and 0.3s when the image size is 486x648 and 194x259.

We next examine the effectiveness of the relative position inference module. It includes the rotation angle inference and lateral shift estimation. We first investigate the relationship between the user-to-crossroad distance and the rotation angle inference errors (termed as angular offset). As Fig. 20 shows, the angular offset increases with the rising of user-to-crossroad distance. When the user is in the vicinity of the crossroad (about 1 meter away), *CrossNavi* achieves remarkable inference accuracy, with an angular error of $11°$ on average. The mean value of angular offsets then increases steadily when the user stays further from the crossroad, and reaches $23°$ when the user is 4 meters away from the crossroad.

We also investigate the relationship between the user-to-crossroad distance and lateral shift ranging accuracy. As the box chart illustrates (Fig. 21), initially, when the user is 1 meter away from the crossroad, the ranging error is confined in a relatively small scope, with an error of 0.37m on average. As the user moves further to the crossroad, the ranging accuracy degrades gradually. The distribution of ranging errors tends to be more sparsely. Once the user is 4 meters away from the crossroad, the ranging errors climbs to 1.5m. Nevertheless, *CrossNavi* still achieves promising ranging accuracy, with 1.2m ranging error on average.

Performance of navigation
To achieve a comprehensive result, each person is required to cross the zebra crossings (with a size of 2.5m x 22.5m) ten times. During each testing, we use an HD camera to record user trail as ground-truth.

Fig. 22 gives a snapshot of user locations when *CrossNavi* broadcasts veer rectification warnings under various k set-
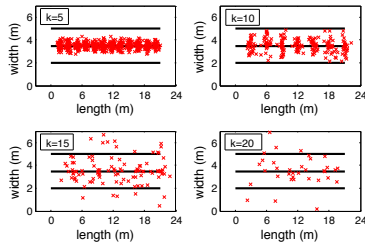
Figure 22. location distribution under different k settings.
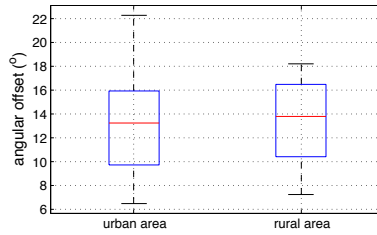
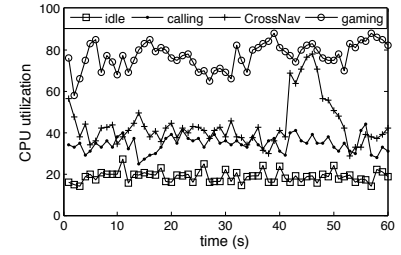

Figure 23. angular offset in different area.

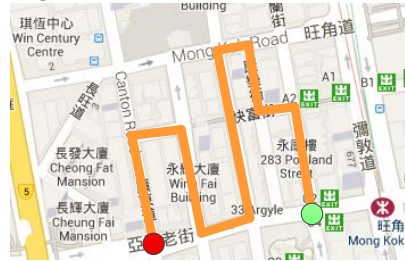Figure 24. CPU utilization under different operations.

Figure 25. user trace for energy consumption detection. We purposely blur the images for double-visually impaireded review.
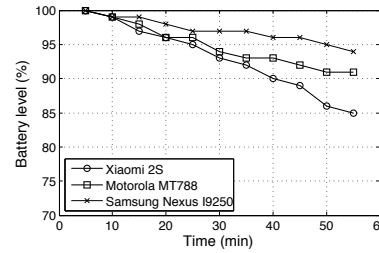

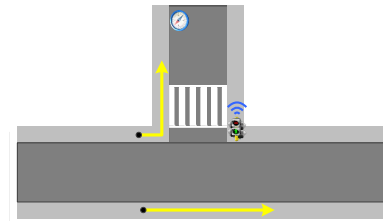Figure 26. energy consumption of *CrossNavi* over three smartphones.

Figure 27. crossing T-junction.

tings. The black lines inside the figure are the boarder of zebra crossings. As aforementioned, the parameter k is critical for the effectiveness of crossroad navigation. If k is too large, then the user may walk too long to surpass the boarder of zebra crossings. While if k is too small, it will take significant computation resources and drain the battery quickly. As this figure indicates, when k is relative small (e.g., k=5), *CrossNavi* continuously reports veer rectification warnings for user heading correction. As a result, users are notified promptly, thereby are always walking within the center part of zebra crossings. As we gradually increases k, warning frequency drops significantly, which results in significant latency in veering rectification message delivering. Such latency renders the user deviate from routes and even surpasses the boarder line of zebra crossings (when k=15, and k=20). Suggested by this figure, we configure k to be 10 in the following experiments.

Further, we examine the accuracy of rectification angles provided by *CrossNavi*. The results are shown in Fig. 23. As this figure shows, when the user is in rural areas, the angular offsets are within the range of 7.5° to 18°. When the similar experiment is conducted in urban areas, the range of angular offsets extends gradually, with the maximum offset over 22° and the minimum offset below 7°. The experiment result indicates that our system tends to suffer limited compass induced errors. This is because when the user is walking, the cane held in hand just waves around in a pattern similar to that in Figure 8. It is well-known that holding the phone in front and waving it around as in Figure 8 is an efficient way to rectify the compass. Moreover, as *CrossNavi* computes the walking heads every 10 swing cycles, such short period prevents the expansion of angular offset.

System Overhead

CPU utilization

We launch *CrossNavi* on a Samsung Nexus I9250 (dual core) smartphone, and write a logger to record the CPU utilization trace. The results (average utilization of CPU1 and CPU2) are shown in Fig. 24. For comparison, we also record the CPU utilization trace of this phone in its idle (no program is running except for system processes), calling, and gaming state. As this figure indicates, when the phone is in idle state, the CPU utilization maintains in a low and relative stable level, with a little fluctuation over 20%. The utilization index then increases slightly and hovers at 39% when the user is making phone calls. Further, as we launch the *CrossNavi* for crossroad navigation, the CPU activities gradually accelerate, achieving an utilization of 43% on average over crossroad detection process ($<$40s) and 57% as the camera is triggered for zebra crossings localization ($>$40s and $<$52s). The highest utilization appears in gaming state, during which the CPU utilization peaks 89% and maintains in a relative high level during the whole process. The results are clearly not perfect, but, it still shows that *CrossNavi* is compatible with the normal use of a phone (calling), and achieves significantly lower resources compared with mainstream games. We believe with the advance of more powerful processors (i.e., quad core), the CPU utilization of *CrossNavi* will decreases dramatically to an ignorable level.

Energy overhead

Let us recall the key steps in *CrossNavi* from the energy usage perspective. Initially, *CrossNavi* fires the microphone to sense the ambient acoustic data. If the crossroad is detected, it automatically triggers the camera and orientation sensors to locate zebra crossings. After the user is aligned to zebra crossings, *CrossNavi* shuts down the orientation sensors and initiates the accelerometer and the compass to infer user trail for veer rectification. Thus, *CrossNavi* attempts to conserve energy in two ways: first, it keeps the sensors activated only

around the right time; second, it adopts low-power sensors to reduce energy consumption on localization process.

For a fair evaluation, we drain the energy of these phones and then charge each one with an equal period (10 hours). The volunteers are asked to go along the planned route (Fig. 25) with our *CrossNavi* activated in these phones. Fig. 26 shows the phone's battery duration as a function of execution time of *CrossNavi*. As it indicates, the energy consumption shows a similar trend yet differs in amount for all of these three phones. Specifically, *CrossNavi* consumes negligible energy (below 1% on average) for most sampling periods. With time passing by, the cumulative energy consumption increases gradually, and finally ends up with 6% for Samsung Nexus I9250, 9% for Motorola MT788, and 15% for Xiaomi 2S. This figure clearly demonstrates that *CrossNavi* is energy efficient, thus is applicable for mainstream smartphones.

DISCUSSION

Manual Trigger and Route Planning: *CrossNavi* is tailored as an aid of safe road crossing, and its current version expects the user to manually launch the app when he intends to pass a crosswalk. However, it happens that the user may forget to close it afterwards. Consequently, due to its beep-trigger mechanism and extended sensing range, *CrossNavi* may be falsely triggered when the user approaches a crosswalk, yet does not intend to pass it (Fig. 27). Manual input of user plans in advance would help, but since *CrossNavi* can be easily integrated with route planning services [26, 27], we envision *CrossNavi* to know the user's intention at each intersection automatically to avoid such false alarms and provide more comprehensive and convenient outdoor navigation services for the blind.

Enhancing Veer Detection: In purpose of energy efficiency, *CrossNavi* employs inertial sensors to detect veers. While these modalities significantly reduce energy consumption and provide sufficient accuracy for 2-lane roads, the intrinsic measurement error accumulation of low-cost inertial sensors may drift user trial estimation on relatively wide roads (e.g. 16-lane) without re-calibration. As our on-going work, we are exploring occasional visual references via CV techniques to rectify user trail estimation with minimal extra energy consumption. Although the CV-based technique is sensitive to the light condition, it is still suitable for zebra crossing detection since the crossroad is usually configured with the streetlight which can enhance the illumination density around.

RELATED WORKS

Wayfinding systems: Great efforts have been paid on investigating Geographic Information System (GIS) and Global Positioning System (GPS) for visually impaired navigation. Golledge et al. [9] pioneered this field by introducing a wearable Personal Guidance System (PGS). This system consists of a GPS receiver, a notebook computer and a fluxgate compass for point-to-point outside navigation. Wilson et al. [25] introduced a GPS-based system which leverages camera and light sensors to detect blocks on road for safe outdoor navigation. Latter on, Angin et al. [5] proposed a mobile-cloud collaborative approach for visually impaired navigation by ex-

ploiting the powerful computational capability of the Cloud as well as the wealth of location-specific resources available on the Internet. Further, researchers also try to leverage advanced localization devices for visually impaired navigation, including ultrasonic based systems [19, 6, 15], and HD-camera based systems [14, 16].

Anti-veering systems: Several researchers also focused on the design of anti-veering systems for crossroad navigation. David [10] presented a gyroscope-based anti-veering system to correct human veering via speech cues. Panels et al. [17] built up an anti-veering system on commercial portable devices. It takes advantage of the built-in sensors on smartphones. However, both of these works assume that the visually impaired are already aligned with zebra crossings, which is difficult, or even infeasible for the visually impaired without any guidance. There are also pioneer works on leveraging RFID technology for user veering rectification. Sesamonet project [24] equipped an RFID reader on the white cane to sense tags embedded in the ground to guide visually impaired people to walk along a safe path. Alghamdi et al. [4] combined RFID navigation with QR-code, and proposed a hybrid anti-veering system to assist visually impaired people to reach their destination of interest via the shortest path.

These systems can provide partial assistance in crossroad navigation. However, all of them are overly dependent on costly hardware, including ultrasonic emitter, tactile input devices, as well as pre-deployment of RFID tags, which raises the important issues of price and confines the usability worldwide. By utilizing equivalent technology integrated in commercial smartphones, *CrossNavi* aims to provide a more compact and cost-effective solution. This we believe is one of the major contributions of this paper, and it potentially opens the door to wide-scale deployment.

CONCLUSION

Designing crossroad navigation systems for the visually impaired is critical yet challenging. Existing solutions either rely on costly hardware or require explicit user participation. In this paper, we present *CrossNavi*, a smartphone-based visually impaired navigation system which helps to detect crossroad, locate zebra patterns, and monitor the user within the width of the crosswalk when passing the road. Limited by sixpenny phone sensors, the ranging and localization accuracy is still undesirable. However, we believe that *CrossNavi* explores the possibility of making crossroad navigation as easy as possible, taking a significant step towards pervasive visually impaired navigation services.

ACKNOWLEDGEMENT

REFERENCES

1. I-Cane. http://www.i-cane.org/en/865.

2. World Health Organization. http://www.who.int/mediacentre/factsheets/fs282/en/.

3. Ahmetovic, D., Bernareggi, C., and Mascetti, S. Zebralocalizer: Identification and localization of pedestrian crossings. In *Proceedings of the 13th International Conference on Human Computer Interaction*, CHI'11 (2011), 275–284.

4. Alghamdi, S., van Schyndel, R., and Alahmadi, A. Indoor navigational aid using active rfid and qr-code for sighted and blind people. In *Intelligent Sensors, Sensor Networks and Information Processing, 2013 IEEE Eighth International Conference on*, IEEE (2013), 18–22.

5. Angin, P., Bhargava, B., and Helal, S. A mobile-cloud collaborative traffic lights detector for blind navigation. In *Mobile Data Management (MDM), 2010 Eleventh International Conference on*, IEEE (2010), 396–401.

6. Bousbia-Salah, M., Redjati, A., Fezari, M., and Bettayeb, M. An ultrasonic navigation system for blind people. In *Signal Processing and Communications, 2007. ICSPC 2007. IEEE International Conference on*, IEEE (2007), 1003–1006.

7. Chan, K.-Y., Manduchi, R., and Coughlan, J. Accessible spaces: Navigating through a marked environment with a camera phone. In *Proceedings of the 9th International ACM SIGACCESS Conference on Computers and Accessibility*, Assets '07 (2007), 229–230.

8. Dakopoulos, D., and Bourbakis, N. G. Wearable obstacle avoidance electronic travel aids for blind: a survey. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on 40*, 1 (2010), 25–35.

9. Golledge, R. G., Klatzky, R. L., Loomis, J. M., Speigle, J., and Tietz, J. A geographical information system for a gps based personal guidance system. *International Journal of Geographical Information Science 12*, 7 (1998), 727–749.

10. Guth, D. Why does training reduce blind pedestrians veering? *Blindness and brain plasticity in navigation and object perception* (2007), 353–365.

11. Guy, R., and Truong, K. Crossingguard: Exploring information content in navigation aids for visually impaired pedestrians. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '12 (2012).

12. Ivanchenko, V., Coughlan, J., and Shen, H. Detecting and locating crosswalks using a camera phone. In *Computer Vision and Pattern Recognition Workshops, 2008. CVPRW '08. IEEE Computer Society Conference on* (2008), 1–8.

13. Loomis, J. M., Marston, J. R., Golledge, R. G., and Klatzky, R. L. Personal guidance system for people with visual impairment: A comparison of spatial displays for route guidance. *Journal of visual impairment & blindness 99*, 4 (2005), 219.

14. Mann, S., Huang, J., Janzen, R., Lo, R., Rampersad, V., Chen, A., and Doha, T. Blind navigation with a wearable range camera and vibrotactile helmet. In *Proceedings of the 19th ACM international conference on Multimedia*, ACM (2011), 1325–1328.

15. Mihajlik, P., Guttermuth, M., Seres, K., and Tatai, P. Dsp-based ultrasonic navigation aid for the blind. In *Instrumentation and Measurement Technology Conference, 2001. IMTC 2001. Proceedings of the 18th IEEE*, vol. 3, IEEE (2001), 1535–1540.

16. Pai, C.-J., Tyan, H.-R., Liang, Y.-M., Liao, H.-Y. M., and Chen, S.-W. Pedestrian detection and tracking at crossroads. *Pattern Recognition 37*, 5 (2004), 1025–1034.

17. Panels, S. A., Varenne, D., Blum, J. R., and Cooperstock, J. R. The walking straight mobile application: Helping the visually impaired avoid veering. *Auditory Display, 2013 International Conference on*.

18. Rai, A., Chintalapudi, K. K., Padmanabhan, V. N., and Sen, R. Zee: zero-effort crowdsourcing for indoor localization. In *Proceedings of the 18th annual international conference on Mobile computing and networking*, ACM (2012), 293–304.

19. Ran, L., Helal, S., and Moore, S. Drishti: an integrated indoor/outdoor blind navigation system and service. In *Pervasive Computing and Communications, 2004. PerCom 2004. Proceedings of the Second IEEE Annual Conference on*, IEEE (2004), 23–30.

20. Ross, D. A., and Blasch, B. B. Wearable interfaces for orientation and wayfinding. In *Proceedings of the fourth international ACM conference on Assistive technologies*, ACM (2000), 193–200.

21. Se, S. Zebra-crossing detection for the partially sighted. In *Computer Vision and Pattern Recognition, 2000. Proceedings. IEEE Conference on*, vol. 2, IEEE (2000), 211–217.

22. Sebsadji, Y., Tarel, J. P., Foucher, P., and Charbonnier, P. Robust road marking extraction in urban environments using stereo images. In *Intelligent Vehicles Symposium (IV), 2010 IEEE* (2010), 394–400.

23. Souman, J. L., Frissen, I., Sreenivasa, M. N., and Ernst, M. O. Walking straight into circles. *Current Biology 19*, 18 (2009), 1538–1542.

24. U, B, C., C, M, M., and E, S. Sesamonet 2.0: improving a navigation system for visually impaired people. In *Ambient Intelligence (AmI), 2010 1th International Conference on* (2010), 248–253.

25. Wilson, J., Walker, B. N., Lindsay, J., Cambias, C., and Dellaert, F. Swan: System for wearable audio navigation. In *Wearable Computers, 2007 11th IEEE International Symposium on*, IEEE (2007), 91–98.

26. Yang, Z., Wu, C., and Liu, Y. Locating in fingerprint space: Wireless indoor localization with little human intervention. In *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*, Mobicom '12 (2012).

27. Zhou, P., Zheng, Y., Li, Z., Li, M., and Shen, G. Iodetector: A generic service for indoor outdoor detection. In *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*, SenSys '12 (2012).