A Smartphone-based Obstacle Sensor for the Visually Impaired

En Peng, Patrick Peursum, Ling Li, and Svetha Venkatesh

Department of Computing, Curtin University of Technology, Perth, Australia
{e.peng,p.peursum,l.li,s.venkatesh}@curtin.edu.au

Abstract. In this paper, we present a real-time obstacle detection system for the mobility improvement for the visually impaired using a handheld Smartphone. Though there are many existing assistants for the visually impaired, there is not a single one that is low cost, ultra-portable, non-intrusive and able to detect the low-height objects on the floor. This paper proposes a system to detect any objects attached to the floor regardless of their height. Unlike some existing systems where only histogram or edge information is used, the proposed system combines both cues and overcomes some limitations of existing systems. The obstacles on the floor in front of the user can be reliably detected in real time using the proposed system implemented on a Smartphone. The proposed system has been tested in different types of floor conditions and a field trial on five blind participants has been conducted. The experimental results demonstrate its reliability in comparison to existing systems.

Key words: obstacle detection; visually impaired; real-time; monocular vision

1 Introduction

Mobility assistance is desperately needed by the visually impaired because obstacles can cause injuries. Consultation with an expert group [16] has highlighted that available mobility assistances are dissatisfactory to the blind for various reasons. Thus this research aims to fill this gap by building a low-cost, non-intrusive and simple system for blind navigation.

Several methods already exist for providing mobility assistance for the visually impaired, ranging from human helpers to modern devices. A human guide is the most intelligent assistant, but is not always readily available. On the other hand, a white cane is the most readily available mobility tool but is very intrusive and makes the blind person highly conspicuous. A guide dog[16] is a good choice because they are loyal and less intrusive but is as expensive as a car and each dog is only capable of providing assistance for a few years due to a long rigorous training process. Some blind people develop the echolocation ability to gain a measure of self-sufficiency in their mobility. Echolocation is the ability to sense objects by listening for echoes - i.e. human based sonar. However, echolocation is a difficult skill to master and is not able to detect small objects.

Due to the issues with these traditional mobility solutions, a range of technological solutions have been developed, commonly referred to as electronic travel aids (ETAs). An ETA based on optical triangulation, e.g. LaserCane [1], is accurate but has a very narrow scan beam and costs more than five times the price of a typical mid-range Smartphone [19]. An ETA based on acoustic triangulation, e.g. MiniGuide [17] and GuideCane [14], can find open areas and can be as cheap as a mid-range Smartphone. However, a MiniGuide is unable to detect low-height objects on the floor and the GuideCane is very bulky. An ad-hoc ETA based on stereo vision, e.g. a Minoru 3D webcam [18] plus a notebook computer running the vOICe software, can recover full depth map but such a system is not popular due to the lack of a stereo camera in many portable devices.

Despite the existence of various mobility assistants, it is still difficult to find an assistant that is ultra-portable, low cost, non-intrusive, and able to detect on-floor obstacles. Motivated by this gap, we propose a solution embedded on a mobile phone platform since many visually impaired people already make use of Smartphones due to their many useful features (camera, optical character recognition, text-to-speech, voice command, GPS navigation, etc.) By utilizing the embedded camera on the Smartphone, it is possible to make the system non-intrusive through computer vision techniques. The main challenge here is how to detect on-floor obstacles through computational efficient computer vision techniques that can run in real time on a Smartphone. There are similar techniques for autonomous navigation employed in robots or autonomous vehicles, but they are computationally expensive [12], restricted to certain environments e.g. roads [11], and/or requiring a training phase to learn the scene's characteristics and thus not portable to different environments [10]. The proposed research seeks to effectively make use of perspective projection knowledge and fast computer vision techniques (color histograms and edge detection). By combining these techniques, the proposed system is able to detect most on-floor obstacles in real time on a Smartphone without any prior training/adapting stages. The system is compared to two other systems, the edge based approach of Taylor et al. [12] and the color histogram based approach of Tan et al.[11], which are also efficient enough to run on a Smartphone. Experiments and field trials have shown that the proposed system has better accuracy whilst being more computationally efficient.

The remainder of the paper is organized as follows: Section 2 introduces the related work. Section 3 explains the proposed on-floor obstacle detection system. In Section 4, the implementation on the Smartphone and the usage are discussed. The evaluation results of the system are presented in Section 5 before the conclusions are made in Section 6.

2 Related Work

As we have decided to utilize the embedded camera on Smartphones, we will review existing approaches that could lead to the obstacle detection using a monocular camera: three-dimensional structure reconstruction, two-dimensional motion analysis, and recognition.

Three-dimensional structure reconstruction from monocular images is probably the most geometrically intuitive way to identify obstacles: the relative distance between the camera and each object can be easily computed for each frame and thus any obstacle can be determined. This problem has been investigated by many researchers for decades. Most of the efforts have been on stationary scene reconstruction, while a few researchers are interested in the scene with moving objects. Among existing approaches, traditional structure from motion (SFM) is the most well known. SFM approach [8] first finds the correspondence between images, and then initializes the 3D scene using matches that satisfy the epipolar constraints while does not related by a homography, and finally other frames are added to refine the scene using bundle adjustment. This approach can be accurate but the computation cost is very high. Monocular simultaneous localization and mapping (MonoSLAM) [2] is a recent approach that can be classified as a real-time online SFM using a monocular camera. MonoSLAM aims to localize the camera by simultaneously recovering the 3D structure of the landmarks in the scene. This approach requires a high frame rate so that landmarks can be tracked within a small search window. The number of landmarks is also limited in this approach in order to achieve real-time performance. When there is a moving object in the scene, traditional SFM will fail to work. In such case, some researchers proposed that the correspondence between images is clustered into different groups [13], as they are related with different fundamental matrixes. However, the research in this area is in the preliminary stage. Furthermore, it is still impratical to implement these SFM algorithms (including MonoSLAM) on the mobile phone due to low computation power.

Two-dimensional motion analysis is another way to identify the obstacles, which does not involve 3D reconstruction. Instead, the 3D movements of objects or the time-to-contact may be estimated. This approach first computes the 2D optical flow motion from an input video, then analyze it using different criteria: for example, by simply comparing the sum of 2D motion field between the left and right halves of the image, obstacles can be avoided by turning to the side with smaller sum (balance strategy) [3]. When the camera is moving straight forward and smoothly in a constant speed, the time-to-contact can be estimated by utilizing the focus of expansion and the divergence from the focus of expansion [9]. If the camera is not constantly moving and its motion is unknown, it becomes very hard to determine the depth from the 2D flow, because it could be projected from different 3D flow.

Through recognition based on knowledge context [7], shape [5], or color [6], it is possible to identify the obstacles. The knowledge on the context or shape normally requires massive training efforts and would be unsuitable to reliably recognition. In contrast to the context/shape, the color information is easier to learn and recognize. The existing methods employ the color information mainly in three ways:

1. Mathematically-defined color models: the colors in the desired regions are modeled with a Gaussian or a Mixture of Gaussians (MoG) models - the training process may takes from as short as a few minutes (context-aware) to as long as several hours (manual) [10]. However, these models are based on snapshots of the scene as it is during training, and so are not applicable if the person moves to a new scene;
2. Color histograms: by sacrificing memory, color histograms represent the desired colors in the simplest form. Due to its simplicity, the computation is very efficient and so can be continuously recalculated which is ideal for application in mobile platform. Tan et al. [11] assume that the small area in front of a vehicle is always clear and the color histogram is sampled. By maintaining a set of such color histograms for the clear road and a histogram for the background determined in previous frame, it calculates the probability of a pixel belonging to road by distance measurements on histograms. After linear combination of the previous and current probability of the same pixel, it can be determined if a pixel belongs to a road or the background. In comparison to mathematical models, histograms directly represent the true (empirical) color distribution rather than fitting it to a (potentially inaccurate) model [10, 11, 15];
3. Edges: when the desired region is known as consisting of a uniform color, its appearance may consist of different colors due to illumination variances on that region or due to camera artifacts. In most cases, the transitions between these colors are smooth. Therefore, edges, or unsmooth color transitions, often indicate borders between objects. Taylor et al.[12] assumes that floors consist of uniform colors and a simple seeded region growing method is used based on normalized red (r), normalized green (g), hue (h) and intensity (i) channels to expand the safe region for navigation.

Among these related works, only limited approaches are possible to run on a device with low-computational power, such as mobile phones, without prior-learning: color histogram based recognition and edge based recognition. The proposed system will hence build on these approaches. The proposed system will be introduced in the next section.

3 Proposed System

Similar to the related work on recognition based color histograms and edges, the proposed system assumes that a small region of the floor in front of the user is safe. In addition, the proposed system assumes that the user is able to maintain the Smartphone at a certain tilt angle, e.g. 45° , all the time so that the floor in front of the user is always visible in the image. With these two assumptions, the idea behind the proposed system is based on the image region that is assumed to be a clear floor region and finding anything that looks different from it. We compute the distance from the user, after which the safe path is found.

The proposed system will be described in three parts: image region of interest, initial histogram for safe region, and safe path finding.

3.1 Image Region of Interest

Many existing methods process all pixels in an image or do so in the worst case[12]. In fact, not all pixels in the image need to be analyzed for the purpose of obstacle detection in our case. As a person normally walks straight ahead, a rectangular floor region in front of him/her is of most concern whether there is an obstacle. Therefore, only the projected image region of that rectangular region is of interest. While the exact depth of a pixel representing the floor can be accurately computed using perspective projection knowledge through a homography matrix, by assuming the camera is pointing forward with a tilt angle downwards and no roll angle, an input image can be simply explained as follows: the pixels at the lower region of the image represents the floor that is closer to the user, and a rectangular region in front of the user will appears like a trapezoid because the object looks smaller when it is farther under perspective projection.

Therefore, a proper trapezoid image region can be computed based on known tilt angle and focal length of the camera and can be defined on the image as the *image region of interest*. Thus, the maximum number of pixels needed to be processed can be decreased.

3.2 Initial Histogram for Safe Region

The small region of the floor in front of the user appears at the bottom of the image and is assumed safe. The knowledge of the safe floor can be learned from that image region. Similar to Tan et al.'s system, a color histogram will be employed to represent this image region.

In order to build a color histogram, the color space must be firstly chosen. Gevers and Smeulders [4] have evaluated several popular color spaces to determine invariance to viewpoint, object geometry and illumination for recognizing multicolored objects. However, there is no single color space that is most appropriate under all circumstances. If the conditions across images are controlled, the *RGB* color space is the most appropriate for recognizing multicolored objects although it has worse performance in terms of discriminative power due to its sensitivity to varying image conditions. As for an embedded camera on a Smartphone handheld by a human, the image changes could be frequent due to auto white balance and auto exposure of the camera affected by the environment where it is pointed to. However, different regions within a single image should have the same white balance level and have the same exposure time. Hence, we decide to build a histogram for each frame and choose the simplest *RGB* color space because it is the most appropriate when the imaging conditions is constant according to [4].

To build a traditional *RGB* histogram from a small region in a tiny image is not robust. The number of pixels in each bin could be very small and thus the color distribution could not be well represented. Bootstrapping is a way to increase the number of samples - but it will increase computational cost and may introduce incorrect samples. We therefore adopt a binary *RGB* histogram with

16^3 bins, which does not concern pixel counts in each bin. The corresponding bin of a pixel in the sampling region is labeled as *true*. In addition, the neighboring 9 bins of this bin are also labeled as true to accommodate small variations to the color. Thus, a binary *RGB* histogram is initially built for the safe region. Next, we will introduce the safe path finding process based on this initial histogram for the safe region in the *image region of interest*.

3.3 Safe Path Finding

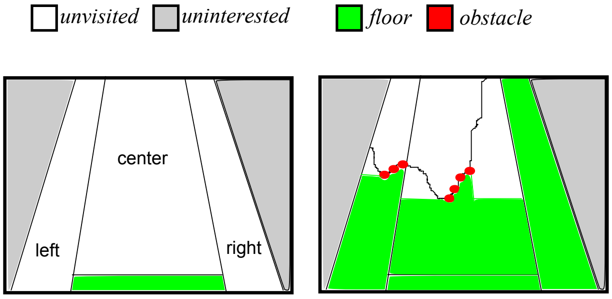
We define four states of a pixel: *uninterested*, *unvisited*, *floor*, *obstacle*. Since only the *image region of interest* is of concern, we label all pixels in that region as *unvisited* and the rest as *uninterested*. As the small region at the bottom of the image is considered as the floor, we label all pixels in this small region as *floor*. Our remaining task is to process any pixel labeled *unvisited*, if that pixel is of interest.

A classifier has to be built with the prior knowledge of both classes (*floor* and *obstacle*). In Tan et al.'s system, in addition to the histogram of the road (*floor*), the histogram of the background (*obstacle*) is also built based on the pixels classified as "background" in the previous frame. The knowledge of two classes enables them to classify a pixel. After linear combination of the current frame and the previous frame on the probability of each pixel, the largest region of pixels classified as "road" is considered as road while other pixels are considered as background. However, their system could not be simply applied in our case because the background information in the previous frame is unreliable due to various factors such as lighting changes, scene changes caused by rapid hand movements.

Since a histogram for *obstacle* could not be reliably obtained, thresholding is the most appropriate approach based on the knowledge on only one class (*floor*). If the floor is known consisting of a single color, the problem becomes easy and can be solved by seeded region growing with a pre-defined threshold for finding edges, e.g. the Taylor et al.' system. However, we may have multiple colors in the histogram sampled from the safe region. We thus propose the following scheme to determine if a pixel belongs to the floor, which uses both histograms and edges:

1. The current pixel for determination should be a neighboring pixel of a pixel already identified as *floor*;
2. The histogram bin corresponding to the color of the current pixel is firstly checked - if it is *true*, the current pixel can be determined as *floor* and the determination process is complete;
3. A 3×3 Laplacian edge detector is convolved with the given pixel in *R*, *G* and *B* channels respectively. If the convolved value in any channel is above a pre-defined threshold, the current pixel is determined as *obstacle* and the determination process is complete;
4. The current pixel is then determined as *floor*, since it has a similar color with one of the known colors of the floor. The current pixel's color is used to update the histogram by labeling the corresponding bin as *true*.

With this scheme, a pixel can be identified as *floor* or *obstacle* with either histogram or edge constraints. Since the pixel to be determined needs to be a neighboring pixel of a pixel determined as *floor*, we will now discuss the image pixel scanning scheme.



(a) (b)

Fig. 1. Image label illustrations: (a) before scanning; (b) an example result after scanning

The *image region of interest* is subdivided into three sub-regions: center, left and right. The center sub-region represents the main path for the user, i.e. the floor area that the user would step on if walking straight ahead. The left and right sub-regions represent the areas to the side and can become alternative paths if the main path is obstructed. Figure 1(a) illustrates such sub-divisions. We propose a pixel scanning scheme that efficiently computes the safe depth in each of these three paths.

1. Considering all pixels at the bottom line of the *image region of interest* as seed points, we first process the center sub-region and then the side sub-regions;
2. For each sub-region, we initialize V_i (i=center, left or right), defined as the y coordinate of its safe depth visualized in the image, as the y coordinate of a top pixel in the *image region of interest*. After that, each seed point will be processed: starting from the seed point that is the closest from bottom center of the image and then propagating to left/right;
3. From each seed point, a set of pixels (a line segment from the seed point to a point where $y = V_i$) are of interest which represents a very narrow path that is parallel to the user's orientation.
4. For each pixel of interest, if it is labeled *unvisited*, we apply the aforementioned scheme to determine if it is *floor* or *obstacle*. After the determination (if required), if a pixel is labeled as obstacle, the rest of unexplored interested pixels from the same seed point is discarded and V_i is updated to the current y coordinate. All pixels beyond $y = V_i$ can be skipped since the obstacle has been found in this region.

After the above pixel scanning process, V_{center}, V_{left} and V_{right} can be obtained which can be converted to corresponding depths d_{center}, d_{left} and d_{right} in metric units, given the camera focal length and tilt angle of the camera. If any of these safe depths is over a pre-defined threshold, the corresponding path is considered a safe path. The safe path(s) can hence be found. Figure 1(b) illustrates an example resulted from the scanning. In this example, the solid thick line represents the border of the expected obstacle. According to the proposed scanning scheme described above, the safe depth of each path can be determined after encountering with only a small number of pixels of the obstacle, eliminating the need to process many other pixels in the *image region of interest*.

4 Implementation and Usage

The proposed system is implemented on a mid-range Smartphone - Nokia E71. The main built-in camera of this Smartphone has the maximum video frame rate of 15 fps and minimum image dimension of 128×96 pixels for video capture. It works on a single 369 MHz ARM 11 processor. With the S60 3rd Edition SDK for Symbian OS and Carbide.c++ IDE, applications can be developed and many features on the phone can be controlled, such as camera, vibration feedback, voice, etc.

Because there is no accelerometer on this particular Smartphone, we assume that the user will hold the phone at a tilt angle of about 45° , which allows the depth threshold for safe path to be similar to the height of the camera. The height of the camera is pre-defined as one meter and is adjustable by simply pushing a button. With the pre-defined camera height, the farthest detectable depth is about two meters since the embedded camera has a field of view of about 40° vertically. In addition, the input image is sub-sampled to 64×48 pixels for storage and performance considerations. In terms of feedback, like other ETAs such as the MiniGuide, the vibration feedback is provided if the main (center) path is not safe. In addition, auditory feedback can also be provided *on demand*, mainly for new users, through a button giving verbal instructions as to the safe depth on the main path. If the safe depth of the main path is under a threshold (unsafe to proceed), it further advises as to which side paths are safe.

The standard usage of the system is as follows:

1. The users may adjust the camera height based on the pose which they find most comfortable. The determination of the camera height may require assistance from another person;
2. The users hold the phone in the correct pose: put it in the center in front of the body, and point it forward with about 45° downward tilt angle;
3. The users can keep walking forward until the Smartphone is vibrating;
4. If the Smartphone is vibrating and the auditory feedback is not utilized, the user may sweep the Smartphone left or right, or point to other directions until it stops vibrating. The users should then adjust themselves to the pointing direction of the Smartphone and continue to proceed forward;

5. Whether the Smartphone is vibrating or not, if the users *want* the auditory feedback, they may push a button to obtain auditory feedback on demand. For example, if there is an obstacle, the users may hear “0.9 meters, bear left”. In this case, the users should step a little bit to the left and the Smartphone should stop vibrating and thus the users can proceed forward. If there is another obstacle far from the user, the users may hear “1.9 meters”, and the user is given an idea how far the obstacle is.

5 Results

The proposed system implemented on the Smartphone is evaluated in different environments before it is tested in a field trial and compared against two existing approaches. The results on the evaluation are first presented, followed by the discussion on the field trial experiment.

5.1 Quantitative Evaluation

We will compare the proposed system with Tan et al.’s color histogram based system and Taylor et al.’s edge based system. We hold the Smartphone at a tilt angle of about 45° and about one meter above the floor. Therefore the depth range for dangerous obstacles is around one meter, which means any obstacle farther than one meter is considered safe. For the purposes of evaluation, we record the input images as we navigate. The recorded input images are then processed using Taylor et al.’s system, Tan et al.’s system, and the proposed system.

In Tan et al.’s system, up to four normalized *rg* histograms are used to represent the road (based on the reference area) and one *rg* histogram is built for the background (from the previous frame). It uses a linear combination of current and previous frame to compute the probability of road at each pixel after the distance measurement. Each pixel can be determined as either road (*floor*) or background (*obstacle*). After that, our pixel scan direction is used to find obstacles in the user’s path (see Section 3.3). In their original paper, the classified pixels are further fit with a road model, which is irrelevant to this paper and hence not implemented.

In Taylor et al.’s system, a basic seeded region growing method is used to find same-color regions in the images based on r, g, h and i channels. Our pixel scan direction is then used to find obstacles in user’s path.

Accuracy We conduct a set of tests to evaluate the accuracy of the three systems in three different indoor environments, where the floor is (1) un-patterned non-reflective, (2) patterned non-reflective or (3) un-patterned reflective. An example from each sequence is shown in Figure 2. In each test, we manually label the ground truths in each input image: if any obstacle is present within the bottom half (up to about one meter’s distance) of central trapezoid region (the center path), the frame is labeled as positive. Otherwise, it is labeled as negative.

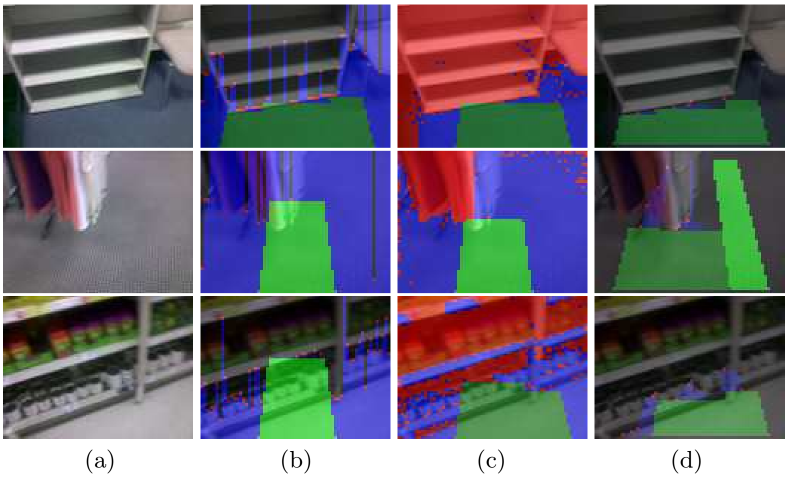


Fig. 2. Example frame from Sequence #1∼#3(Row 1∼3): (a) original input image; (b) output using Taylor's method; (c) output using Tan's method; (d) output using the proposed method.

For each device, the true positives (TP) which represents an obstacle has been correctly identified after manual verification, the false positives (FP), the true negatives (TN) and the false negatives (FN) are recorded for each frame in each sequence. We compute

- *positive predictive value* (a.k.a *precision*, $TP/(TP + FP)$) indicating how much a positive feedback (e.g. vibration) can be trusted;
- *negative predictive value* ($TN/(TN + FN)$)indicating how much a negative feedback (e.g. no vibration) can be trusted;
- *sensitivity* (a.k.a *recall*, $TP/(TP + FN)$ indicating how reliable the system is to pick up all obstacles;
- *specificity* ($TN/(TN+FP)$) indicating how reliable the system is to identify a safe path;
- the overall accuracy of the system ($(TP + TN)/(TP + FP + TN + FN)$).

As Tables 1∼3 show, the proposed system outperforms other systems in every test. In general, Tan et al. and Taylor et al.'s systems are usually quite poor at avoiding false alarms. Overall, the proposed system has a overall accuracy of over 94% in all these situations, while the other two systems can only achieve about 80% in certain situation(s).

Speed While accuracy is important in determining if an obstacle can be correctly identified, the speed of processing is also important as it controls how heavy the computation is and how quickly the user will receive a response. The performance data are listed in Table 4. In comparison between these three real-time systems, the proposed system only takes 7ms to compute which is 10% of

Table 1. Results - Quantitative - Sequence #1 (464 frames)

Measurement	Taylor et al.	Tan et al.	Proposed
Positive Predictive Value/Precision	89.58%	45.51%	96.00%
Negative Predictive Value	77.17%	83.69%	99.31%
Sensitivity/Recall	50.59%	86.47%	98.82%
Specificity	96.60%	40.14%	97.62%
Overall Accuracy	79.74%	57.11%	98.06%

Table 2. Results - Quantitative - Sequence #2 (375 frames)

Measurement	Taylor et al.	Tan et al.	Proposed
Positive Predictive Value/Precision	13.00%	64.91%	100.0%
Negative Predictive Value	77.63%	91.82%	94.26%
Sensitivity/Recall	46.03%	58.73%	69.84%
Specificity	37.82%	93.59%	100.0%
Overall Accuracy	39.20%	87.73%	94.93%

Table 3. Results - Quantitative - Sequence #3 (496 frames)

Measurement	Taylor et al.	Tan et al.	Proposed
Positive Predictive Value/Precision	34.48%	73.86%	97.50%
Negative Predictive Value	72.65%	86.70%	93.36%
Sensitivity/Recall	65.22%	70.65%	84.78%
Specificity	42.71%	88.44%	99.00%
Overall Accuracy	49.83%	82.82%	94.50%

the idle time and thus significantly save the battery life while Tan et al.'s system requires about 45% of the idle time and Tayler's system takes up to 100%. Taylor et al.'s system has wide variation in speed because it depends on the distance of the obstacle: it performs faster when the obstacle is close as fewer pixels need to be processed. Based on the results from accuracy and performance comparison, we then provide the proposed system to real blind people for a field trial.

Table 4. Results - Speed

	Taylor et al.	Tan et al.	Proposed
Computation time per frame	8∼83 ms	30 ms	7 ms
Time spent (percentage of idle time)	12%∼125%	45%	10%
Theoretical frame rate (fps)	12∼125	33	150
Actual frame rate (fps)	8∼15	15	15

5.2 Real-world Evaluation

The proposed system implemented on the Smartphone was given to several blind users for evaluation. We will discuss the goal, environment, subjects, procedures, issues, feedback and limitations of the field trial.

Goal The goal of this experiment is to evaluate to what extent a blind user feels the differences among the three systems and how good the systems are in relation to each other from a human perspective. The reason for doing this is that it is difficult to translate quantitative accuracy numbers from Table 1∼3 into human experience.

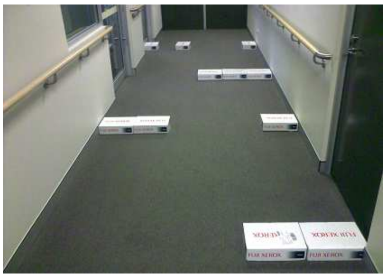


Fig. 3. Experiment setup

Environment The experiment took place in the Association for the Blind of Western Australia. Eleven paper boxes were randomly placed in a corridor by ensuring there was no straight safe path (see Figure 3). Low height boxes are chosen due to safety concerns and the blind participant would not be able to sense the boxes even by taking advantage of echolocation. Therefore, the result obtained under such environment setup would not be affected by the skills of the participant.

Subjects Five blind adult volunteers were invited to participate in this experiment. We identify them as P1 to P5 in this paper. To ensure that the evaluation is unbiased, we did not tell the participant which system was developed by us until after the experiment was finished. Each participant was first given a random system for testing, during which the participant can get familiar with correctly holding the Smartphone and learn how to obtain/understand feedback from the Smartphone before the evaluation process starts. The order of the systems given to the participant for evaluation was randomly defined because the number of participants does not meet the requirement of Latin square.

Procedures During the testing and evaluation of the systems, the participants were requested to walk through the corridor with only the Smartphone, without white cane, guide dog or other ETAs. The participant was asked to point the Smartphone to the front with about 45° tilt angle and stop proceeding once the Smartphone starts vibrating. Once the Smartphone vibrates, the participant either swept the phone to find a clear path or used auditory feedback (their choice). Though detailed instructions were given, the participant was followed by a human guide to ensure safety. The human guide only provided three types of assistance: 1) reminding the participant of the correct holding posture of the Smartphone; 2) stopping the participant from proceeding when an obstacle is about to be encountered but the participant shows no sign of stopping; 3) informing the participant of the safe path to proceed when the participant could not find a clear path using the Smartphone. After the participants finished experiencing the three different systems, they were asked to rank the systems (if they could feel the differences) and give each system a score between one (worst) and ten (best). Further comments from the participant were also recorded.

Issues One of the five participants (P3) had difficulty holding and using the phone as needed, and due to the limited training time was unable to overcome this. Hence P3 could not distinguish between the three systems. In contrast, P2 got used to the device very quickly as we were told that P2 had been using another ETA, the MiniGuide, for about 10 years. Obviously, the system presents usability barriers to its usage, but all existing ETAs also suffer from similar issues.

Feedback The feedbacks from all participants except P3 are then consolidated and the rating for each system is shown in Table 5. Only P1 thought Tan et al.'s system was better. All others preferred the proposed system. As Table 5 shows, the proposed system is generally 50% better than Tan et al.'s system and is almost twice as good as Taylor et al.'s system according to the participants.

Table 5. Results - Real World

Subject	Taylor et al.	Tan et al.	Proposed
P1	5	10	7.5
P2	7	4	10
P3	N/A	N/A	N/A
P4	4	4	10
P5	4	7	10

Limitations During the experiment, we observed that it is not easy for most participants to hold the Smartphone at the requested tilt angle (around 45°) all the time. Since there is no embedded accelerometer in this specific Smartphone, longer training time would be useful for the user to get accustomed to it. The problem can be addressed if a Smartphone with accelerometer is used.

6 Conclusions and Future Work

This paper presents a real-time obstacle detection system implemented on a Smartphone, which can be used by the visually impaired as a mobility tool. By combining color histograms, edge cues and pixel-depth relationship, the proposed system is able to detect on-floor obstacles and provides feedback to the user through vibration and voice-on-demand. The proposed system has been tested in different environments and provides consistent and reliable results despite the simplicity of the system. The proposed system has been evaluated by blind users and received a high ranking. There is still plenty of room to improve the proposed system, such as utilizing the embedded accelerometer in some Smartphone and dealing with more complex floor patterns, etc.

References

1. J. M. Benjamin, N. A. Ali and A. F. Schepis. A Laser Cane for the Blind. Proceedings of the San Diego Biomedical Symposium 12:53-57 (1973)
2. A. J. Davison, I. D. Reid, N. D. Molton and O. Stasse. MonoSLAM: real-time single camera SLAM. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 29(6): 1052-1067 (2007)
3. A. P. Duchon, W. H. Warren and L. P. Kaelbling. Ecological robotics. Adaptive Behavior, 6(3-4):473-507 (1998)
4. T. Gevers and A. W. M. Smeulders. Color-based object recognition. Pattern Recognition, 32(3): 453-464 (1999)

5. J. Liebelt, C. Schmid and K. Schertler. Viewpoint-independent object class detection using 3D feature maps. In Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (2008)
6. C. Plagemann, F. Endres, J. Hess, C. Stachniss and W. Burgard. Monocular range sensing: a non-parametric learning approach. In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA) (2008)
7. A. Saxena, M. Sun and A. Y. Ng. Make3D: learning 3-D scene structure from a single still image. IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI), 30(5): 824-840 (2009)
8. N. Snavely, S. M. Seitz and R. Szeliski. Modeling the world from Internet photo collections. International Journal of Computer Vision (IJCV), 80(2): 189-210 (2008)
9. K. Souhila and A. Karim. Optical flow based robot obstacle avoidance. International Journal of Advanced Robotic Systems, 4(1): 13-16 (2007)
10. M. Sridharan and P. Stone. Color learning and illumination invariance on mobile robots: a survey. Robotics and Autonomous Systems (RAS) Journal, 57(6-7):629-644 (2009)
11. C. Tan, T. Hong, T. Chang and M. Shneier. Color model-based real-time learning for road following. In Proceedings of IEEE Intelligent Transportation Systems Conference (ITSC) (2006)
12. T. Taylor, S. Geva and W. W. Boles. Monocular vision as a range sensor. In Proceedings of International Conference on Computational Intelligence for Modelling (CIMCA) (2004)
13. E. Tola, S. Knorr, E. Imre, A. A. Alatan and T. Sikora. Structure from motion in dynamic scenes with multiple motions. In Workshop on Immersive Communication and Broadcast Systems (ICOB) (2005)
14. I. Ulrich and J. Borenstein. The GuideCane-applying mobile robot technologies to assist the visually impaired.IEEE Transactions on Systems, Man, and Cybernetics, Part A, 31(2):131-136 (2001)
15. I. Ulrich, and I. Nourbakhsh. Appearance-based obstacle detection with monocular color vision. In Proceedings of the AAAI National Conference on Artificial Intelligence (2000)
16. The Association for the Blind of WA. http://www.abwa.asn.au (2010)
17. GDP Research. http://www.gdp-research.com.au (2010)
18. Minoru 3D webcam. http://www.minoru3d.com (2010).
19. Currently Available Electronic Travel Aids for the Blind. http://www.noogenesis.com/eta/current.html (2010)