# An Edge Detection Framework Conjoining with IMU Data for Assisting Indoor Navigation of Visually Impaired Persons

Kit Yan Chan[*,1], Ulrich Engelke[**3], Nimsiri Abhayasinghe[*,2]
[*]Department of Electrical and Computer Engineering, Curtin University, Australia
[1]kit.chan@curtin.edu.au, [2]K.Abhayasinghe@curtin.edu.au
[**]Digital Productivity Flagship of the Commonwealth Scientific and Industrial Research Organisation (CSIRO), Hobart, Australia
[3]Ulrich.Engelke@csiro.au

*Abstract* — **Smartphone applications based on object detection techniques have recently been proposed to assist visually impaired persons with navigating indoor environments. In the smartphone, digital cameras are installed to detect objects which are important for navigation. Prior to detect the interested objects from images, edges on the objects have to be identified. Object edges are difficult to be detected accurately as the image is contaminated by strong image blur which is caused by camera movement. Although deblurring algorithms can be used to filter blur noise, they are computationally expensive and not suitable for real-time implementation. Also edge detection algorithms are mostly developed for stationary images without serious blur. In this paper, a modified sigmoid function (MSF) framework based on inertial measurement unit (IMU) is proposed to mitigate these problems. The IMU estimates blur levels to adapt the MSF which is computationally simple. When the camera is moving, the topological structure of the MSF is estimated continuously in order to improve effectiveness of edge detections. The performance of the MSF framework is evaluated by detecting object edges on video sequences associated with IMU data. The MSF framework is benchmarked against existing edge detection techniques and results show that it can obtain comparably lower errors. It is further shown that the computation time is significantly decreased compared to using techniques that deploy deblurring algorithms, thus making our proposed technique a strong candidate for reliable real-time navigation.**

***Keywords* — Edge detection, sigmoid function, particle swarm optimization, smartphone navigation, vision impaired persons, inertial measurement unit (IMU).**

## 1. INTRODUCTION

Today, about 4% of the world's population (285 million people) are visually impaired, 13% of whom are blind and the other 87% have low vision (W.H.O., 2014). Vision frameworks and mechanism have been developed for assisting visually impaired users in navigation and detecting objects (Ivanchenko, Coughlan, & Shen, 2010; Kang, Chae, Sun, Yoo, & Ko, 2015; Mascetti, Ahmetovic, Gerino, Bernareggi, Busso, & Rizzi, 2016; Pham, Le, & Vuillerme, 2016; Pradeep, Medioni, & Weiland, 2010; Solymár, Stubendek, Radványi, & Karacs, 2011; Winlock, Christiansen, & Belongie, 2010). To assist indoor navigation for visually impaired persons, a range of smartphone platforms have been developed. These smartphones provide pathway from the current position to the destination position (Moreno, Shahrabadi, Jose, Buf, & Rodrigues, 2012; Williams, 2014). Unlike fully sighted persons, visually impaired persons are not able to see pathways from visual scenes which are in front of them, although travel routes are provided by smartphones. To provide mobility assistance for visually impaired people, white canes and guide dogs have traditionally been used. However, a white cane is very intrusive and is not safe enough to navigate insecure places. Training a guide dog is expensive and also guide dogs are socially unacceptable in many indoor environments such as offices (News, 2013). In addition to the traditional mobility assistance, ultrasonic sensing can be used to scan the pathways (Antoun, & McKerrow, 2007; Peng, Peursum, Li, & Venkatesh, 2010). The echoes of ultrasonic waves are interpreted to form an auditory scene enabling the person to identify the pathway. However, ultrasonic sensing requires a very narrow scan beam, and works effectively only on a hard and flat surface by receiving ample sound echo.

Motivated by the popularity of smartphone, navigation systems have been developed by visually impaired persons (Rodriguez-Sanchez, Moreno-Alvarez, Martin, Borromeo, & Hernandez-Tamames, 2014). The navigation systems use the smartphone camera to capture the pathway scenes as images. Based on the captured images, corridors, entries, exits (Segvic, & Ribaric, 2001; Shi, & Samarabandu, 2006; Yang, & Tsai, 1999), can be used to provide semantic interpretations for the pathways, and warming signals can be generated in order to avoid collisions with walls (Asad, & Ikram, 2012; Ivanchenko, Coughlan, & Shen, 2009; Shen, Chan, Coughlan, & Brabyn, 2008). This information can be used to help visually impaired people to avoid collisions and to find doors and corridors which are necessary to take them to their destination. Prior to performing object detection or pattern recognition from captured images, it is necessary to identify edges on the objects of which the object edges are the perceivable features (Rodriguez-Sanchez, Moreno-Alvarez, Martin, Borromeo, & Hernandez-Tamames, 2014; Segvic, & Ribaric, 2001; Asad, & Ikram, 2012;

Ivanchenko, Coughlan, & Shen, 2009). This can be accomplished using edge detection algorithms (Lee, Haralick, & Shapiro, 1987).

The edge detectors generally evaluate the image contrast (Peli, 1990), which represents the luminance difference between each neighboring pixel, in order to illustrate objects in the image. When the image contrast of a set of continuous pixels is above the threshold, an object edge is identified. However, while a person is walking, blur is typically introduced into the video frames due to the technical limitations of smartphone cameras. Typically, this blur decreases image activity, thereby reducing the overall image contrast. Because of the reduced contrast, object edges are less likely to be detected. Recent edge detection algorithms are mostly developed on detecting edges on stationary images without serious blur (Shi, & Samarabandu, 2006; Venkatesh, & Rosin, 1995; Guerrero, Vasquez, & Ochoa, 2012; Rajakaruna, Rathnayake, Abhayasinghe, & Murray, 2014). Based on our knowledge, edge detection has not been studied on blurred images which are taken by cameras with large movements.

Although the deblurring algorithm can be implemented in the smartphone platform to improve image quality degraded with blur (Joshi, Kang, & Zitnick, 2010), it can only remove the blur caused by small camera motion such as long hand-held exposure. To remove blur caused by large camera motion, approaches for video deblurring can be used (Deng, Shen, Song, Tao, Bu, & Chen, 2010; Cho, Wang, & Lee, 2012; Huang, & Fan, 2011; Lee, Jeong, Lee, & Song, 2013). These approaches first estimate the camera motion trajectory. Based on the trajectory, a deblurring kernel is computed in order to remove the blur from the original frame. However, the approaches for video deblurring are computationally expensive and are therefore not suitable for implementation in real-time navigation. Sigmoid functions are computationally inexpensive for image enhancement and can be implemented in real-time (Braun, & Fairchild, 1999). Based on the sigmoid functions, the luminance of the image pixels can be increased when they are relatively high compared with the luminance of the whole image, and the luminance of the image pixels can be reduced when they are relatively low. The sigmoid function attempts to make the dark pixels in an image darker, and the light pixels lighter. Hence, the image contrast can be boosted while maintaining the original image structure, and thus the object patterns are more likely to be detected.

The sigmoid function has been applied to perform edge detection, which generates zero, half or one (Douglas, & Meng, 1989). Edges are represented by one; empty spaces are represented by zero; boundaries between edges and empty spaces are represented by half. In this approach, the original images are first passed to a Finite Impulse Response (FIR) adaptive filter before passing to the sigmoid function for performing edge detection. Although the

FIR adaptive filter is optimized with respect to a set of collected past images, the approach can only process images with similar edge and noise characteristics to the past collected images. Also the FIR adaptive filter is developed to process images with slight blur. The FIR adaptive filter may not work effectively with respect to images captured under large camera movement and also images which are different to the past collected images. Although (Mhaned, Abid, & Fnaiech, 2012) and (Braun, & Fairchild, 1999) have applied sigmoid functions to perform image enhancement for defect detection and object detection, the sigmoid function parameters are sensitive to different visibility conditions caused by blur (Kannan, Deepa, & Ramakrishnan, 2010). A single set of sigmoid function parameters can work properly only on a single blur level and cannot work effectively on a range of blur levels. This parameter set has to be specified in order to increase the effectiveness of detecting interested object rather than that of other uninterested objects in the image.

In this paper, a modified sigmoid function (MSF) framework is proposed in order to improve image quality for corridor detection. The MSF framework consists of a modified sigmoid function (MSF) and an inertial measurement unit (IMU), where the MSF is used to improve the effectiveness of detecting interested objects and the IMU is used to capture the camera motion in order to determine the appropriate parameters for the MSF. Based on the determined MSF parameters, the captured image can be enhanced with respect to the corresponding camera motion. The MSF framework first uses the IMU (Lane, Miluzzo, Lu, & Peebles, 2010) to capture the camera motion, where the IMUs are usually embedded in smartphones and tablets. Based on the captured motion, the blur levels can be estimated (Joshi, Kang, & Zitnick, 2010). An optimization algorithm, namely the particle swarm optimization (PSO) (Parsopoulos, & Vrahatis, 2004) which is effective in solving difficult optimization problems, is used to determine a database for sigmoid function parameters. Each single set of sigmoid function parameters is specified with respect to a particular motion level which is captured by the IMU. In the PSO, a cost function is formulated in order to optimize a single set of sigmoid function parameters with respect to contrast enhancement. The cost function is particularly developed to detect object edges which are interested rather than the whole scene. Based on the determined sigmoid function parameters, the object contrast can be enhanced for interested objects and also the MSF can generate more precise object edges for different motion levels. As a result, the effectiveness of edge detection can be improved by the proposed MSF.

To evaluate the effectiveness of the MSF framework, we consider four scenarios involving visually impaired persons engaged in navigation. Video sequences associated with the IMU datasets are used. The sequences simulate

the blind persons using white canes and guide dogs for navigation. The white cane sequence exhibits a small movement and contains weak blur. The guide dog sequence exhibits a large movement and contains strong blur. The MSF framework is used to detect indoor corridors and entries which are importance for indoor navigation of visually impaired persons (Segvic, & Ribaric, 2001; Shi, & Samarabandu, 2006; Yang, & Tsai, 1999). Four commonly used edge detectors (Lee, Haralick, & Shapiro, 1987; Canny, 1986; Kimmel, & Bruckstein, 1984) are used within the MSF framework. The following results are considered: the one based on solely using the edge detectors; the one generated by using the proposed MSF framework; and the one generated by using two deblurring algorithms including the deconvolution filter (Horstmeyer, 2010) and Whyte et al.'s filter (Whyte, Sivic, Zisserman, & Ponce, 2012), where the deconvolution filter can be implemented on smartphones to improve blurred images (Joshi, Kang, & Zitnick, 2010) and Whyte et al.'s filter is a recently developed deblurring method. For video sequences with small blur, similar results are obtained based on the tested methods. For those with strong blur, results show that the proposed MSF framework obtains smaller corridor and entry detection errors compared with the other tested methods. Also the proposed MSF requires less computational time. Hence, it is a strong candidate for real-time navigation.

The key contributions of the proposed MSF framework are summarized as follows:

1. The MSF framework is developed for edge detections for images which are distorted with large blur, while traditional approaches for edge detection are only developed for stationary images with small blur.
2. The MSF framework is computationally inexpensive compared with the deblurring algorithms for enhancing blurred image. Hence, it is a good candidate for real-time implementation.
3. The MSF framework obtains comparably lower object detection errors compared with commonly used edge detection algorithms. It further shows that MSF framework is a strong candidate for assisting indoor navigation.

The rest of this paper is organized as follows: Subsection 2.1 presents the limitation of the commonly used edge detection algorithms. Subsection 2.2 discusses the formulation of the MSF and the motivation of why we proposed the MSF. Subsection 2.3 presents the mechanism of the proposed MSF framework. Section 3 presents two case studies which are used to evaluate the effectiveness of the proposed MSF framework. One case demonstrates the small camera movement and another one demonstrates the large one. Finally, a conclusion and discussion of possible future research are given in Section 4.

## 2. MSF FOR IMPROVING EDGE DETECTION

### *2.1 Limitation of the commonly used edge detection algorithms*

We consider an indoor navigation for a vision impaired person who is led by a guide dog. A smartphone is hung around the person's neck to capture video for edge detection. Figure 1a shows one of the video frames. It shows the corridor view, where the head of the white guide dog appears on the bottom left of the image. The circles highlight the corridor edges bordered by the walls and the floor. We can observe that the corridor edges are not clearly bordered, as motion blur is generated by the person's movement. Hence, the corridor edges can barely be distinguished.

Based on the captured video frame, the commonly used Canny edge detector (Canny, 1986) is used to generate the edge diagram shown in Figure 1b, where the threshold for handing images with no blurs is used on the algorithm. The edge detector first evaluates the contrast between the luminance of each neighboring pixel. When the contrast of a set of neighboring pixels is above the pre-defined threshold, an edge between two different regions is identified on the image. Figure 1b shows that some of the left and right edges of the corridor are missing. As blur typically reduces the amount of image activity, the image is smoothed and values on neighboring pixels exhibit lower contrast. Hence, corridor edges are less likely to be detected by the algorithm and thus some corridor edges cannot be detected in Figure 1b.

One can use a lower threshold in order to identify edges which are formed based on neighboring pixels with less contrast. The corridor edges in the blurred image are more likely to be detected. However, with a smaller threshold, the edge detectors are more sensitive to artifacts and are more likely to produce more false than true edges (Martin, Fowlkes, & Malik, 2004; Venkatesh, & Rosin 1995). It downgrades the quality of the edge diagram as more false edges are created.

**Figure 1a** Original video frame.

**Figure 1b** Edge diagram of the original video frame; Canny edge detector (Canny, 1986) was used.

### *2.2 Modified sigmoid function (MSF)*

To detect corridor edges more effectively, a modified sigmoid function (MSF) (Braun, & Fairchild, 1999) in equation set (1) is proposed by increasing the luminance differences between the regions of floor and walls. The image contrast between the two regions is increased with the increased luminance differences. Hence, the corridor edges for the two regions are more likely to be detected, and thus the MSF increases the effectiveness of the edge detector. The MSF is given as:

$$y_{ij} = \begin{cases} d & \text{if } x_{ij} < c \quad (1a) \\ d + f\left(x_{ij}, a, e\right) & \text{if } c \le x_{ij} < (c+b) \quad (1b) \\ 1 & \text{if } x_{ij} \ge (c+b) \quad (1c) \end{cases}$$

$$\text{where } f\left(x_{ij}, a, e\right) = \frac{1}{1+\exp\left(-a\left(x_{ij}-e\right)\right)} \cdot (1-d). \quad (1d)$$

The derivative of the MSF is given as:

$$y_{ij} = \begin{cases} 0 & \text{if } x_{ij} < c \quad (2a) \\ \dfrac{d\left(f\left(x_{ij}, a, e\right)\right)}{d\left(x_{ij}\right)} & \text{if } c \le x_{ij} < (c+b) \quad (2b) \\ 0 & \text{if } x_{ij} \ge (c+b) \quad (2c) \end{cases}$$

$$\text{where } \frac{d\left(f\left(x_{ij}, a, e\right)\right)}{d\left(x_{ij}\right)} = \frac{a \cdot (1-d) \cdot \exp\left(-a\left(x_{ij}-e\right)\right)}{\left[1+\exp\left(-a\left(x_{ij}-e\right)\right)\right]^2}. \quad (2d)$$

Here $x_{ij}$ denotes the luminance of the image pixel with the coordinates $i$ and $j$, where $x_{ij}$ is between 0 to 1; $i = 1, 2, ..., m$; $j = 1, 2, ..., n$; and $m$ and $n$ are the image dimensions. The value of $x_{ij}$ corresponds to the intensity. The MSF consists of three sub-functions namely (1a), (1b) and (1c). (1a) is used to process the floor luminance in Figure 1a, where the floor luminance is smaller than $c$. $d$ with $(0 \le d < 1)$ is multiplied to the original luminance, $x_{ij}$, at the floor, and thus the processed luminance, $y_{ij}$, at the floor becomes smaller. As a result, the floor shown in Figure 4a becomes darker. (1c) is used to identify the wall which has the luminance higher than $(c+b)$. Its corresponding luminance is multiplied by 1 and thus it remains unchanged.

Equation (1b) is engaged with the sigmoid function, where the sigmoid slope is represented as the derivative (2b) as illustrated in Figure 3. The sigmoid slope is adjusted by $a$, and the parameter, $e$, is used to adjust the sigmoid width

which is also illustrated in Figure 3. When the image blur is strong, the border width between walls and floor would be large. The image activity of the borders between walls and floor is small, and thus corridor edges are unlikely to be identified by the edge detectors. As the edge detector generally identifies object edges based on luminance differences between image pixels, object edges are more likely to be detected when the luminance derivatives of image pixels are high. The MSF with an appropriate sigmoid slope and sigmoid width can be used particularly to multiply to the luminance on the border, in order to narrow the border width between corridors and floor. With a narrower border, the luminance derivative between corridors and floor is larger. Hence, the border can easily be identified by the edge detector. We attempt to determine an appropriate set of MSF parameters in order to increase the likelihood of identifying the object edges of interest.

As an illustration, the characteristic of the MSF is given in Figure 2 and the derivative of the MSF is given in Figure 3, when $a$=1, 2, 3, or 6; $b$=0.55; $c$=0.15; $d$=0.55 and $e$=4. This example demonstrates the effect of determining the edge when different slopes are used, as changes of pixel intensities are different with different slopes. The values are derived manually in order to see this effect. The processed luminance for each pixel namely $I_{ij}$ is given by multiplying $y_{ij}$ by $x_{ij}$, which is given as

$$I_{ij} = x_{ij} \times y_{ij} \tag{3}$$

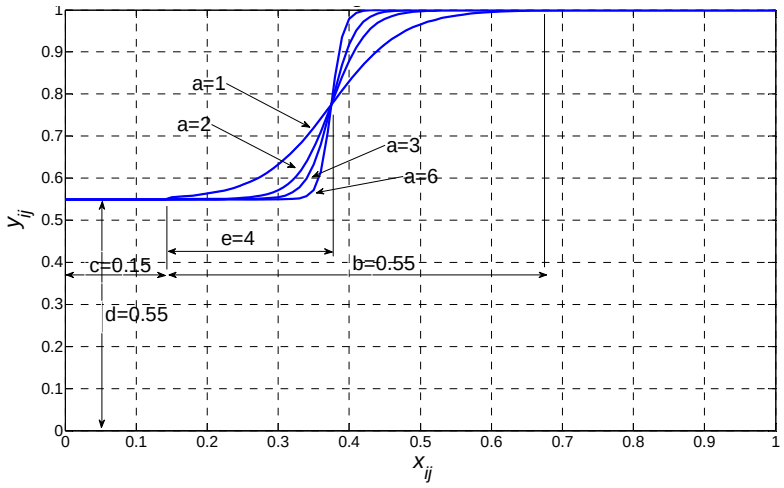



**Figure 2** Illustration of the modified sigmoid function.

When comparing the original video frame shown in Figure 1a with the processed image shown in Figure 4a, we can observe that the luminance difference between the walls and the floor shown in the processed video frame is greater than that shown in the original video frame. Also, we can compare the edge diagrams, Figure 1b and Figure 4b, for the original image and the processed image respectively. Figure 4b shows that more corridor edges can be detected from the processed image; while Figure 1b shows that some corridor edges cannot be detected from the original image. Therefore, the MSF improved the effectiveness of detecting corridor edges in this particular example. A thorough validation of these observations is provided in the edge images, Figures 1b and 4b, generated based on the original and processed images respectively.

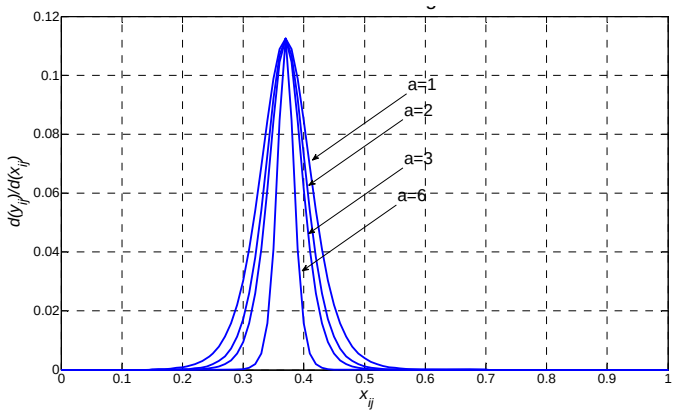



**Figure 3** Derivative of the modified sigmoid function.

**Figure 4a** Processed video frame.

**Figure 4b** Edge diagram of the processed video frame.

Apart from using the MFP for corridor detection, we consider the entry detection which is also important for navigation by visually impaired persons (Shi, & Samarabandu 2006; Tian, Yang, & Arditi 2010). The MSF parameters are set as $a$=1.382, $b$=1.123, $c$=0.508, $d$=0.417 and $e$=0.459. Figure 5a shows the original image with three

entries, where the image was captured under a small motion. Hence, we can observe that image blur exists. Figure 5b shows the edge image for the original image, where the Canny edge detector is used to detect object edges. The figure shows that only a few edges of the entries can be detected, as the existing blur reduces the amount of image activity and thus many entry edges cannot be detected. It is unlikely that a person would be able to perceive the door patterns.

To improve the edge diagram, the MSF is used to process the original image by increasing the luminance differences between entries and walls. It attempts to increase the likelihood of identifying the entry edges. Figure 6a shows the processed image which was generated by the MSF. We can observe that the luminance difference between the walls and the doors in the processed image is greater than that shown in the original video frame. Therefore, edges are more likely to be detected in the processed image and thus Figure 6b shows that more entry edges are detected when the processed image is used. Based on the edge diagram of the processed image, the door patterns are more likely to be identified by a person than the edge diagram of original image.

**Figure 5a** Original image.

**Figure 5b** Edge diagram of the original image

**Figure 6a** Processed image.

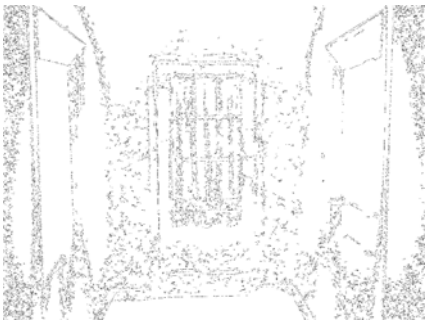


**Figure 6b** Edge diagram of the processed image

In the MSF, the appropriate parameters are necessary to be determined in order to detect edges between two regions more effectively. Figure 7a shows the segment with $6 \times 6$ pixels which is subtracted from the original image shown in Figure 1a. We can barely observe that there are two regions, where the region at the left hand side is formed by the

slightly lighter pixels and the region at the right hand side is formed by the slightly darker pixels. It is difficult to distinguish the borders between the two regions.

The borders can be identified more clearly when the original segment is processed by the MSF. Figures 7b, 7c and 7d show the processed images when three different sets of MSF parameters are used. We can observe that the clearest border exists in Figure 7b compared with the other two processed images in Figures 7c and 7d. It seems that the MSF parameters for Figure 7b are more appropriate than those for Figures 7c and 7d. Therefore, it is important to determine an appropriate set of MSF parameters for the blurred image to optimize edge detection. However, the amount of blur in each video frame is not a constant, as the camera movement is time-varying when recording the video. Therefore, the MSF parameters need to be adjusted with respect to camera movement against time. In Section 2.3, the MSF framework based on camera movement captured by IMU is proposed to adjust MSF parameters in order to detect corridor edges more effectively.

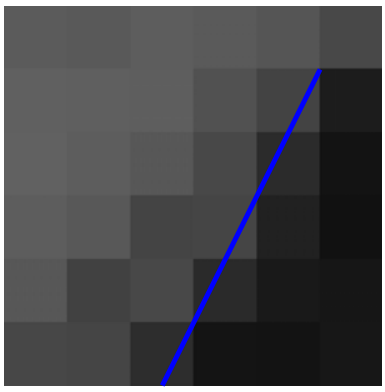
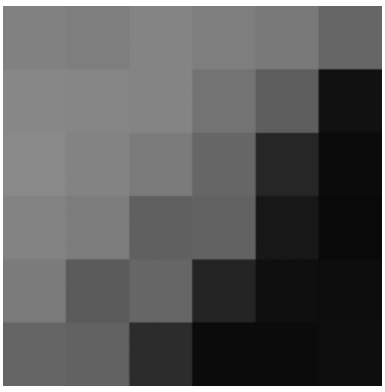
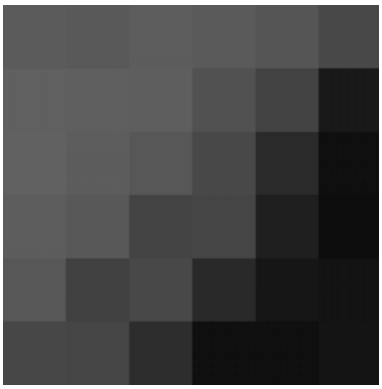


**Figure 7a** The Original segment.



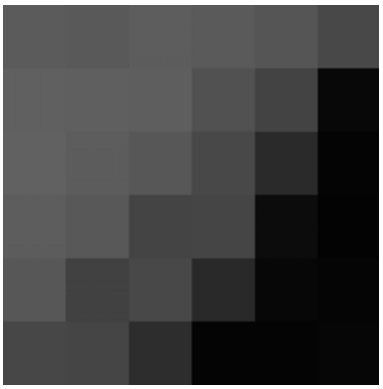
a=1, b=0.55, c=0.15, d=0.55, e=4

**Figure 7b** Processed segment A.



a=1, b=0.75, c=0.05, d=0.85, e=1

**Figure 7c** Processed segment B.



a=4, b=0.75, c=0.1, d=0.45, e=2

**Figure 7d** Processed segment C.

### 2.3 A MFS FRAMEWORK FOR EDGE DETECTION

Figure 8 shows the MFS framework for edge detection. First, a camera is used to capture the original image, *I*, from the video frame for detecting corridors. The IMU is used to detect the camera movement T, where its captured T is synchronized with the exposure duration on capturing *I*. Computation of T is discussed in Section 2.3.2.

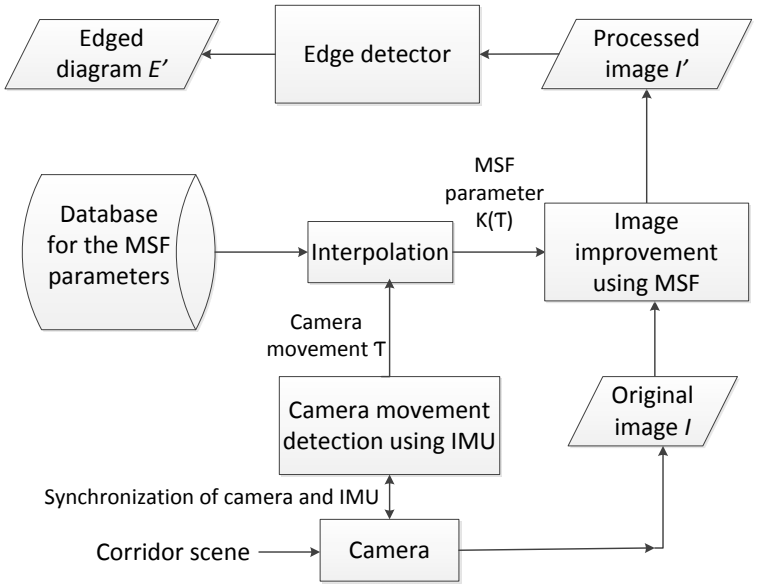



**Figure 8** MSF framework for image improvement and edge detection.

Section 2.3.1 discusses the mechanism of capturing T. Based on T, a set of MSF parameters, K(T)=[*a*(T), *b*(T), *c*(T), *d*(T), *e*(T)] is determined by interpolating with the whole set of MSF parameters which is stored in a database. The simple interpolation is used as its algorithmic complexity is lower than machine learning methods like neural networks, although machine learning methods may generate more accurate results. Section 2.3.1 discusses how the database for the MSF parameters can be developed. When the MSF is engaged with K(T), the processed image, *I'*, can be generated by the MSF with respect to *I*. An edge detector is used to generate an edge image *E'*. Hence, object edges in *E'* are more likely to be detected from *I'* rather than from *I*.

#### *2.3.1 Database for the MSF parameters*

To develop a database for the MSF parameters, a set of video frames namely $I_i$ with $i$=1,2,…$N$ is first captured by the camera, where $I_i$ is the $i$-th video frame. During the exposure time on capturing $I_i$, the IMU is used to capture the camera movement namely $T_i$ generated during the exposure for $I_i$, where each frame exposure and each movement

caption are synchronized. The synchronization mechanism can be referred to (Abhayasinghe, & Murray, 2014). Based on the collected $I_i$ and $\tau_i$, a database consisting of $N$ MSF parameters namely $K(\tau_i)=[a(\tau_i), b(\tau_i), c(\tau_i), d(\tau_i), e(\tau_i)]$ with $i=1,2,\ldots N$ is generated, where $K(\tau_i)$ corresponds to the camera movement, $\tau_i$, captured by the IMU for the $i$-th video frame, $I_i$. Using $I_i$ with $i=1,2,\ldots N$, the hand-made edge image, $E_i$, is determined manually based on human judgment. On each $E_i$, the $k$-th edge of the interested object are denoted as $e_{k,i}$, where $e_{k,i} \in E_i$.

In Figure 9, the optimization method namely particle swarm optimization (PSO) (Parsopoulos, & Vrahatis 2004) is used to generate the database for the MSF parameters, $K(\tau_i)=[a(\tau_i), b(\tau_i), c(\tau_i), d(\tau_i), e(\tau_i)]$ with $i=1,2,\ldots N$, when $e_{k,i}$ and $I_i$, are given. The PSO is used, as it is effective in solving optimization problems with discontinuous, vastly multimodal and non-differentiable landscapes (Fu, Ding, Zhou, & Hu, 2013; Hui, & Zhang, 2015; Xu, Ma, & Xu, 2015; Zeng, Liu, & Hui, 2015). In the PSO, the particle is coded with a parametrical representation of five elements, where the particle is represented as the MSF parameter, $K(\tau_i)$, and the five elements are represented as $a(\tau_i)$, $b(\tau_i)$, $c(\tau_i)$, $d(\tau_i)$ and $e(\tau_i)$, illustrated in (1a) to (1d).

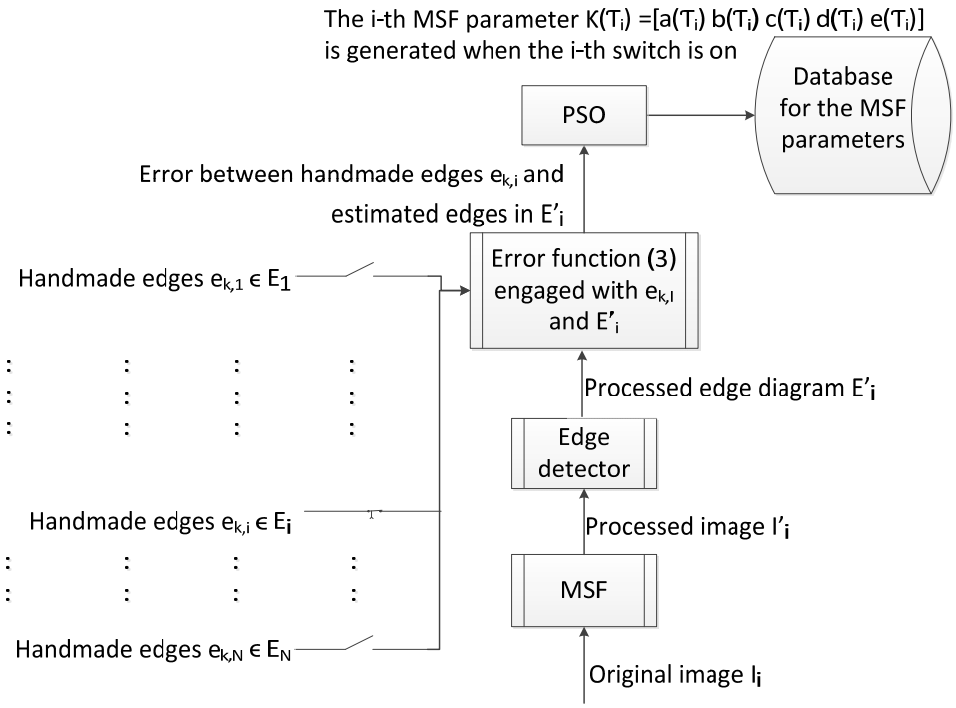



**Figure 9** Development of MSF database using the PSO.

Generally, edge detection performance is evaluated by comparing the detected edges and true edges which are manually determined by human judgement (Lopez-Molina, & Baets, 2013). Edge detectors are developed by targeting

the interested objects. For example, corridor and entry edges are importance for indoor navigation of visually impaired persons. Therefore, both corridor and entry edges are included on the hand-made edge images for developing the edge detector.

Each particle in the PSO is evaluated based on the error measure between the true edges (i.e. $e_{k,i}$) from $E_i$ and the estimated edges from the generated edge image $E_i'$, where $E_i'$ is generated by the edge detector from the processed image $I_i'$; $I_i'$ is generated by the MSF from the original image $I_i$, where the MSF parameter represented by the particle in the PSO is used. The error function (4a) (Lopez-Molina, & Baets, 2013) is formulated to evaluate the PSO particle. (4a) attempts to achieve the tradeoff between information and noisiness in the detection results. The parameters of the proposed MSF in (1) are determined by optimizing (4a).

$$D\left(E_i, E'_i\right) = \frac{1}{\sum_{e_{k,i} \in E_i} \left|e_{k,i}\right|} \cdot \left( \sum_{p \in e_{k,i}} d\left(p, E'_i\right) \right) \tag{4a}$$

where $d\left(p_1, p_2\right)$ refers to the Euclidean distance between two points $p_1$ and $p_2$ in the image; $\left|e_{k,i}\right|$ is the number of pixels of the edge $e_{k,i}$. Additionally, $d\left(p, E\right)$ can also represent the closest distance from a point $p$ to an edge in the edge image $E$. It is given as:

$$d\left(p, E\right) = \min\left\{ d\left(p, p'\right) \middle| p' \in E \right\}. \tag{4b}$$

Unlike the commonly used error measure in (Lopez-Molina, & Baets, 2013) which addresses all errors between all real and estimated edges, the numerator of (4a) addresses only the errors for the edges which are interested for navigation. Minimizing the numerator of (4a) attempts to increase the edge contrast for the interested objects on the collected video frame. Hence, the effectiveness of edge detection can be enhanced. Also the denominator of (4a) minimizes the total number of pixels of edges. It attempts to achieve a tradeoff between interested objects and uninterested objects (Elinabrouk, & Aggoun, 1998; Yitzhaky, & Peli, 2003). Here we can consider that the edges of entries or corridors are the necessary information and unnecessary edges are the noise.

The following PSO parameters were used: the particle population is 100; the maximum and minimum inertia weights are 0.9 and 0.2 respectively; the cognitive parameter and the social parameter are both set as 2 which are defaulted in (Parsopoulos, & Vrahatis, 2004). The PSO is terminated when no improvement can be obtained regarding the error measure.

### *2.3.2 Camera movement detection using IMU*

Figure 10 shows the scene plane of the image captured by the smartphone installed with a camera and an IMU, where $l$ is the distance between the scene plane and the smartphone. Figure 10 is redeveloped based on (Rajakaruna, Rathnayake, Chan, & Murray, 2014; Rajakaruna, Rathnayake, Abhayasinghe, & Murray, 2014) (Figure 2 in (Rajakaruna, Rathnayake, Abhayasinghe, & Murray, 2014) and Figure 1 in (Rajakaruna, Rathnayake, Chan, & Murray, 2014).

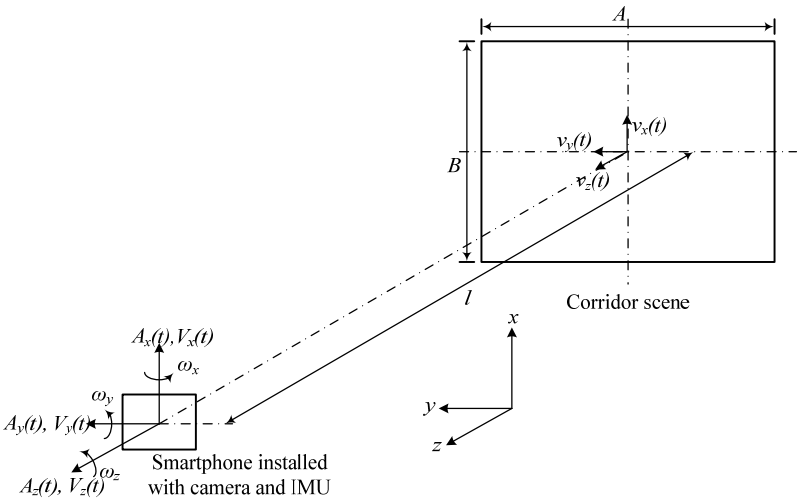



**Figure 10** Camera movement detection using IMU

The angular velocities captured by the IMU are denoted by $\omega_x(t)$, $\omega_y(t)$, and $\omega_z(t)$, with respect to the $x$, $y$, and $z$ axes respectively, where $t$ is within the exposure time for capturing the image; $t = 0, T_s, 2 \cdot T_s, \ldots, N_s \cdot T_s$; $T_s$ is the sampling period; and $N_s$ samples are captured. The linear velocities captured by the IMU are denoted by $V_x(t)$, $V_y(t)$, and $V_z(t)$, with respect to the $x$, $y$, and $z$ axes respectively. The linear velocities for the corridor scene, namely $v_x(t)$, $v_y(t)$, and $v_z(t)$, are determined based on (5a), (5b) and (5c) with respect to the $x$, $y$, $z$ axes respectively as,

$$v_x(t) = -V_x(t) + l \cdot \omega_y(t), \tag{5a}$$

$$v_y(t) = -V_y(t) - l \cdot \omega_x(t), \tag{5b}$$

$$v_z(t) = -V_z(t). \tag{5c}$$

During the exposure, the overall movement, $\tau$, for the smartphone is determined based on $v_x(t)$, $v_y(t)$, and $v_z(t)$. $\tau$ is given as:

$$\tau = \sqrt{D_x^2 + D_y^2 + D_z^2} \tag{6}$$

where $D_x$, $D_y$ and $D_z$ are the movement with respect to $x$, $y$ and $z$.

$$D_x = \sum_{t=1}^{N_s} \left( v_x(t) - v_x(t-1) \right) \cdot T_s \,, \tag{7a}$$

$$D_y = \sum_{t=1}^{N_s} \left( v_y(t) - v_y(t-1) \right) \cdot T_s \,, \tag{7b}$$

$$D_z = \sum_{t=1}^{N_s} \left( v_z(t) - v_z(t-1) \right) \cdot T_s \,. \tag{7c}$$

The detailed formulations for $D_x$, $D_y$ and $D_z$ can be referred to (Rajakaruna, Rathnayake, Chan, & Murray, 2014; Rajakaruna, Rathnayake, Abhayasinghe, & Murray, 2014).

## 3. EXPERIMENTAL METHODOLOGY AND RESULTS

Experiments are performed to evaluate the performance of the proposed MSF framework in relation to benchmarking techniques. In this section, we provide a detailed description of the experimental methodology and results. Section 3.1 presents the experimental conditions for the indoor environments, simulation platform and hardware used for collected image and IMU data. Sections 3.2 and 3.3 present the results obtained for small and large camera movements respectively.

### *3.1 Experimental methodology*

We used the Sony Xperia TX smartphone equipped with a camera and an IMU to capture the video sequences and the movements associated with the camera motion respectively, with a frame rate of 14 frames per second in the camera and a sampling time of 20ms in the IMU.

Two video sequences associated with two IMU datasets were collected. The first sequence, namely Whitecane, represents a blind person walking with a white cane. The second sequence, namely Guidedog, represents a blind person walking with a guide dog. For both Whitecane and Guidedog, the smartphone was hung around the person's neck to capture the video sequences and the camera movement data. The frame size of both sequences was 240×320 pixels. For each sequence, 200 video frames were captured, where the first 100 frames were used as a training set for developing the MSF database and the last 100 frames were used as a validation set to evaluate the generalization capability of the proposed MSF framework.

Figure 11 shows the amount of camera movement captured for the Whitecane and the Guidedog [1]. We can see that the camera movements of Whitecane are smaller than those of Guidedog. When capturing the Whitecane, the person walked slowly and swigged with the whitecane slowly in order to seek obstacles. When capturing the Guidedog, the person was led by a fast-paced guide dog, which caused additional camera movement. As a result, more blur was generated in the Guidedog sequence than in the Whitecane sequence. Therefore, the Whitecane sequence evaluates the performance of the proposed MSF framework when the camera movement is small and the blur is little. The Guidedog sequence evaluates the performance when the camera movement is large and the blur is high. For both Whitecane and Guidedog sequences, we consider the corridor and entry detections which are both important for indoor navigation by visually impaired persons (Shi, & Samarabandu, 2006; Tian, Yang, & Arditi, 2010). Therefore four detection scenarios are considered.

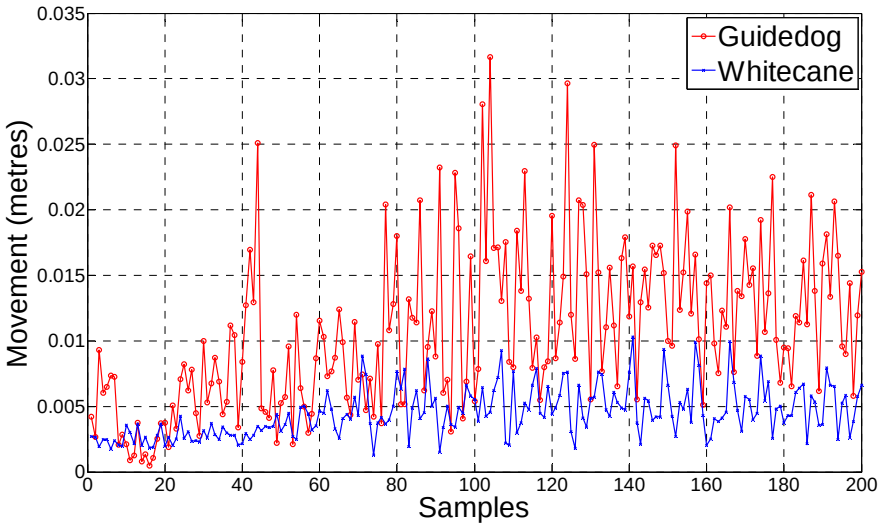



**Figure 11** Camera movement for Whitecane and Guidedog.

To evaluate the proposed MSF framework, each of the four commonly used edge detectors, namely Canny algorithm (Canny 1986), Roberts algorithm (Lee, Haralick, & Shapiro, 1987), Sobel algorithm (Lee, Haralick, & Shapiro, 1987) and zerocross algorithm (Kimmel, & Bruckstein, 1984) is used within the framework, where the threshold for handling images with no blurs is used on the algorithms. The MSF framework is evaluated by performing edge detection for the validation sets for both Whitecane and Guidedog. In this research, all algorithms are implemented and evaluated using Matlab 7.7 on a PC which has a CPU of Intel(R) Core(TM)2 Duo 2.66GHz and a memory of 8 GB.

[1] The videos and IMU data can be collected from https://dl.dropboxusercontent.com/u/48145823/Video_2013_10_24_8_38_31_760.zip

### *3.2 Experimental Results for small camera movement*

We consider the Whitecane sequence, which exhibits small camera movement. The MSF framework is aimed for the both corridor and entry detections. To compare the effectiveness of the proposed MSF framework, two sets of results were considered. One set was generated by solely using the edge detectors. Another set was generated by using the MSF framework. This enabled us to evaluate the performance when solely using edge detectors and that when using the MSF framework.

#### *3.2.1 Corridor detection*

Figure 12a shows the 1st video frame from the validation set. We can see from the figure that no serious blur exists. Figures 12b shows the edge diagram generated by the Canny algorithm and Figure 12c shows the edge diagram which is generated based on the image processed by the MSF framework. Both edge diagrams show that the corridor edges can be correctly detected. As the blur captured on the original frame was little, both the Canny algorithm and the MSF framework work effectively. Similar results are obtained when the Roberts algorithm, Sobel algorithm and zerocross algorithm are used. Figure 13 shows the distance error formulated in (3) which evaluates the error between the true corridor edges and the estimated corridor edges. It shows the average distance errors over the 100 video frames of the validation set. Generally, similar results are produced when solely using the edge detectors and when using the MSF framework.

#### *3.2.2 Entry detection*

The MSF framework is used to perform entry detection for the Whitecane sequence, where the camera movement is small. The distance error in (3) is formulated by evaluating the error between the true entry edges and the estimated entry edges. The entry detection results obtained by the MSF framework are compared with those obtained by purely using the edge detection algorithms, Canny, Roberts and Sobel algorithms. Figure 14 shows that similar errors are obtained by both using the MSF framework and purely using the detection algorithms. Therefore, these results further demonstrate that similar performance is achieved by solely using the edge detectors and using the MSF framework, when the camera movement is small.

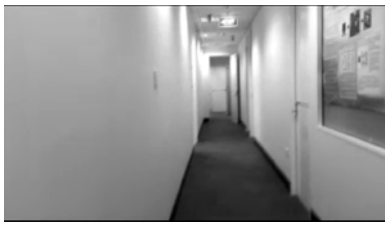


**Figure 12a** Original video frame.

**Figure 12b** Edge image generated by the Canny algorithm.

**Figure 12c** Edge image generated by the MSF framework.

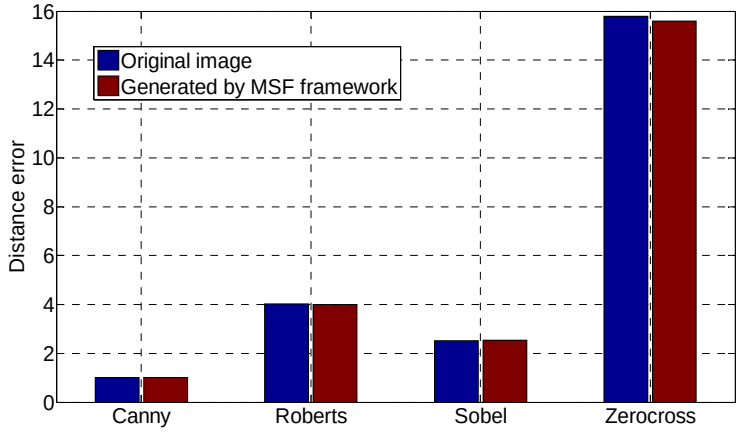



**Figure 13** Corridor detection: Distance errors for Whitecane

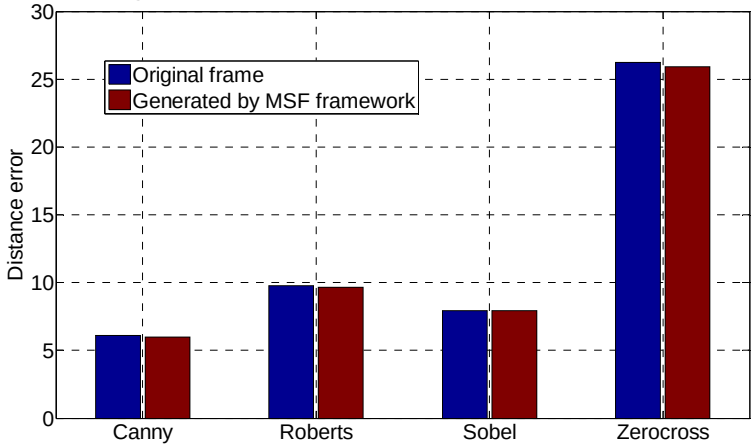



**Figure 14** Entry detection: Distance errors for Whitecane

### *3.3 Experimental Results for Large camera movement*

We consider the Guidedog sequence, which exhibits large camera movement. The effectiveness of MFS framework is evaluated by both corridor and entry detections.

#### *3.3.1 Corridor detection*

Figure 15a shows the 1st original video frame from the validation set. Figure 15a shows the edge diagram generated by solely using the Canny algorithm. The edge diagram shows that the left corridor is missing on the edge image generated based on the original frame. Therefore, the sole use of the Canny algorithm is inadequate for generating correct corridor edges, and thus image improvement is required for large camera movement. Apart from using the

MSF framework to improve images before performing edge detections, two deblurring filters, namely the deconvolution filter (Horstmeyer 2010) and the Whyte et al.'s filter (Whyte, Sivic, Zisserman, & Ponce, 2012), are used for image enhancement. The deconvolution filter is used as it is implemented on smartphones to improve blurred images (Joshi, Kang, & Zitnick, 2010). The deconvolution filter removes the image blur before performing edge detection. The deblurred image can contribute a better edge detection as the blur is removed. Hence, the edge image developed with the deblurred image is compared with that developed by the proposed MSF framework. Also, Whyte et al.'s filter (Whyte, Sivic, Zisserman, & Ponce, 2012) is used to enhance the image quality before performing the edge detection, as Whyte et al.'s filter has recently been developed for image deblurring.

To evaluate the effectiveness of the proposed MSF framework, four sets of results are considered:

a) the one based on solely using the edge detectors, Canny algorithm (Canny 1986), Roberts algorithm (Lee, Haralick, & Shapiro, 1987), Sobel algorithm (Lee, Haralick, & Shapiro, 1987) and zerocross algorithm (Kimmel, & Bruckstein, 1984);
b) the one generated by the deconvolution filter (Horstmeyer 2010);
c) the one generated by the Whyte et al. filter (Whyte, Sivic, Zisserman, & Ponce, 2012);
d) the one generated by the proposed MSF framework.

These four sets of results are represented with a, b, c and d in Figures 15 to 17, where Figure 15 show the images and Figures 16 and 17 show the edge diagrams. These edge diagrams shown in Figures 16 and 17 are developed by the Canny algorithm and Sobel algorithm respectively. We show the edge diagrams generated by the Canny algorithm and the Sobel algorithm, as the Canny algorithm is generally more effective than the other three edge detection algorithms, and the Sobel algorithm is computationally simpler than the other three algorithms (Canny 1986).

**Figure 15a** Corridor detection: Original image.

**Figure 15b** Corridor detection: Processed image using the deconvolution filter.

**Figure 15c** Corridor detection: Processed image using Whyte et al. filter.

**Figure 15d** Corridor detection: Processed image using the proposed MFS framework.

**Figure 16a** Corridor detection: Canny algorithm for the original image.

**Figure 16b** Corridor detection: Canny algorithm after the deconvolution filter.

**Figure 16c** Corridor detection: Canny algorithm after Whyte et al. filter.

**Figure 16d** Corridor detection: Canny algorithm within the proposed MFS framework.

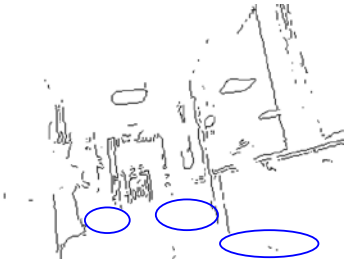


**Figure 17a** Corridor detection: Sobel algorithm for the original image.

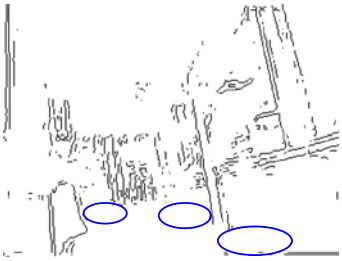


**Figure 17b** Corridor detection: Sobel algorithm after the deblurring algorithm.

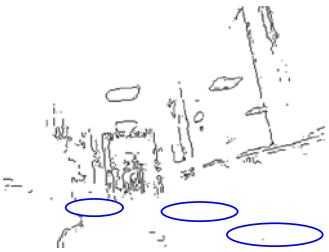


**Figure 17c** Corridor detection: Sobel algorithm after Whyte et al. filter.

**Figure 17d** Corridor detection: Sobel algorithm within the proposed MFS framework.

Figures 16a-16d show the edge diagrams generated by the Canny algorithm. As mentioned, some corridor edges are missing in Figure 16a which is generated based on the original video frame. Figure 16b shows the edge diagram associated with the deconvolution algorithm. Although the edge diagram shows that the right corridor exists, the left corridor cannot be seen on the edge diagram. Also, a lot of noise and artifact is generated on the edge diagram. Figures 16c and 16d show the edge diagrams associated with the Whyte et al. filter and the MSF framework. The two edge diagrams show that both left and right corridors exist. Hence, clearer corridor edges are generated when the Whyte et al. filter and the MSF framework are used.

Also, Figures 17a-17d show the edge diagrams generated by the Sobel algorithm. Figures 17a-17c show that both left and right corridors are missing in the edge diagrams, where they are generated with the original images and the processed images which are associated with the deconvolution and the Whyte et al. filters. Figure 17d shows that the right corridor exists and partial left corridor exists, where the edge diagram was generated based on the MSF framework. Hence, the MSF framework improves the effectiveness of detecting corridor edges. Similar results have been found for Roberts and zerocross algorithms. Due to the page limit of the paper, those results are not reported here.

Figure 18 shows the distance measure formulated in (3) where the corridor edges are considered. It shows the average distance errors over the 100 video frames of the validation set. It shows that the deconvolution filter, Whyte et al.'s filter and the MSF framework can produce edge diagrams with comparably smaller errors to the edge diagrams

based on the original video frames. Also, slightly smaller errors can be obtained based on the MSF framework compared with the deconvolution filter and the Whyte et al filter.

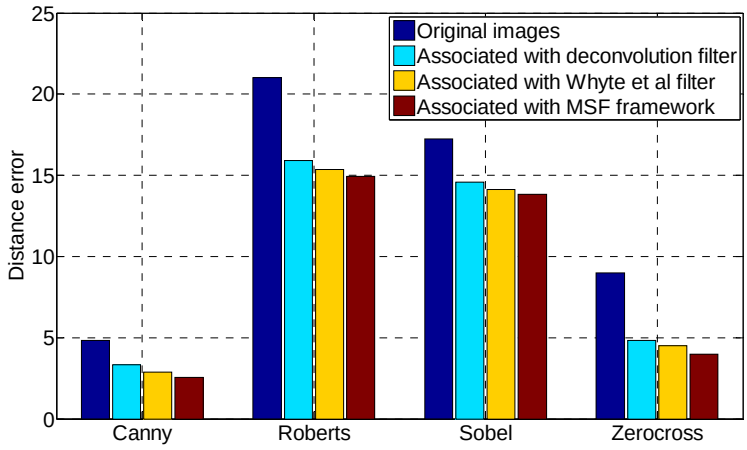



**Figure 18** Corridor detection: Distance errors for Guidedog.

Figure 19 shows the average computational time used for the four approaches in performing edge detections on the validation set. It shows that the two deblurring methods, deconvolution filter and Whyte et al.'s filter, require considerably more computational time than that required by the proposed MSF framework, where the deconvolution filter required about 0.2 seconds; the Whyte et al. filter required about 15 seconds; and the proposed MSF framework required only 0.04 seconds which is much lesser than the other two methods. The computational time required by the Whyte et al. filter is much greater than the other methods. Only slightly more computational time is required by the MSF framework than that required when solely using the edge detectors. Therefore, the MSF framework is effective on corridor detection, as better edge detection can be achieved and only marginally additional computational time is required compared with solely using the edge detectors.

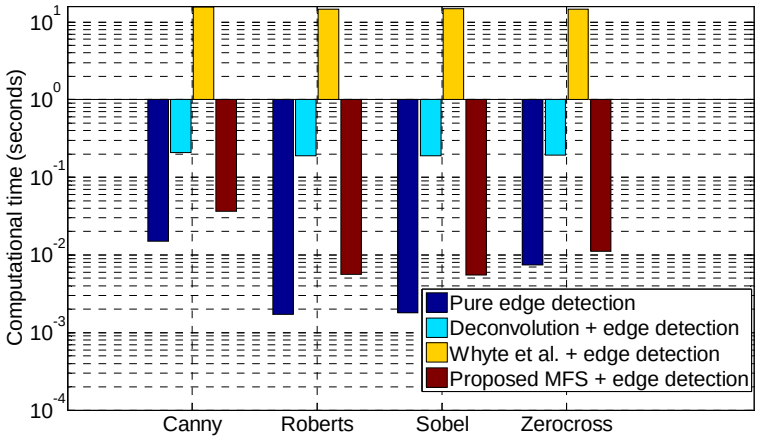



**Figure 19** Corridor detection: Computational time for Guidedog.

### *3.3.2 Entry detection*

The 181th original video frame from the validation set is shown in Figure 20a. We attempt to detect the entry circled on the image. Figure 21a shows the edge diagram which is generated solely by the Canny algorithm and it shows that entry patterns can barely be identified. Therefore, the original image must be processed before performing the edge detection, as the blur is generated by large camera movement and it degrades the effectiveness of entry detection. Figures 20b, 20c and 20d show the enhanced images generated by the deconvolution filter (Horstmeyer 2010), Whyte et al.'s filter (Whyte, Sivic, Zisserman, & Ponce, 2012) and the proposed MFS framework respectively. We can see that sharper entry can be identified from the three processed images compared with the original image. Based on the three enhanced images, the Canny algorithm is used for edge detection and the three edge diagrams corresponding to the three processed images are shown in Figure 21b, 21c and 21d. We can observe from the three edge diagrams that the entry patterns are more likely to be observed from the processed images. However, noise and artifact is generated on the edge diagrams associated with the deconvolution filter (Horstmeyer 2010) and Whyte et al.'s filter (Whyte, Sivic, Zisserman, & Ponce, 2012). Clearer entry patterns with less noise and artifact can be generated on the edge diagram associated with the proposed MSF framework.

Also, Figures 22a to22d show the edge diagrams generated by the Sobel algorithm. The entry patterns can barely be identified on Figures 22a and 22b which correspond to the original image and are associated with the deconvolution filter respectively. Although the Whyte et al. filter can generate a clearer entry pattern as shown in Figure 22c, noise and artifact are generated. Figure 22d shows the edge diagram associated with the proposed MFS framework and it shows the clearer entry patterns with less noise and artifact compared with the other edge diagrams. Similar results have been found for the Roberts and zerocross algorithms. Due to space constraints, those results are not shown here.

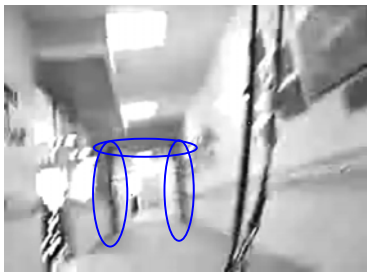


**Figure 20a** Entry detection: Original video frame.

**Figure 20b** Entry detection: Processed frame using the deconvolution filter.

**Figure 20c** Entry detection: Processed frame using Whyte et al.'s filter.

**Figure 20d** Entry detection: Processed frame using the proposed MFS framework.

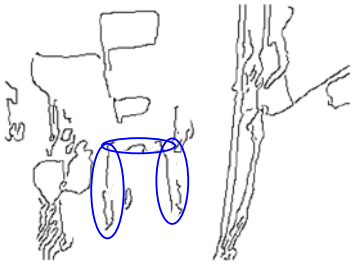


**Figure 21a** Entry detection: Canny

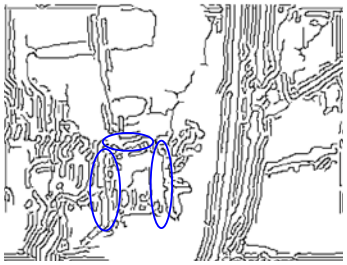


**Figure 21b** Entry detection: Canny

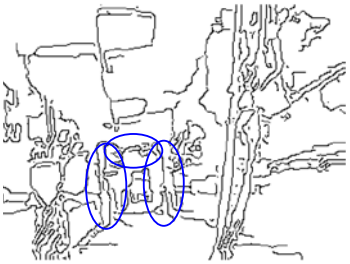


**Figure 21c** Entry detection: Canny

**Figure 21d** Entry detection: Canny

algorithm for original video frame. algorithm after deconvolution filter. algorithm after Whyte et al.'s filter. algorithm within proposed MFS framework.

**Figure 22a** Entry detection: Sobel algorithm for original video frame.

**Figure 22b** Entry detection: Sobel algorithm after deconvolution filter.

**Figure 22c** Entry detection: Sobel algorithm after Whyte et al.'s filter.

**Figure 22d** Entry detection: Sobel algorithm within proposed MFS framework.

Figure 23 shows the average distance errors formulated in (3) over the 100 original images of the validation set, where the entry edges are considered. It shows the errors for the original images and the three processed images associated with the deconvolution filter, Whyte et al. filter and the proposed MSF filter. Results show that smaller errors can be obtained on the original images comparing with the processed images. Also, the proposed MSF framework can obtain slightly smaller errors than those obtained by the proposed MSF framework, and it obtained similar errors compared with the Whyte et al.'s filter which is a recently developed deblurring method.

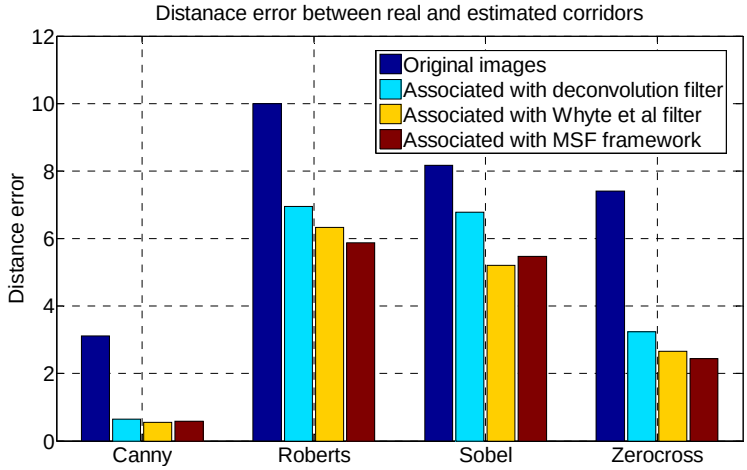



**Figure 23** Entry detection: Distance errors for Guidedog.

Figure 24 shows the average computational time used for edge detection corresponding to the original images and the processed images associated with the deconvolution filter, Whyte et al.'s filter and the proposed MSF framework. Similar to the computational time used on corridor detection, Whyte et al.'s filter requires much more computational time than the deconvolution filter which requires more computational time than the proposed MSF framework. Hence, the proposed MSF framework requires the smallest amount of computational time compared with the other enhancement methods. Therefore, similar to corridor detection, better edge detection can be achieved by the proposed MSF framework and only marginally additional computational time is required compared with solely using the edge detectors.

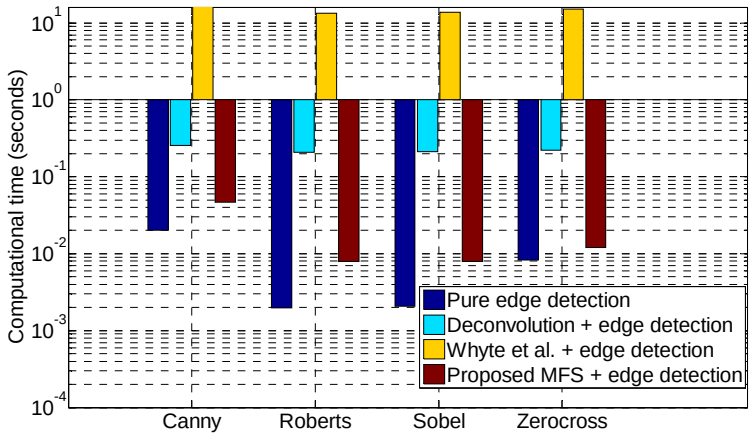



**Figure 24** Entry detection: Computational time for Guidedog.

In this research, the proposed MSF framework and the other tested methods are implemented on a PC platform as described in Section 3.1. Results show that smaller detection errors are obtained by the MSF framework and smaller computational time is required when performing detection on large movement. As the MSF framework and the other tested methods are deterministic algorithms, same algorithmic steps are performed when different platforms are implemented. Same results in term of the detection errors can be obtained. Therefore, this is expected that the MSF framework can obtain small errors for large movement, when different platforms are implemented.

Results also show that faster computational time is required by the MSF framework, when they are implemented in PC. Hence, the computational complexity of the MSF framework is smaller than the other tested methods. As the tested methods and the MSF framework are all deterministic algorithms, their asymptotic computational complexities are the same when implementing on different platforms. Therefore, it can be expected that the MSF framework is faster than the other tested methods when different platforms are implemented.

## 4. Conclusion and Future Work

In this paper, a modified sigmoid function (MSF) framework was proposed based on the inertial measurement unit (IMU) data in order to assist indoor navigation. It attempted to enhance the navigation effectiveness for visually impaired persons who use smartphones for indoor navigation. The MSF framework improves the contrast of the video frames containing motion blur, resulting in more precise edge detection. It overcomes the limitation of deblurring algorithms, which are computationally expensive and are not practical for real-time implementations.

The effectiveness of the MSF framework was evaluated based on four scenarios involving visually impaired persons engaged in navigation. One used a white cane and another used a guide dog for corridor and entry detections which are important for indoor navigation of visually impaired persons. The detections with a white cane exhibited a small movement and thus small blur was generated. The detections with a guide dog exhibited a large movement and

thus large blur was generated. Four commonly-used edge detectors were implemented within the MSF framework for performance evaluation. Three approaches were considered: solely using the edge detectors; using two deblurring algorithms including the deconvolution filter and Whyte et al.'s filter; and using the proposed MSF framework, where the deconvolution filter is commonly used on image deblurring on smartphones and the Whyte et al.'s filter is a recently developed deblurring method.

For the video frames containing strong blur, results showed that the proposed MSF framework is an improvement on existing techniques with regard to both edge detection performance and computational complexity. Specifically, smaller errors for detections of indoor corridors and entries were obtained compared to existing benchmark methods. Furthermore, the MSF framework performed similarly well to an advanced system based on deblurring, while maintaining an average 10-fold smaller computational complexity than the deconvolution filter and 100-fold smaller than Whyte et al.'s filter.

Future work is to develop a database which includes visual and movement data for navigation of visually impaired persons, as this database is not available in the public. This data will be collected from visually impaired persons who simulate navigations in different indoor environments and with different movement levels. This database will consist of video sequences synchronized with the IMU data which are captured by a mobile device installed with a camera and a synchronized IMU. Using this database, the effectiveness of the MSF framework can be further evaluated. Also this database can be used as a benchmark to evaluate effectiveness of image enhancement algorithms which are particularly developed for indoor navigation of visually impaired persons.

The work presented in this paper aims to develop an image enhancement algorithm in order to help the edge or object detection. The final goal is to build a comprehensive navigation system for blind and vision-impaired people using arrange sensors in a hand held devices. In the next stage, we will implement the algorithm in a hand held device for real time evaluation.

## REFERENCES

Abhayasinghe, N., & Murray, I. (2014). Human Activity Recognition Using Thigh Angle Derived from Single Thigh Mounted IMU Data. *The International Conference on Indoor Positioning and Indoor Navigation*, 111-115.

Antoun, S.M., & McKerrow, P.J. (2007). Perceiving a corridor with CTFM ultrasonic sensing. *Proceedings of the Australasian Conference on Robotics and Automation*.

Asad, M., & Ikram, W. (2012). Smartphone based guidance system for visually impaired person. *International Conference Image Processing Theory, Tools and Applications*, 442-447.

Braun, G.J., & Fairchild, M.D. (1999). Image lightness rescaling using sigmoidal contrast enhancement function. *Journal of Electronic Imaging,* 8(4), 380-393.

Canny, J. (1986). A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 8(6), 679-698.

Cho, S., Wang, J., & Lee, S. (2012). Video deblurring for hand-held cameras using patch based synthesis. *ACM Transactions on Graphics,* 31(4), 1-9.

Deng, X., Shen, Y., Song M., Tao, D., Bu, J., & Chen, C. (2012). Video based nonuniform object motion blur estimation and deblurring, *Neurocomputing,* 86, 170-178.

Douglas, S.C., & Meng, T.H.Y. (1989). An adaptive edge detection method using a modified sigmoid-LMS algorithm, *Twenty-Third Asilomar Conference on Signals, Systems and Computers*, 252-256.

Elinabrouk, A., & Aggoun, A. (1998). Edge detection usin local histogram analysis. *Electronics Letters*, 34 (12), 1216-1217.

Fu, Y., Ding, M., Zhou, C., & Hu, H. (2013). Route planning for unmanned aerial vehicle (UAV) on the sea using hybrid differential evolution and quantum-behaved particle swarm optimization. *IEEE Transactions on Systems, Man and Cybernetics: Systems,* 43(6), 1451-1465.

Guerrero, L.A., Vasquez, F., & Ochoa, S.F. (2012). An indoor navigation system for visually impaired. *Sensors,* 12, 8236-8258.

Horstmeyer, R. (2010). *Camera motion tracking for deblurring and identification*, MIT Media Laboratory, MIT.

Huang, Y., & Fan, N. (2011). Inter-frame information transfer via projection onto convex set for video deblurring. *IEEE Journal of Selected Topics in Signal Processing,* 5(2), 275-284.

Hui, Q., & Zhang, H. (2015). Optimal balanced coordinated network resource allocation using swarm optimization. *IEEE Transactions on Systems, Man and Cybernetics: Systems,* 45(5), 770-787.

Ivanchenko, V., Coughlan, J., & Shen, H. (2009). Staying in the crosswalk: A system for guiding visually impaired pedestrians at traffic intersections. *Assistive Technology Research Series,* 25, 69.

Ivanchenko, V., Coughlan, J., & Shen, H. (2010). Real-time walk light detection with a mobile phone. *Computers Helping People with Special Needs*, 229–234.

Kang, M.C., Chae, S.H., Sun, J.Y., Yoo, J.W., & Ko, S.J. (2015). A Novel Obstacle Detection Method based on Deformable Grid for the Visually Impaired. *IEEE Transactions on Consumer Electronics*, 61 (3), 276-383.

Kannan, P., Deepa, S., & Ramakrishnan, R. (2010). Contrast enhancement of sports images using modified sigmoid mapping function. *IEEE Conference of Communication Control and Computing Technologies*, 651-656.

Kimmel, R., & Bruckstein, A.M. (1984). On regularized Laplacian zero crossings and other optimal edge integrators. *International Journal of Computer Vision,* 53(3), 225-243.

Joshi, N., Kang, S.B., & Zitnick, C. L. (2010). Image deblurring using inertial measurement sensors. *ACM Transactions on Graphics,* 29(4), 1-9.

Lane, N. D., Miluzzo, E., Lu, H., & Peebles, D. (2010). A survey of mobile phone sensing. *IEEE Communications Magazine,* 48(9), 140-150.

Lee, D. B., Jeong, S. C., Lee, Y. G., & Song, B. C. (2013). Video deblurring algorithm using accurate blur kernel estimation and residual deconvolution based on a blurred unblurred frame pair. *IEEE Transactions on Image Processing,* 22(3), 926-640.

Lee, J.S.J., Haralick, R.M., & Shapiro, L.G. (1987). Morphologic edge detection. *IEEE Journal of Robotics and Automation,* 3(2), 142-156.

Lopez-Molina, C., & Baets, B.D. (2013). Quantitative error measures for edge detection. *Pattern Recognition,* 46(4), 1125-1139.

Martin, D., Fowlkes, C., & Malik, J. (2004). Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence,* 26(5) 530-549.

Mascetti, S., Ahmetovic, D., Gerino, A., Bernareggi, C., Busso, M., & Rizzi, A. (2016). Robust traffic lights detection on mobile devices for pedestrians with visual impairment. *Computer Vision and Image Understanding*, 148, 123–135.

Mhaned, I. B., Abid, S., & Fnaiech, F. (2012). Weld defect detection using a modified anisotropic diffusion model. *EURASIP Journal on Advances in Signal Processing,* 46, 1-12.

Moreno, M., Shahrabadi, S., Jose, J., Buf, J.M.H.D., & Rodrigues, J.M.F. (2012). Realtime local navigation for the blind: detection of lateral doors and sound interface. *Procedia Computer Science,* 14, 74-82.

Guide Dog News (G.D.N.), (2013). Guide dog training on display for new education campaign. *Guide Dogs NSW/ACT*, 23 April.

Pham, H.H., Le, T.L., & Vuillerme, N. (2016). Real-time obstacle detection system in indoor environment for the visually impaired using microsoft kinect sensor. *Journal of Sensors*.

Parsopoulos, K. E., & Vrahatis, M. N. (2004). On the computation of all global minimizers through particle swarm optimization. *IEEE Transactions on Evolutionary Computation,* 8(3), 211-224.

Peli, T. (1990). Contrast in complex images. *Journal of the Optical Society of America A,* 7(10), 2032-2040.

Peng, E., Peursum, P., Li, L., & Venkatesh, S. (2010). A smartphone based obstacle sensor for visually impaired. *7th International Conference on Ubiquitous Intelligence and Computing*, (pp. 590-604).

Pradeep, V., Medioni, G., & Weiland, J. (2010). Robot vision for the visually impaired. *Proceedings of IEEE on Computer Vision and Pattern Recognition - Workshops*, 15–22.

Pulli, K., Baksheev, A., Kornyakov, K., & Eruhimov, V. (2012). Real-time Computer Vision with OpenCV, *Communications of the ACM,* 55(6), 61-69.

Rajakaruna, N., Rathnayake, C., Chan, K.Y., & Murray, I. (2014). Image deblurring for navigation systems of vision impaired people using sensor fusion data. *International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, 1-6.

Rajakaruna, N., Rathnayake, C., Abhayasinghe, N., & Murray, I. (2014). Inertial Data Based Deblurring for Vision Impaired Navigation. *International Conference on Indoor Positioning and Indoor Navigation*, 416-420.

Rodriguez-Sanchez, M. C., Moreno-Alvarez, M.A., Martin, E., Borromeo, S., & Hernandez-Tamames, J.A. (2014). Accessible smartphones for blind users: a case study for a wayfinding system. *Expert Systems with Applications,* 41, 7210-7222.

Segvic, S., & Ribaric, S. (2001). Determining the absolute orientation in a corridor using projective geometry and active vision. *IEEE Transactions on Industrial Electronics,* 48, 696-710.

Shen H., Chan, K., Coughlan, J., & Brabyn, J. (2008). A mobile phone system to find crosswalks for visually impaired pedestrians. *Technology and disability*, 20(3), 217-224.

Shi, W.X., & Samarabandu, J. (2006). Investigating the performance of corridor and door detection algorithms in different environments. *in IEEE International Conference on Information and Automation*, 15-17.

Solymár, Z., Stubendek, A., Radványi, M., & Karacs, K. (2011). Banknote recognition for vi- sually impaired. *Proceedings of European Conference on Circuit Theory and Design*, 841–844.

Tian, Y., Yang, X., & Arditi, A. (2010). Computer vision based door detection for accessibility of unfamiliar environments to blind persons. *in the 12th International Conference on Computers Helping People with Special Needs*, 263-270.

Venkatesh, S., & Rosin, P.L. (1995). Dynamic threshold determination by local and global edge evaluation. *Graphical Models and Image Processing,* 57(2), 146-160.

Whyte, O., Sivic, J., Zisserman, A., & Ponce, J. (2012). Non-uniform deblurring for shaken images, *International Journal of Computer Vision,* 98(2), 168-186.

Who Health Organization (W.H.O.). Visual impairment and blindness. (2014). http://www.who.int/mediacentre/factsheets/fs282/en/.

Williams, R. (2014) Google Project Tango smartphone creates 3D maps of user surroundings. *The Telegraph*, 21 February.

Winlock, T., Christiansen, E., & Belongie, S. (2010). Toward real-time grocery detection for the visually impaired. *Proceedings of IEEE on Computer Vision and Pattern Recognition - Workshops*, 49–56.

Xu, J., Ma, Y., & Xu, Z. (2015). A bilevel model for project scheduling in a fuzzy random environment. *IEEE Transactions on Systems, Man and Cybernetics: Systems,* 45(10), 1322-1335.

Yang, Z.F., & Tsai, W.H. (1999) Viewing corridors as right parallelepipeds for vision-based vehicle localization. *IEEE Transactions on Industrial Electronics,* 46, 653-661.

Yitzhaky, Y., & Peli, E. (2003). A method for objective edge detection evaluation and detector parameter selection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25 (8), 1027-1033.

Zeng, X., Liu, Z., & Hui, Q. (2015). Energy equipartition stabilization and cascading resilience optimization for geospatially distributed cyber-physical network systems. *IEEE Transactions on Systems, Man and Cybernetics: Systems,* 45(1), 25-43.