Edge Based Obstacle Detection Model Focused on Indoor Floor-Based Obstacles

Khairul Azim Bin Za'aba
Faculty of Engineering, Computing and Science
Swinburne University of Technology
Kuching, Sarawak
kazim@swinburne.edu.my

Lau Bee Theng
Faculty of Engineering, Computing and Science
Swinburne University of Technology
Kuching, Sarawak
blau@swinburne.edu.my

***Abstract*— There are several obstacle detection and avoidance technologies developed to assist the visually impaired community in the current market. They are categorised into different types such as non-vision or vision-based technologies. However, a majority of these technologies are not ready for distribution as it may require additional steps due to mass production and distribution. This paper introduces a simple obstacle detection model built for mobile device. Utilising the Canny Edge detection algorithm, the edge information is used to compare whether the user's walking path is obstructed by obstacles. The proposed model utilises a mobile device's proximity sensor and accelerometer to adapt itself accordingly in various environment. The model is tested in various scenarios resulted to an average obstacle detection rate of 85%. The model may be used as a supplementary assistance to the white cane, for people with visual impairment.**

***Keywords*— Optical imaging, visual impairment, obstacle detection, android, monocular camera.**

I. INTRODUCTION

A survey done in 2015 showed that there is at least a total of 253 million people who are visually impaired and 36 million from that number who are entirely blind [3]. There are a few types of visual impairments which are commonly categorised into low vision and no vision [8]. This research focused on aiding the low vision community, which is the community with visual impairments that cannot be corrected or improved with regular prescription glasses.

The white cane is known as the main assistance used by the visually impaired people to detect and avoid obstacles. Other than this, there are limited options for other supplementary assistive gadgets available. This problem arises when not all developed methods or models by the researchers are readily available to be used by the community, as a lot of them require monetary investments, mass production and distribution to the people. The researchers face this problem when their developed models require specific hardware or contraptions of different hardware to provide the expected returns.

One of the examples of contraptions built to provide obstacle detection was combining the usage of Kinect camera to a portable computer [1], and a combination of different non-vision-based sensors such as ultrasonic and infrared sensors onto the white cane as demonstrated by the Smart Walking Stick [12]. Thus, they require repackaging into smaller form factor for better mobility and manufacturing before they could be distributed [9]. The processing power required for the existing model to work was one of its weaknesses. To accommodate the required high computational or large memory requirements, larger devices are used. Also, models which use databases or deep learning approach would require access to an external server to accommodate its processing needs [10]. Before high-powered mobile devices were introduced, only large-sized portable devices such as a laptop or Intel NUC could utilise those models. These devices commonly require additional external sensors since they do not equip with suitable sensors for real time obstacle detection.

II. BACKGROUND

In this research, light weight mobile app running on mobile devices such as smartphones and tablet computers is the main focus. Mobile devices are commonly equipped with multiple sensors, which are suitable to be used in real time obstacle detection. Furthermore, a report by the government stated that there are 80% of Malaysians use smartphones with internet connectivity in 2016 and the numbers are growing [11].

However, due to the affordability and costs of high end mobile devices, the proposed model is designed to run on low cost mobile devices. In this research, the edge detection approach is selected as it only requires a small amount of processing power to perform real time detection [7]. Thus, it is feasible to develop an edge-based obstacle detection model which can be used on a low end mobile device to provide supplementary assistance to the visually impaired.

Numerous models have been reviewed in the process of conducting the research. However, the models documented in this paper is the one which affects the research process the most. This is because of the space limitation introduced in the paper.

In [4], an algorithm that exploits both the monocular vision camera of a smartphone and inertial measurements registered from its gyroscope to create an estimation of a 3D sparse depth map of the environment is developed (Figure 10).

Stimulating and estimating depth map is difficult by using monocular vision smartphone cameras as there is no initial depth information is provided by the camera [13]. Thus, the method requires heavy processing, only one frame is processed within 2 seconds using a low end smartphone, LG Nexus 5.

A similar technique could be stimulated by edge detection approach. Instead of creating a depth map, changes of edges information could be compared to detect obstacles. In comparison to the 3D depth map estimation algorithm, edge-based approach minimises the processing power required. Common mobile device sensors could help to increase detection accuracy.

978-1-5386-8546-4/19/$31.00 ©2019 IEEE

Authorized licensed use limited to: UNIVERSITY OF ULSTER. Downloaded on September 30,2022 at 10:06:17 UTC from IEEE Xplore. Restrictions apply.

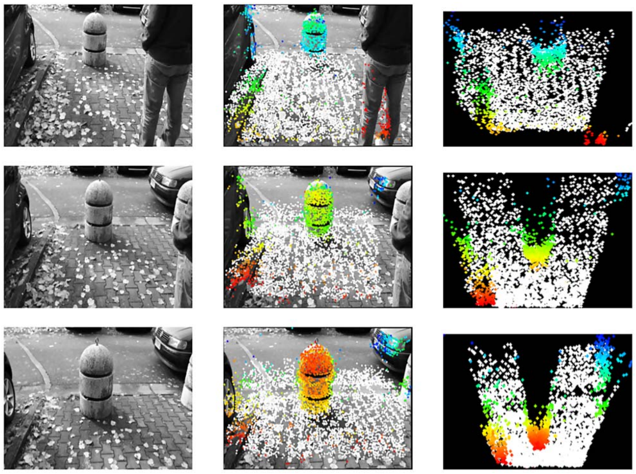
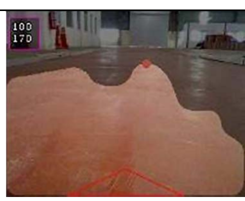


Fig. 1. Obstacle detection example - Original image (first column), Sparse depth map with ground-plane points in white (second column), bird's eye view (third column) [4]

Obstacle detection is used in gadgets such as automated vehicles or robots for collision avoidance. However, models created for those can also be modified for obstacle detection for the visually impaired. Taha and Jizat [14] from University Malaysia Pahang published a comparison between two approaches for collision avoidance in Automated Guided Vehicle. It compares the edge detection approach using Canny edge detection algorithm and floor sampling approach.

TABLE I
COMPARISON OF NAVIGATION POINT BETWEEN FLOOR SAMPLING APPROACH AND CANNY EDGE [14]

Time	Floor Sampling	Canny Edge Detection
Morning		
Night		

The experiment shown in Table 1 done by Taha and Jizat [14], showed that the Canny Edge detection approach in obstacle detection and avoidance is a better solution in comparison to Floor sampling. Both approaches could produce reliable results but, Canny edge detection is more versatile in comparison to Floor Sampling method. This is because edges could be detected in almost all lighting conditions, but area sampling could have resulted in wrong pixel values captured to compare with the rest of the image [14].

As lighting affects Canny Edge detection algorithm accuracy as mentioned by Taha and Jihat [14], by using an extra sensor such as lighting sensor, the sensitivity of Canny Edge detection algorithm can be changed to accommodate different lighting situations to increase the accuracy of the detection which applies in the current model introduced in this paper.

The next approach of providing obstacle detection was made by Castells, Rodrigues, and Du Buf [6] as part of their SmartVision project. The model detects sidewalk borders which act as its Obstacle Detection Windows (ODW). There are two different methods of obstacle detection applied to the ODW. The first one would be to count variations of grey value on each horizontal line inside the ODW, while the second method is based on irregularities in the vertical and horizontal histogram [6]. The second method was created as a confirmation of obstacles detected by the first method to increase its accuracy.

Canny edge detection algorithm is utilised well in the methods discussed. Although, there are some known weaknesses in the solution presented. As the methods presented required the detection of sidewalk borders to determine its detection path, wrong detection of walking borders would significantly affect the obstacle detection accuracy. Finally, the model would not work if there is no sidewalk border in the user's path such as indoors.

Next is to clarify the reason why the researcher chose Canny edge detection algorithm for the proposed model. The accuracy of the edge detection algorithm varies from one to another; some may need more processing power, and some may output nosier results. Factors like extra noise on the edge detection might affect the accuracy of obstacle detection [7]. Thus, a comparison of different edge detection algorithm needs to be reviewed to find an optimal edge detection algorithm which is suitable for obstacle detection and avoidance.

As stated by Li Bin [2], the greater the threshold is, the more precise the image edge processing effect and the edge points are significantly more coherent. However, passing the threshold of 0.3, the information of the image's edges is lost. Canny produced the best results compared to other algorithms. This is because Canny could filter noise and maintain information integrity. The canny algorithm is also not susceptible to noise interference, which enables it to detect weak edges [2].

The ability to use various threshold values is required to ensure higher accuracy. While the lighting condition may affect the result of obstacle detection, it can be controlled by adjusting the threshold values. Canny edge detection algorithm, which can retain information with a higher threshold, is essential.

III. PROPOSED METHODS

This paper proposed a real time obstacle detection model for mobile devices for the low vision community.

A. Model Overview

Using a mobile device as the main gadget would introduce the least mobility restrictions, in comparison to larger device such as a laptop. In addition, the model was designed only to use sensors commonly found in a mobile device which is cameras, gyroscopes and proximity sensors. This ensures the solution would not need any hardware production before distribution.

The model was designed after reviewing various other vision-based obstacle detection approaches, such as floor sampling using edge detection [14], depth map estimation [4]

Authorized licensed use limited to: UNIVERSITY OF ULSTER. Downloaded on September 30,2022 at 10:06:17 UTC from IEEE Xplore. Restrictions apply.

and obstacle detection within a sidewalk frame [6]. The model developed in this paper is shown in Figure 2. It was developed to provide obstacle detection in real time via edge-based algorithm (to obtain surrounding information) and different common sensors in a mobile device. The sensors used in this model are the camera (primary input), proximity sensor (lighting information), accelerometer (movement information), and gyroscope (camera angle information).

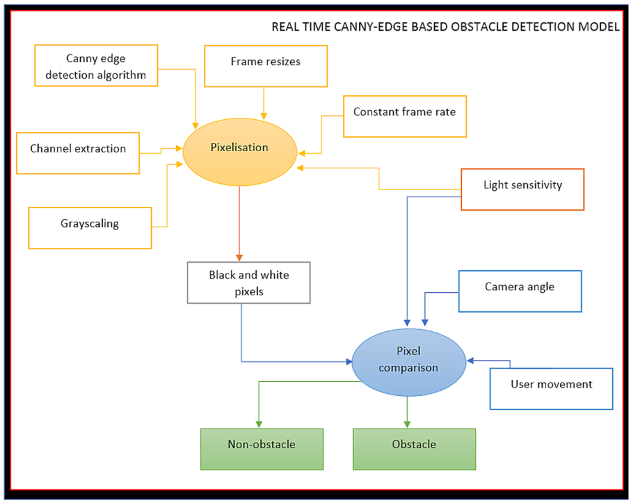


Fig. 2. Realtime Obstacle Detection Model Overview

As shown in Figure 2, the model requires edge information of the user's surrounding as the primary input to detect an obstacle. The edge detection algorithm chosen is the Canny Edge detection algorithm [5]. The ability of Canny edge detection algorithm to retain better details in different threshold levels, in comparison to other edge detection algorithms, is essential to obtain stable input data when the detection sensitivity is manipulated accordingly [14]. Additional information from secondary sensors such, as the proximity sensor for lighting information, is used to apply real-time tuning to the edge detection sensitivity.

Figure 3 shows the flow of the model's algorithm. First, the model requires real-time camera input. Different camera models may have different specifications. To solve this problem, the input was optimised. Constant values for the input variables were set. The optimisation process in this model starts with setting the various input resolutions to a constant resolution of 640 x 480 pixels. As reviewed, this resolution is supported in all the low end mobile devices. This ensures that the model can be used in different devices while keeping the processing power low.

Next, the framerate requested from the input was reduced to 30FPS (frames per second). Setting the framerate with this constant would also trigger the rest of the model to operate in 30 loops per second. To further reduce the processing power required in the pixelisation process, the frame input was converted to grayscale.

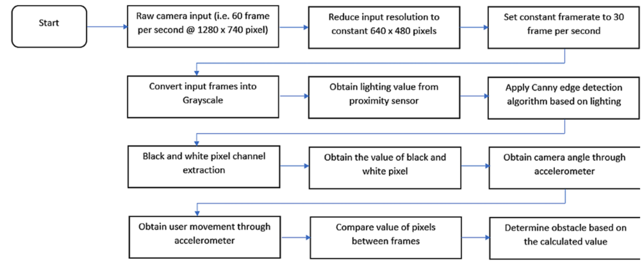


Fig. 3. Model Algorithm Flow

Before applying Canny edge detection algorithm, lighting information was obtained by using the proximity sensor. This provides essential surrounding lighting information which enables adaptability of the model to different lighting environments by adjusting the detection sensitivity. Depending on the implementation, this process could be repeated differently to obtain the best performance to power usage ratio. After the sensitivity threshold of the detection is obtained, the edge information is obtained by applying the said Canny edge algorithm. The edges information comes in grayscale which later converted into obtaining only black and white pixels matrix. The black and white pixels value were assigned to separate variables respectively. The model later calculates the total value for each channel. This is an essential step before the pixel calculation step.

The next sensor used before the pixel calculation is the accelerometer. Firstly, the sensor is used to obtain the angle of the camera. The obstacle detection uses this angle to adapt to the environment in terms of frontal or floor-based detection. However, this paper focused on the floor-based obstacle detection. Next, the information obtained from the accelerometer would also be used to determine the user's movements. This information determines the (re-)calibration process of the model algorithms and the pixel calculation for obstacle detection.

B. *Pixels Calculation*

This section explains the last stage of this model, which is the pixels calculation. It is done through the calculation of pixel differences. The pixel calculation algorithm introduced in this paper would be the method of comparing extra white pixels in the path as an obstacle. It would compare the differences between the current frame and the previous frame for extra white pixels. After including other information from the secondary sensors, the pixel comparison calculation would confirm the presence of an obstacle in the scene in real time.

Shown in Figure 4(a) is an example of the path without any obstacle. After applying the model, the result of the view is shown in Figure 4(b). Without any obstacle, the value of the total white pixels, for example, is approximately 6000. As the model was designed to use a constant resolution for all mobile devices, the value would be the same. Next, Figure 4(c) is the path with a basket to represent an obstacle. This would be an obvious obstacle to the pixel comparison calculation. Figure 4 (c) approximately consists of an additional 8000 white pixels in comparison to the previous path which does not have any obstacle. However, rather than immediately detecting it as an obstacle, the model would verify it first by using different information provided by the secondary sensors. It would still

Authorized licensed use limited to: UNIVERSITY OF ULSTER. Downloaded on September 30,2022 at 10:06:17 UTC from IEEE Xplore. Restrictions apply.

consider the information provided by proximity and accelerometer to confirm the obstacle detected. For this implementation, the alert would only be sent out when the user is on the move and when there is no difference in the lighting condition. This method reduces the amount of unnecessary sound alert while decreasing false detection. When an obstacle is confirmed, an alerting sound would be played to the user to warn them about the obstacle in their path.

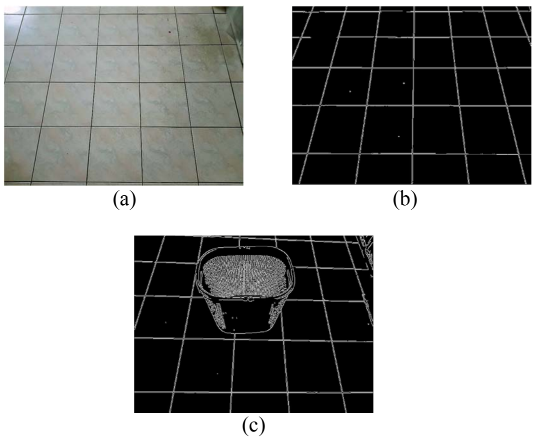


Fig. 4. Input and Output Example of the Model. (a) Raw Input of the Camera. (b) No Obstacle Example (c) Obstacle Example

Pseudocodes in Figure 5 shows the general method of how to detect the floor-based obstacle detection. However, the accuracy of the calculation and the obstacle detection sensitivity would depend on the value used in the calculation.

```
If optimised environment is met for this type of pixel comparison as confirmed by the sensors, utilise this pixe
comparison.

Var PreviousFrame
Var Current Frame
If CurrentFrame.Whitepixels is more than PreviousFrame.Whitepixels
Then check user movement, lighting condition and camera angle information
If drastic changes in that said information, no obstacle is confirmed.
If the obstacle already confirmed in the last half of a second, and the user has not moved, no obstacle is confirmed.
Else if the optimised value is obtained from those sensors as well, the obstacle is confirmed.
Recalibrate this algorithm when needed. This will run according to the framerate set, which is 30 frame per second.
```

Fig. 5. Obstacle Detection Pseudocode

IV. Results and Discussion

The model was implemented into an Android 7.0-based smartphone, Samsung Galaxy S6 Edge. Although this smartphone model may have been a high-end smartphone in 2015, its performance is now considered mid-tier in 2019. This is to strike a balance between testing the performance of the model while keeping the gadget cost low. Similarly performing smartphones could be obtained in the current market at an estimated price of RM 600. However, the model was designed to perform similarly even with lower performing devices, as long as the sensors required are presented.

A. Lighting Test

To ensure proper lighting values were used to adjust the sensitivity of the obstacle detection, a test was conducted. The value was obtained by using the smartphone's proximity sensor to adjust the Canny edge threshold accordingly for different scenario.

TABLE 2
Indoor Lighting Test Results and Verdict

Situation	Lighting Value (Lux)	Verdict
Dim hallway	10-40	Decrease Canny edge detection threshold which will increase its sensitivity to accommodate for the lack of lighting. Suggested 150-200 threshold.
General walking area	40-100	The main focus of the indoor obstacle detection as the majority of the walking path is in this condition. Suggested 200-250 threshold.
Dim working area	100-200	As the scenario is getting brighter, lower sensitivity is required. Suggested threshold 300-350.
Well-lit working area	200-450	In a well-lit area, the suggested threshold would be 450-470. This may change according to the initial pixel detected. However, brighter lighting condition is not the focus of this paper.

Table 2 shows the result of the lighting testing and the conclusion obtained from the test. Other than introducing different stages of lighting sensitivity for the algorithm to adapt to its surrounding, lighting information is also used to verify the obstacle detected. This is done by calculating the lighting condition differences between the current frame in comparison to the previous frame. The detected pixels are affected by the lighting condition. If the differences of lighting are above a certain threshold, the extra pixels would not be verified as an obstacle. Recalibration would also be done base on the lighting differences.

B. Orientation and Angle Testing

To ensure the obstacle detection accuracy would not be affected by the orientation of the mobile device held by the user, a camera orientation test was conducted. The mobile device was mounted on a tripod to ensure the angle is stabilised. The height of the camera to the ground was recorded and fixed to 110cm. It was also tested with the same area to ensure the consistency.

TABLE 3
Camera Orientation Test

Orientation / Angle	Lighting	Pixel Value	Height
Upright Landscape /-90°	8 lux	4945 px	110 cm
Opposite Landscape /89°	14 lux	4675 px	110 cm
Upright Portrait /2°	11 lux	4683 px	110 cm
Opposite Portrait /172°	4 lux	5211 px	110 cm

As shown in Table 3, there were no significant differences when using the proposed model in different orientations. The differences in pixels detected were mainly caused by the lighting differences in each orientation. Thus, the orientation of the mobile device would not be the focus in testing the model.

Next, the device angle test was conducted with the same settings. It is required as the time limitations introduced in the research does not allow the model to be optimised in every single angle it operates in. This testing provides the researcher

Authorized licensed use limited to: UNIVERSITY OF ULSTER. Downloaded on September 30,2022 at 10:06:17 UTC from IEEE Xplore. Restrictions apply.

with ranges of angles which would be optimised to increase the obstacle detection accuracy of the model. The objective of this test was to obtain an angle range which would be suggested for best accuracy. The result of the test is presented in Table 4 below.

TABLE 4
CAMERA ANGLE TEST

Angle	Lighting	Pixel Value	Height	Vertical Detection Range
-10°	4 lux	16363 px	110 cm	105 cm
-26°	4 lux	13036 px	110 cm	120 cm
-34°	5 lux	11446 px	110 cm	165 cm
-45°	7 lux	9465 px	110 cm	260 cm
-66°	6 lux	7744 px	110 cm	Undefined

After conducting the test, it can be concluded that the suggested angle range for floor-based obstacle detection to be +-30° to +-45°. This is because lower than +-30°, user's legs would interfere with the camera point of view while more than +-50°, the majority of the camera view would not capture the floor area anymore thus the vertical detection range becomes undefined.

C. *Obstacle Detection Test*

To test the model accuracy in detecting obstacles, few testing scenarios were created. Chest harness was used to equip the device to the user's chest with height approximated at 120cm. The device angle was set at 35° as suggested by the previous test. The user was blindfolded through each test and was supervised by a third-party helper who records the test result while ensuring no accident would happen in case of failure. In all the testing scenario, the user was required to return to the starting point which effectively doubles the obstacle to be detected in one testing round.

The first scenario is a bright indoor environment with fixed obstacles. Theoretically, this would be the best-case scenario for the model. The objective of this test was to discover the highest average accuracy obtainable by the model while still be able to test the point of difficulties faced upon detecting obstacles. Figure 6 (a) shows the scenario for this test in real-life. The scenario was populated with four boxes with different shape, and colour resulted in a total of eight expected obstacles. One of the challenges for the model in this scenario was the differences in lighting condition from 80 lux to 400 lux as the lighting was affected by the fluorescent lights shown in Figure 6 (a). The test resulted in an overall detection rate of 92% from 3 round of testing as shown in Table 5. The false negative of this test was mainly due to the white coloured small box shown in Figure 6 (a) at a certain angle of user's movement, the colour blends in with the user's path which made it unable to detect the obstacle in current sensitivity level.

The second scenario was similar to the first scenario but with dimmed lighting. This was done to measure the accuracy of the adaptability of changing its sensitivity in dimmed lighting. As shown in Figure 6 (b), the only light allowed in the scenario was the natural light from the nearest windows. The measured lighting value range for this scenario was 0 lux to 10 lux. The overall accuracy of this scenario was 87.5% as shown in Table 5. Similar to the previous test, the only false negative for the test was the small white coloured boxes which may blend with the path pixels in some moving angles.

To ensure the model would work to detect the differently shaped objects, the third scenario obstacles consists of four organic shaped obstacles as shown in Figure 6 (c). The testing environment was similar to the first scenario which means the challenges of lighting condition differences caused by the fluorescent was still applied. In this scenario, the overall accuracy obtained from the test was 79.1% as shown in Table 5. The main problem was detecting the blue clothes as shown in Figure 6 (c) as at a certain angle of movement. The obstacle was too small for the current sensitivity set for the model to verify as an obstacle.

Similar to the second scenario, the fourth scenario was to test the model capability to adapt with dim lighting condition but with organically shaped obstacles. Shown in Figure 6 (d) are the scenario obstacles and lighting condition. The result of this testing scenario was 87.5%. In contrast to the previous test, the color differences of the blue clothes in comparison to the dark path helps with the detection. However, the black bag introduced a problem where the edges information could not be obtained in a certain angle as it was too dark for the current edge detection sensitivity threshold.

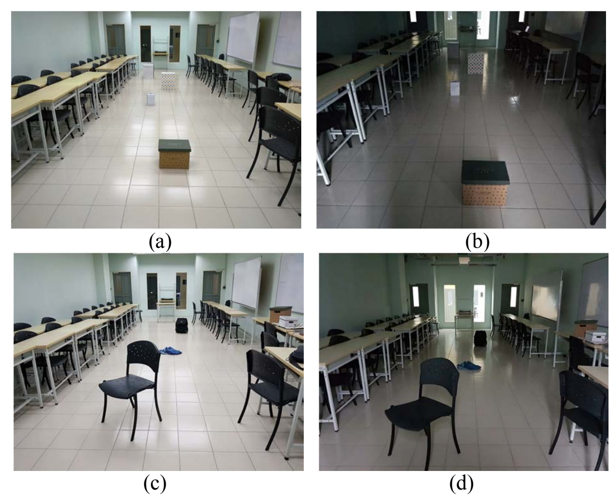


Fig. 6. Indoor Fixed Obstacles Testing Scenario. (a) Boxes Obstacles Bright Scenario, (b) Boxes Obstacles Dim Scenario. (c) Organic Obstacles Bright Scenario. (d) Organic Obstacles Dim Scenario.

The last testing scenario introduced for this paper was a fixed path with real-life obstacles. In this scenario, there were no pre-defined obstacles. The user would have to walk in a fixed path as shown in Figure 7 (a) then later turn to the right path which is shown in Figure 7 (b) before returning to the starting point. The objective of this test was to stimulate the real-life usage of the model. Every obstacle detected, and false negatives were recorded. The overall accuracy result of the test was 84.6%. Most of the obstacles were detected with no difficulties. However, most of the detection errors were due to the inability to detect plain walls as an obstacle. The plain wall would appear to have no extra white pixels in comparison to the previous frame. Other than that, light reflection from the path caused by

Authorized licensed use limited to: UNIVERSITY OF ULSTER. Downloaded on September 30,2022 at 10:06:17 UTC from IEEE Xplore. Restrictions apply.

the type of light used in the scenario would also cause a false positive which may confuse the user.

(a) (b)

Fig. 7. Indoor Fixed Path Real-Life Obstacles (a) Straight Path (b) Path Continuation After a Corner

TABLE 5
OBSTACLE DETECTION TEST RESULTS SUMMARY

Testing Scenario	Test 1	Test 2	Test 3	Average Accuracy
Bright indoor fixed obstacles	100%	87.5%	88%	92.1%
Dim indoor fixed obstacles	75%	100%	75%	83.3%
Bright indoor organic obstacles	75%	75%	88.8%	79.6%
Dim indoor organic obstacles	87.5%	75%	100%	87.5%
Indoor fixed path real-life obstacles	88.8%	71.4%	93.7%	84.6%
Overall Model Accuracy	85.42%			

After conducting the different testing scenarios, the overall detection accuracy is 85.42% as shown in Table 5. The test introduced some limitations. For instance, the model would face difficulties when detecting dark or small sized objects. Other than that, light reflections may be detected as an obstacle. In future works, an additional algorithm may be introduced to enable the model to detect a plain wall as an obstacle. Increasing the value of detection sensitivity may also help with detecting smaller sized obstacles.

V. CONCLUSION

The edge-based obstacle detection model has potentials in assisting the visually impaired. Implementing this model into mobile devices would enable one to have a supplementary assistive device in addition to the already available white canes. As the model only utilises sensors commonly available in a mobile device, the supplementary gadget could be available to the community without having them to purchase additional sensors or hardware, as long as the user owns a suitable mobile device. However, there are further improvement of the model such as adding algorithm to detect plain walls and automatic adjustment of sensitivity to detect smaller obstacles.

REFERENCES

[1] M. Al-Shehabi, M. Mir, A. Ali and A. Ali, "An Obstacle Detection and Guidance System for Mobility of Visually Impaired in Unfamiliar Indoor Environments", International Journal of Computer and Electrical Engineering, vol. 6, no. 4, pp. 337-341, 2014.

[2] L. Bin, "Comparison for Image Edge Detection Algorithms", IOSR Journal of Computer Engineering, vol. 2, no. 6, pp. 01-04, 2012.

[3] R. Bourne et al., "Magnitude, temporal trends, and projections of the global prevalence of blindness and distance and near vision impairment: a systematic review and meta-analysis", The Lancet Global Health, vol. 5, no. 9, pp. e888-e897, 2017.

[4] A. Caldini, M. Fanfani and C. Colombo, "Smartphone-Based Obstacle Detection for the Visually Impaired", Image Analysis and Processing — ICIAP 2015, pp. 480-488, 2015.

[5] J. Canny, "A Computational Approach to Edge Detection", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. -8, no. 6, pp. 679-698, 1986.

[6] D. Castells, J. Rodrigues and J. du Buf, "Obstacle Detection And Avoidance On Sidewalks", Proceedings of the International Conference on Computer Vision Theory and Applications, 2010.

[7] B. Crnokic, S. Rezic and S. Pehar, "Comparision of Edge Detection Methods for Obstacles Detection in a Mobile Robot Environment", Proceedings of the 27th International DAAAM Symposium 2016, pp. 0235-0244, 2016.

[8] V. Publications, "Vision Disability: Types, News and Information", Disabled World, 2019.

[9] N. Kanwal, E. Bostanci, K. Currie and A. Clark, "A Navigation System for the Visually Impaired: A Fusion of Vision and Depth Sensor", Applied Bionics and Biomechanics, vol. 2015, pp. 1-16, 2015.

[10] B. Lin, C. Lee and P. Chiang, "Simple Smartphone-Based Guiding System for Visually Impaired People", Sensors, vol. 17, no. 6, p. 1371, 2017.

[11] Malaysian Communications and Multimedia Commission, "Internet Users Survey 2016 - Statistical Brief Number Twenty", Mcmc.gov.my, 2016.

[12] G. Prasanthi and P. Tejaswitha, "Sensor assisted stick for the blind people", Transactions on Engineering and Sciences, vol.3, no.1, pp.12-16, 2015.

[13] A. Saxena, S. Chung, and A. Ng, "Learning Depth from Single Monocular Images", Stanford: Computer Science Department Stanford University, 2015.

[14] Z. Taha, and J. Jizat, "A Comparison of Two Approaches for Collision Avoidance of an Automated Guided Vehicle Using Monocular Vision", Applied Mechanics and Materials, vol. 145, pp.547-551, 2011.

Authorized licensed use limited to: UNIVERSITY OF ULSTER. Downloaded on September 30,2022 at 10:06:17 UTC from IEEE Xplore. Restrictions apply.