INVys: Indoor Navigation System for Persons with Visual Impairment Using RGB-D Camera

Widyawan[1], Muhammad Risqi Saputra[2], Paulus Insap Santosa[3]

[1,2,3] Department of Information and Electrical Engineering, Faculty of Engineering Universitas Gadjah Mada, Jl. Grafika No. 2 Kampus UGM, Yogyakarta 55281 INDONESIA (tel.: 0274- 552305; fax: 0274-552305; email: [1]widyawan@ugm.ac.id, [3]insap@ugm.ac.id)
[2] Monash University, Indonesia, Green Office 9 Building, Jl. BSD Green Office Park, Sampora, Cisauk, Tangerang Regency, Banten 15345 INDONESIA (fax: 0274-552305; email: risqi.saputra@monash.edu)

[Received: 20 February 2023, Revised: 4 October 2023]
Corresponding Author: Widyawan

ABSTRACT — This research presents the INVys system aiming to solve the problem of indoor navigation for persons with visual impairment by leveraging the capabilities of an RGB-D camera. The system utilizes the depth information provided by the camera for micronavigation, which involves sensing and avoiding obstacles in the immediate environment. The INVys system proposes a novel auto-adaptive double thresholding (AADT) method to detect obstacles, calculate their distance, and provide feedback to the user to avoid them. AADT has been evaluated and compared to baseline and auto-adaptive thresholding (AAT) methods using four criteria: accuracy, precision, robustness, and execution time. The results indicate that AADT excels in accuracy, precision, and robustness, making it a suitable method for obstacle detection and avoidance in the context of indoor navigation for persons with visual impairment. In addition to micronavigation, the INVys system utilizes the color information provided by the camera for macro-navigation, which involves recognizing and following navigational markers called optical glyphs. The system uses an automatic glyph binarization method to recognize the glyphs and evaluates them using two criteria: accuracy and execution time. The results indicate that the proposed method is accurate and efficient in recognizing the optical glyphs, making it suitable for use as a navigational marker in indoor environments. Furthermore, the study also provides a correlation between the size of the glyphs, the distance of the recognized glyphs, the tilt condition of the recognized glyphs, and the accuracy of glyph recognition. These correlations define the minimum glyph size that can be practically used for indoor navigation for persons with visual impairment. Overall, this research presents a promising solution for indoor navigation for persons with visual impairment by leveraging the capabilities of an RGB-D camera and proposing novel methods for obstacle detection and avoidance and for recognizing navigational markers.

KEYWORDS — Assistive Technology, Image Recognition, Object Detection, Wearable Computers.

I. INTRODUCTION

Indoor navigation is one of the most difficult activities for persons with visual impairment, particularly if they must do it independently. Most people, including persons with visual impairment, spend most of their time indoor, so safe navigation in this environment is a must. In order to navigate safely, persons with visual impairment need to handle two important aspects of wayfinding, namely micronavigation and macronavigation [1], [2]. Micronavigation deals with sensing of the immediate environment for obstacles and hazards, while macronavigation is about navigating to remote destinations beyond the immediate perceptible environment [3]. Moreover, to successfully navigate to desired destination inside a building, persons with visual impairment should be able to confirm whether the place in front of them is the right destination. Without the aid of others or assistive tools, they may encounter difficulties in performing those tasks.

In order to overcome this problem, conventionally, persons with visual impairment use white cane or guide dog. A white cane allows persons with visual impairment to detect obstacles at average distance of 1,500 mm [4], but its utility is only limited in the aspect of micronavigation. Guide dog could help in both aspects of wayfinding, but it needs to be trained and the cost is prohibitive for most people [5].

Many researchers have been developing assistive technology (AT) in the form of electronic travel aids (ETA). Some studies opt to fuse a cane with laser [6], [7] or ultrasonic technology [8], [9] to assist persons with visual impairment in avoiding obstacles. The information of the obstacle is conveyed to the user by sound, voice, or vibration. Judging from that ability, these tools are better than the conventional white cane because they can automatically detect obstacles and actively give feedback to the users. However, laser and ultrasonic provide poor information about the environment [10] and they only support micronavigation function.

In area of macronavigation, global positioning system (GPS) is widely used to guide persons with visual impairment in outdoor [11], [12]. However, GPS is unsuitable for indoor environments because the coverage of satellite signal is poor, which in turn significantly decreases its accuracy [13]. Many researchers utilize radio-frequency identification (RFID) tag [14], [15] that is used as markers of places in indoor environments. However, RFID is an infrastructure-dependent technology that needs to be preinstalled inside a building by technicians. The absence of the infrastructure renders the building is not disability friendly.

Even though most of the studies focus only on one aspect of wayfinding (either micronavigation or macronavigation), some researchers try to combine both functions [16], [17]. However, the simultaneous integration of many technologies (such as laser, ultrasonic, GPS, or RFID) could potentially make the system too complex, cumbersome, infrastructure-dependent, and expensive to build.

Recently, the advancement of technology leads to the development of low-cost RGB-D cameras such as Microsoft Kinect. Unlike conventional camera, RGB-D camera provides two images: depth image from the depth camera and color image from the RGB camera. This kind of camera is useful for many scenarios and applications, including indoor navigation for persons with visual impairment. The depth image can be

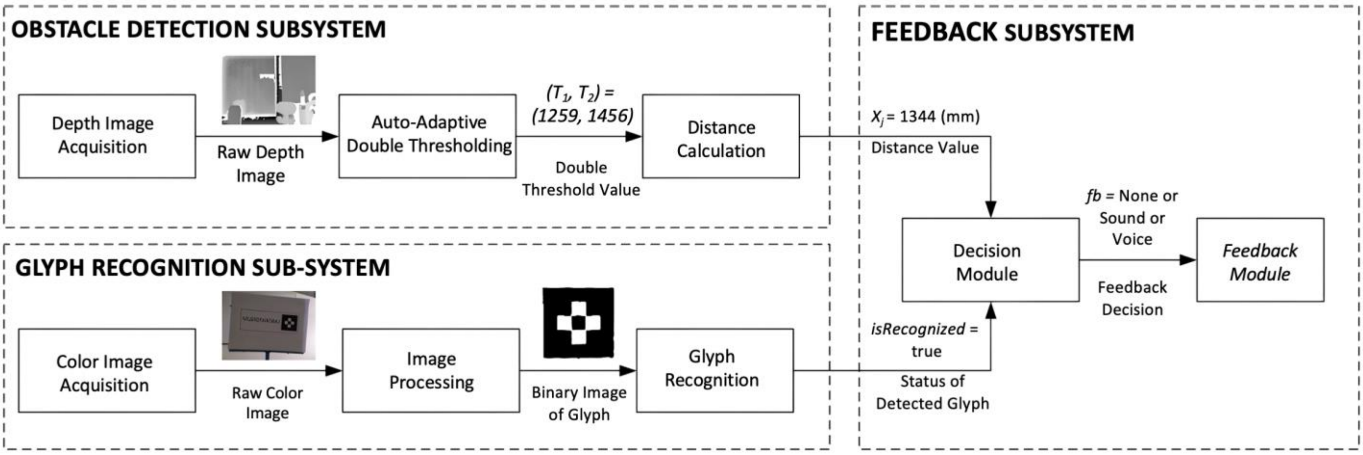


Figure 1. INVys architecture consists of three subsystems.

used to estimate the distance to the nearest physical surface by providing depth information for each pixel [18]. While the colour image can be utilized for macronavigation function by recognizing objects or printed markers collocated in interest points/places.

This research developed an indoor navigation system for persons with visual impairment, called INVys, based on an RGB-D camera. INVys employs a novel auto-adaptive double thresholding (AADT) method to detect obstacle and send feedback regarding the obstacle to the user. AADT splits the depth image area and searches the best two-threshold values automatically.

In area of macronavigation, INVys utilizes optical glyphs as printed marker to tag interest points. This study enhances Kirillov's glyph recognition method [19] using automatic glyph binarization.

The rest of this paper is organized as follows. Section II discusses micronavigation and its related studies. Section III gives overview of INVys's system architecture. Section IV explains obstacle avoidance function and Section V explains the algorithm to recognize glyph. Experimental setup and results are presented in the Section VI. Finally, Section VII sums up the conclusion.

II. MICRONAVIGATION

Micronavigation seems to elicit more interest and publications than macronavigation in the field of navigational aid for people with visual impairment. One of the early and prominent micronavigation technologies is laser, from laser cane [20] to virtual white cane [21] and many others that detect obstacles and hazardous objects [22], [23]. Triangulation method is generally used to calculate distance of the obstacle using laser measurement, by means of calculating the angle made by the diffuse reflected ray passing through the receiving lens [24]. Other researcher [25] combined infrared laser and Wiimote to locate user position in a room. However, this approach needs expensive preinstalled infrastructure. Moreover, the distance measurement using laser consumes a lot of energy [26].

Ultrasonic devices are mostly used for obstacle avoidance by combining it with a cane [6], [9], a wearable device [16], a mobile robot/system [17], or even a guide-dog robot [27]. Time of flight (ToF) is mostly used to calculate the distance of obstacles by measuring the time interval between sending the signal and receiving back the echo. However, both laser and ultrasonic technology provide poor information about the environment [28]. Moreover, by using those two technologies, the system needs to incorporate more than one sender/receiver of laser/ultrasonic sensors to map the complete scene in front of the user, which could make the system very complex, whereas simplicity and portability are important aspects in developing ETA for people with blindness [29].

Differ from the previous research, this study used depth camera that can directly perceive the complete view of map of the scene in front of the user with only one sensor without additional devices. Using depth camera, the ETA system could be much simpler and easier to use. However, in order to detect obstacles, raw depth data from depth image must be processed further. Some researchers have been developing methods to detect obstacle from depth image. The basic method is implemented by considering the height and width of persons with visual impairment to find the farthest point reachable on depth image [30]. Other approaches are applied by splitting depth image into many areas and processing each area separately to reduce the complexity of the scene [5].

Other methods combine the information from depth image and color image and process the data using image processing [31], [32] or computer vision [28], [33] algorithms to perform obstacle avoidance. INVys applies AADT method by combining the principle of dividing depth image into several areas based on the need of easy-to-learn/understand direction recommendation for persons with visual impairment and the depth-histogram based analysis that results in fast-effective-robust obstacle avoidance.

RFID tag is frequently used as marker of specific room or direction to guide persons with visual impairment to move from one room to another room [15]. The utilization of this RFID technology is considered effective for guiding persons with visual impairment; however, RFID is infrastructure-dependent technology, so numerous RFID tags should be installed within a building. Instead of using RFID technology, INVys leverages color camera to recognize printed marker.

III. SYSTEM ARCHITECTURE

INVys relies on reliable software processing of data from its RGB-D camera to perform micronavigation and macronavigation functions. The system is divided into three subsystems as shown in Figure 1. The first subsystem, obstacle detection, uses the AADT module to detect the closest obstacle and calculate its distance in millimeters. The second subsystem,

ω_1

glyph recognition, processes color images to find and classify glyphs. The third subsystem, feedback, uses the decision module to determine if feedback needs to be given to the user based on data from the first two subsystems. If feedback is required, the feedback module transmits the information in the form of sound or voice.

IV. OBSTACLE DETECTION

A. BASIC REQUIREMENT

The "friendliness" is one of the aspects that define the maturity of an obstacle detection and avoidance system [29]. In order to meet this requirement, this study divided region in front of persons with visual impairment into three, each representing path that they can traverse: the middle area (straight front), left area, and right area. By splitting into three regions, persons with visual impairment only needs to remember three directions, making it easier for them follow the instruction from the system.

Figure 2 illustrates implementation of the divided area. By this method, each subarea of the depth image can fully cover any big objects, such as a human body. For instance, refers to Figure 2, at distance of 1,500 mm from Kinect, each subarea of the depth image can cover object with maximum width of 549 mm. It is still wider than the average shoulder breadth of Singaporean and Indonesian [34].

B. BASELINE APPROACH

The depth image contains depth information in millimeters for each pixel, so basically it can be used for obstacle detection. As the depth image had resolution of 640 × 480 and the sensor registers of 30 fps, the total number of depth samples analyzed was more than 9 million. The problem is how to make the overload data become meaningful for persons with visual impairment.

The basic approach is developed by splitting depth image into many areas and calculating the average distance value for each area [5]. To improve efficiency, depth image of 640 × 480 was divided into 32 × 40 blocks and the average of pixel values was calculated. Subsequently, all the blocks were grouped into 5 × 3 regions, the ten middle values within each region were selected after sorting process, and the last average value of the obstacle distance was calculated based on those values. This approach yielded 15 distance values (one for each region) that were used as a metric of obstacle.

C. AUTO-ADAPTIVE THRESHOLDING (AAT)

Auto-adaptive thresholding (AAT) is the initial version of the algorithm to detect obstacle, AADT method. AAT refers to a method that can automatically generate threshold values that are specific to subarea of depth image. The threshold is used to separate the closest object and another object behind it. AAT starts by dividing depth image into three areas, converting each area into depth histogram, and finding the two closest peaks on that histogram [35]. The threshold value for each area of depth image was determined automatically by applying Otsu thresholding method [36] to the area between those two closest peaks on depth histogram. The Otsu thresholding method produces the best threshold value by maximizing the separability between two classes which are yielded by the threshold value. Thus, the data between the two closest peaks were divided into two classes, *C₀* and *C₁*, by maximizing the between-class variance using (1) as follows:

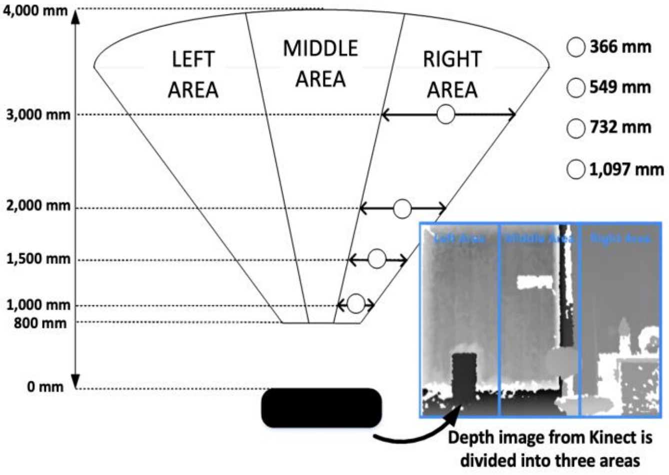


Figure 2. Captured image in front of persons with visual impairment is divided into three regions.

$$\sigma^2(k^*) = \ arg\,max(\sigma^2(k)), 1 \leq k \leq L \qquad (1)$$

where $\sigma^2(k^*)$ is the threshold value of k^* that maximize the between-class variance, $\sigma^2(k)$ is the between-class variance for each threshold value of k, and k is every possible threshold value that exists within range 1 and the maximum value of L in the image. The between-class variance itself is computed using (2).

$$\sigma^2 = \omega_0(\mu_0 - \mu_T)^2 + \omega_1(\mu_1 - \mu_T)^2 \qquad (2)$$

where σ^2 is the between-class variance, ω_0 is the probability of class C_0, ω_1 is the probability of class C_1, μ_0 is the mean of class C_0, $\mu_1$is the mean of class C_1, and μ_T is the overall mean of the whole image.

Based on the previous experiment, AAT is effective to detect and to calculate the distance of the obstacle when its distance is closer than 2,500 mm [35]. Afterwards, the accuracy begins to decline. It happens because AAT is unable to distinguish between the object and the floor in front of it when the floor is detected at a distance of farther than 2,500 mm.

D. AUTO-ADAPTIVE DOUBLE THRESHOLDING (AADT)

The AADT is a further development of AAT. It is developed to solve the primary problem of AAT, which is distinguishing the object from the floor in front of it when the object is at a distance of greater than 2,500 mm. Unlike AAT, which only employs a single threshold value, AADT produces two threshold values to distinguish between the obstacle and the floor in front of it. AADT generates a double threshold value automatically that varies for each area of the depth image (adaptive). The main differences between AADT and AAT are how to select peaks on the histogram, how to generate thresholds, and how to calculate the final distance value. The complete processes of the AADT method are described as follows.

1) DEPTH IMAGE ACQUISITION

INVys acquires raw depth data from Kinect and converts them into 8-bit grayscale images to visualize them on the screen using (3):

$$i_n = 255 - \left(\frac{255-(max(d_n-800.0))}{3{,}200}\right) \qquad (3)$$

where i_n is the *n*th pixel in grayscale image and d_n is the *n*th depth information in depth image.

2) DIVIDING DEPTH IMAGE INTO THREE AREAS

Based on the basic requirement explained in the previous section, depth image was divided into three areas: left, middle, and right. The occurrence frequency of this step and thereafter was 2 Hz. All next steps were processed separately and simultaneously for each subarea.

3) DOWNSAMPLING

In order to improve efficiency and to accelerate computing process, downsampling was conducted by using only one pixel value for each 2 × 2 block of pixels. Sampling for each 2 × 2 pixels is considered sufficient to reduce processed data while still maintaining enough information of the images.

4) DEPTH HISTOGRAM CONVERSION

The downsampled data from the previous step were converted into depth histogram by categorizing each pixel into 100 groups. The image is represented as distribution of pixel with interval of 40 mm between each group (as the depth camera coverage is 0 to 4,000 mm and it is divided into 100 groups.

5) PEAK DETECTION AND SELECTION

When depth information is converted into depth histogram, it will be full of peaks and valleys that represent local maximum and local minimum. Usually, a local maximum in depth histogram indicates an object [35]. Therefore, the closest object is determined by finding the closest local maximum in depth histogram. This process begins by calculating all of contrast values for each group in depth histogram using function in (4).

$$contrast(i,n) = \sum_{k=i-n}^{i+n} P_k - \sum_{k=i-2n}^{i-n-1} P_k - \sum_{k=i+n+1}^{i+2n} P_k (4)$$

where i is the observed position, n is the parameter for adding up the contrast of the peak and its neighbor, and p_k is the value in position of k in depth histogram. The number n is used to filter noises and unexpected local peak positions [37].

After all of the contrast values were found, the process continued by selecting the contrast value that might represent a local maximum. As a local maximum, the contrast value on the histogram has a positive value. The selection process was carried out by sorting contrast values from the highest to the lowest ones and choosing the group that fit the following criteria. First, the contrast value at the *i*th position should be larger than 300. If the value is below 300, the histogram at that position is not steep enough to be considered an independent local maximum. Second, the distance among the local maximum should be more than four groups of histograms; otherwise, it is regarded as part of another local maximum. The numbers were determined based on observation. After finding all the local maximum, the system selected the one closest to the Kinect (a peak/local maximum with the smallest number of groups).

6) DOUBLE THRESHOLD SELECTION

The next step was to find two threshold values (T_1, T_2) that would exclude the closest peak from any other data surround it. The first threshold value (T_1) is the first local minimum, and its distance value (x) is smaller than the distance value of the closest peak. Meanwhile, the second threshold value (T_2) is the same, but it has a bigger distance value (x) than the closest peak. Finding a local minimum from a local maximum required calculating the slope of each neighboring data point around the local maximum and halting when the slope changed extremely. The threshold is the distance value at the point when the gradient is changing dramatically. The detailed process to determine the value of the first and the second threshold is shown in Algorithm 1 and Algorithm 2.

Algorithm 1 Determining the First Threshold Value

```
i = p where y_p is the closest peak in depth histogram
x_i = the ith group of data
y_i = the frequency of depth information in ith group
T_1 = the first threshold value
d = distance between two groups
j ← 0
while i > 0 do
    slope ← (y_i − y_i -1)/( x_i − x_i -1)
    if slope ≤ 0 then
        if y_i -j < (y_i /2) then
            T_1 ← y_i -j × d
        end if
    end if
    j ← j + 1
    i ← i – 1
end while
```

Algorithm 2 Determining the Second Threshold Value

```
i = p where y_p is the closest peak in depth histogram
x_i = the ith group of data
y_i = the frequency of depth information in ith group
n = the total number of groups in depth histogram
T_2 = the second threshold value
d = distance between two groups
j ← 0
while i < n do
    if i = n-1 then
        T_2 ← i × d
    end if
    slope ← (y_{i+1} − y_i)/(x_{i+1} − x_i)
    if slope ≥ 0 then
        if y_{i+j} < (y_i/2) then
            T_2 ← y_{i+j} × d
        end if
    end if
    j ← j + 1
    i ← i + 1
end while
```

Based on those two algorithms, minimum requirements to determine the threshold have been added, namely the frequency of depth information on the threshold value should be less than half of the frequency of depth information on the observed peak ($<y_i/2$). It is done to avoid finding local minimum that is too close with the peak since local minimum can be a part of it.

7) FINAL DISTANCE CALCULATION

The final distance calculation was carried out by calculating average values of the depth information that existed between the first threshold (T_1) and the second threshold (T_2) using (5).

$$x_j = \sum_{k=1}^{n} \frac{i_k}{n}, T_1 < i_k < T_2 \qquad (5)$$

where x_j is average distance of the closest object at depth image area of j, i_k is depth value at pixel position of k at depth image area of j, T_1 is the first threshold value, T_2 is the second threshold value, and n is the total of i_k. The result of this process is three distance values of the closest object (in millimeter), one for each depth image area.

V. GLYPH RECOGNITION

Optical glyph is a printed marker represented with a square grid that its rows and columns are divided equally. Each cell is filled with either black or white color, except the first and the last row/column that is filled with only black color.

Optical glyph is generally used in augmented reality to generate 3D object exactly on the top of the recognized glyph. This technique was adapted to this research by recognizing the glyph to give information to the persons with visual impairment about navigational signs, places, or other interest points. The method to recognize glyph is based on the method developed by Andrew Kirillov [19] with a few modifications and enhancement in glyph binarization process. The method initiated with color image acquisition from RGB-D camera. The resolution and frame rate of the color camera are 640 × 480 and 30 fps, respectively. The next steps after the color image acquisition were divided into two major steps: image processing and glyph recognition steps. Those two steps were executed by image processing module and glyph recognition module.

A. IMAGE PROCESSING

The aim of this step is to process the acquired color image and to find the potential optical glyph to be recognized. The step is described as follows.

1) CONVERSION TO GRAYSCALE IMAGE

Using the ITU-BT.709 recommendation, the acquired RGB image was converted into a grayscale image in order to facilitate processing. This recommendation multiplies the red-green-blue value of the processing pixel with specific weight that is 0.2125 for red, 0.7154 for green, and 0.072 for blue.

2) EDGE DETECTION

The next step was to find all of edges on the grayscale image resulted from the previous step. The purpose of this step is to find all of edges that form rectangle shape so that the system can check whether this rectangle is a glyph or not. The method used to find object's edges was by finding the maximum difference value between neighbor pixels in four directions around the processing pixel. This is done using (6).

$$p' = max(|p_1 - p_5|, |p_2 - p_6|, |p_3 - p_7|, |p_4 - p_8|) \quad (6)$$

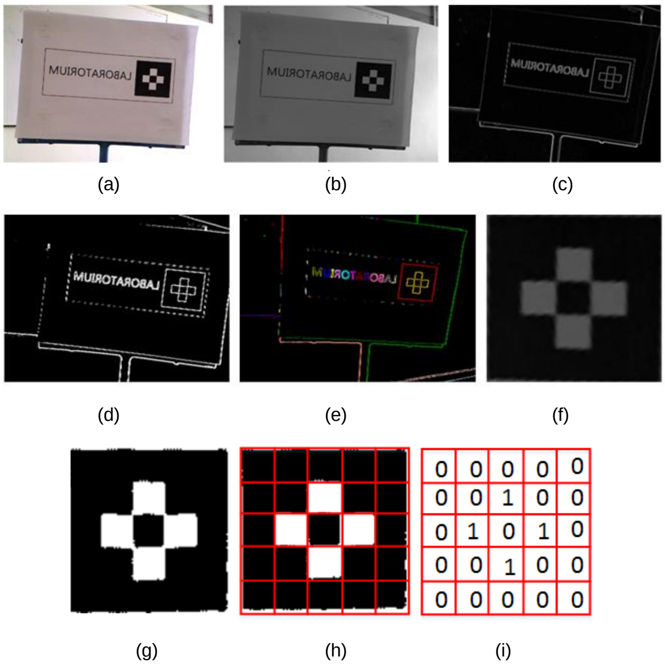
where *p'* is the final difference value, p_1 is the left-top neighbor pixel, p_2 is the middle-top neighbor pixel, p_3 is the right-top neighbor pixel, p_4 is the right-middle neighbor pixel, p_5 is the right-bottom neighbor pixel, p_6 is the middle-bottom neighbor pixel, p_7 is the left-bottom neighbor pixel, and p_8 is the left-middle neighbor pixel.

3) BLOB DETECTION

In order find a blob, the edges image was first converted into binary image using thresholding. Then, the blob was detected using a connected component labelling algorithm on top of the binary image. Connected component labelling scans all of pixels on the binary image and categorizes them into group based on the virtue of their connectivity [38].

4) RECTANGLE DETECTION

After the groups of connected pixels were found, each group was analyzed to find which one was a rectangle. The complete steps involved determining bounding box of the connected pixels, finding the center of the bounding box, detecting the first corner of connected pixels by finding the farthest point in connected pixels from the center of bounding box, detecting the second corner of connected pixels by finding the farthest point in connected pixels from first corner, detecting third and fourth corners by finding the two farthest points from a straight line formed by first and second corners, and checking whether all of lines connected by those four corners were within the boundary of distortion limit. This distortion limit was used to make sure that the processing connected pixels forms quadrilateral object. The distortion limit (*dm*) is calculated using (7).



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

0	0	0	0	0
0	0	1	0	0
0	1	0	1	0
0	0	1	0	0
0	0	0	0	0

Figure 3. Sample result of glyph recognition process, (a) acquired RGB image, (b) grayscale image, (c) edges image, (d) binary image, (e) blob image, (f) detected glyph image in grayscale, (g) detected glyph in binary image, (h) glyph binarization process, and (i) result of glyph binarization.

$$dm = max\left(m, \left(r \times \frac{w+h}{2}\right)\right) \quad (7)$$

where *m* is the minimum allowed distortion, *r* is relative distortion limit, *w* is width of the bounding box, and *h* is height of the bounding box. The result of *dm* will be compared to the average distance of the processing side of the rectangle with the center of the bounding box.

5) GLYPH DETECTION

Following the previous step, all detected rectangles were checked whether they were glyphs. This step was done by comparing average brightness between pixel area inside and outside the rectangle. Because the glyph border is black and it is always surrounded by white area, so the brightness difference is extremely high. The system then cropped only area of the image that contains the glyph.

6) CONVERSION TO BINARY IMAGE USING OTSU THRESHOLDING

The final process of the image processing step was converting the cropped image into a binary image by using Otsu thresholding. Thus, the result was an image with only two classes: the object class, which is usually indicated with white color, and the background class, which is indicated with black color. The sample result of all these processes can be seen in Figure 3.

B. RECOGNITION

The recognition steps were divided into two processes. The first one is called glyph binarization. Glyph binarization converts detected glyphs into binary values of 0 and 1. It is done by splitting the glyph into an equal number of rows and columns (*s*) and calculating the number of color pixels (white or black) from each cell. If a cell is dominated by the black pixels (beyond 60%), the cell is turned into 0. Otherwise, if it is full of white pixels (beyond 60%), then the cell is filled with 1. The result of glyph binarization is a matrix with only 0 and 1 values.

Kirillov's glyph binarization method has a drawback, that is the size of the glyphs must be set first. Hence, the user should create a glyph with a predefined size. This research solved this drawback by proposing automatic glyph binarization. Automatic glyph binarization detects the size of the glyph automatically based on the glyph size categories. The sizes of the glyphs generally used are from 5 × 5 to 10 × 10. A glyph size larger than 10 × 10 is rarely used because it is difficult to create without computer software. Automatic glyph binarization will check if the detected glyphs meet one of the glyph sizes, ranging from 5 × 5 to 10 × 10. The detection method is based on the characteristic of the glyph that the white data (value = 1) will always start from the second row/column.

The size detection process began with dividing the glyph into a 10 × 10 matrix (s = 10) and checking whether the white data (value = 1) existed (beyond 60%) in the second row. If it does, it means the size of the glyph is matched. Thus, the size of the glyph was determined as 10 × 10. If it does not exist, the process is continued by dividing the glyph into smaller sizes, that is, 9 × 9 (s = 9). The details of the glyph size detection process can be seen in Algorithm 3.

Algorithm 3 Glyph Size Detection

```
$I(w,h)$ = an $I$ glyph image, with $w$ width and $h$ height
$I_{m,n}$ = pixel at $m$th column and $n$th row in $I$ image
$s$ = size of glyph ($s \times s$)
$cw_s$ = width of each cell in $s \times s$ glyph
$ch_s$ = height of each cell in $s \times s$ glyph
$result$ = final detected glyph size
$s \leftarrow 10$
while $s \geq 5$ do
    $cw_s \leftarrow w / s$
    $ch_s \leftarrow h / s$
    for $i \leftarrow 1$ to $s - 2$ do
        $r \leftarrow 0$
        for $a \leftarrow 1$ to $ch_s$ do
            $n \leftarrow ch_s + a$
            for $b \leftarrow 1$ to $cw_s$ do
                $m \leftarrow (cw_s \times i) + b$
                if $I_{m,n}$ equals to white color then
                    $r \leftarrow r + 1$
                end if
            end for
        end for
        if $r > (0.6 \times cw_s \times ch_s)$ then
            $result \leftarrow s$
            break
        end if
    end for
    $s \leftarrow s - 1$
end while
```

The second step, called glyph matching, compared the matrix from the previous step with the data saved in the database. The glyph matching was conducted for four possible conditions: ideal condition (the glyph is not rotated), rotated 90°, rotated 180°, and rotated 270°. It was done by rotating the matrix for each condition, looping each value on the matrix, and comparing the value with the matrix resulting from saving the image in the database. The sample result of glyph binarization and glyph matching process can be seen in Figure 3(h) and Figure 3(i).

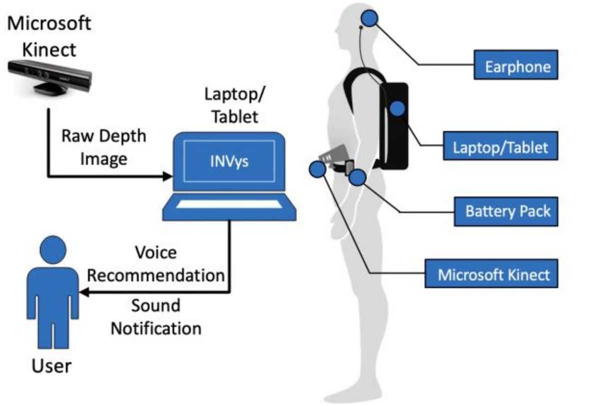
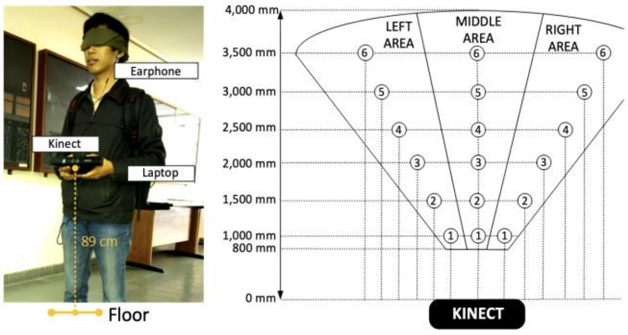


Figure 4. INVys consists of four main parts that will be used by persons with visual impairments as depicted in figure on the right side. The data flow between each part is shown in the left side figure.



(a)

(b)

Figure 5. Images of, (a) a blind-folded man conducting experiment, (b) the position of object in obstacle detection experiments. There are 18 positions in total, 6 positions for each depth image coverage.

VI. EXPERIMENTS

The experiments were conducted to evaluate two main functions of INVys system, namely obstacle avoidance as a micronavigation function and glyph recognition as a macronavigation function. The experiments setup and the results were grouped into two clusters. Hardware configuration, experimental setup, performance indicator, and result are described in following sections.

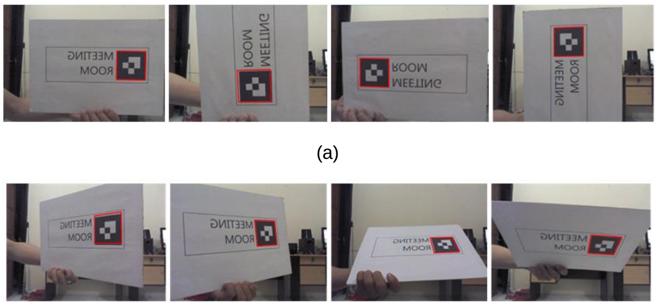
A. HARDWARE CONFIGURATION

As shown in Figure 4, INVys consists of four main hardware parts: Microsoft Kinect 360 as RGB-D camera, battery pack as the power supply for Kinect, laptop/computer tablet as the main processor that contains INVys software, and an earphone as a device to give the feedback to the user. Kinect provides 640 × 480 depth image resolution, 3,500 mm maximum range, 57.5° horizontal and 43.5° vertical angle of vision [39] that is considered suitable to develop vision-based indoor navigation.

B. SETUP AND PERFORMANCE INDICATOR

1) OBSTACLE DETECTION

The primary purpose of an obstacle detection experiment is to check whether the obstacle detection algorithm works well and to measure its distance calculation accuracy. The RGB-D camera was placed at a height of 890 mm from the floor (based on the average hip height of 50-year-old Indonesian people [34]), as seen in Figure 5(a). The experiment was conducted by placing six different objects commonly found in an indoor environment (human, chair type 1, chair type 2, trash, electric



(a)

(b)

Figure 6. Experiment of glyph recognition, (a) in normal condition and (b) in tilted condition.

fan, piles of cardboard) at 18 different positions (six positions for each depth image area) as seen in Figure 5(b) and comparing the distance of obstacle obtained by the system and the real distance in millimeter. Three methods were tested, namely the baseline approach, AAT, and AADT. Then, those three algorithms were compared using four performance indicators.

The first indicator is accuracy, which is the most important performance metric. It is indicated by the difference between the estimated location and the true location [40]. It is measured by calculating average distance of error measurement (μ_{obs}) using (8) where x is the real distance of obstacle, x' is the distance calculated by the algorithm, n is the number of experiments.

$$\mu_{obs} = \sum \frac{|x-x'|}{n}. \quad (8)$$

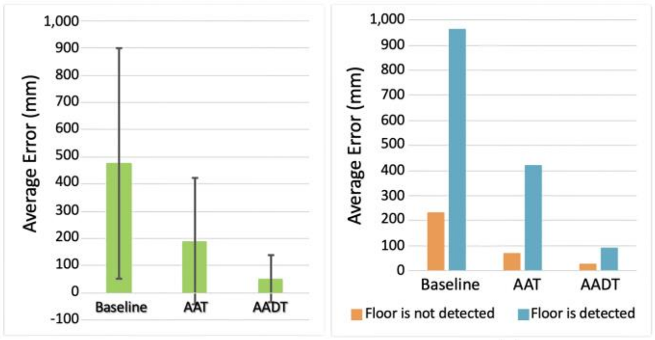
The second indicator is precision. If accuracy only measures the value of average distance errors, precision measures how consistently the system works. It measures the robustness of the algorithm as it reveals the variation in its performance over many trials. In area of positioning technique, cumulative distribution functions (CDF) are commonly used to measure precision [40]. This research used CDF as well to measure the precision of the compared algorithm.

The third indicator is the robustness of the accuracy along depth image coverage. When precision measures the robustness toward the number of trials, this third metric measures the robustness of the accuracy of the algorithm along depth image coverage. It considers how consistent the accuracy of distance calculation algorithm from the close distance of the obstacle until the distance of the obstacle reaches the coverage limit of RGB-D camera. It is important because the pixel accuracy of Kinect's depth image decreases when the distance between scene and sensor increases [41]. Thus, basically, the accuracy of distance calculation algorithm may decrease when distance of the observed object increases. From Kinect's point of view, it may happen because of the missing depth information in depth image. Whereas, a positioning technique with high robustness should function normally even when some signals are not available [40]. Therefore, this indicator is significant to evaluate the performance of the compared algorithm.

The last indicator is execution time. It is a metric to measure how fast the algorithm works in the real-time conditions. The algorithm was executed on notebook with Intel processor core i3 1.8 GHz and 4 MB of RAM.

2) GLYPH RECOGNITION

The glyph recognition experiment was conducted using five different patterns of glyphs and two different sizes of glyphs, namely 6 × 6 cm (glyph A) and 11.5 × 11.5 cm (glyph B). It was done to find out the best glyph size used in the context of persons with visual impairment so that the system could recognize the glyph at an acceptable distance from the persons with visual impairment. This experiment tried to recognize glyphs for each 300 mm range from the RGB-D camera until the maximum distance the system could not recognize the glyph anymore. For each 300 mm distance, the glyphs were tested in two different scenarios: normal and tilted conditions.



(a) (b)

Figure 7. Graphs of, (a) average error of distance calculation for each obstacle detection algorithm and (b) comparison of average distance errors for each algorithm when the floor is detected and not.

Normal condition means the glyph is recognized in the perpendicular direction toward the RGB-D camera. In this scenario, the glyph was tested in four different positions: ideal position (not rotated), rotated clockwise 90°, rotated clockwise 180°, and rotated clockwise 270°. The sample of this experiment can be seen in Figure 6(a).

In tilted conditions, the system attempts to recognize the glyph position that is not perpendicular to the RGB-D camera. This experiment was conducted because, in actual implementation, the RGB-D camera was moving along with persons with visual impairment, causing the glyph to not always be visible perpendicular to the RGB-D camera. In this scenario, the glyphs were tested in four different positions; namely, the glyph was tilted 45° to the left-back, right-back, top-back, and bottom-back. The sample of this experiment can be seen in Figure 6(b).

In order to evaluate the two types of glyphs mentioned before, two performance indicators were used, namely accuracy and execution time. Accuracy compares how many times the system can recognize the glyph with the number of trials for each 30 cm range. The value is measured in percentage. Meanwhile, execution time measures how fast the system execute the algorithm, starting from color image acquisition to glyph recognition.

C. RESULTS

1) OBSTACLE DETECTION

Figure 7(a) depicts each compared algorithm's average distance calculation errors. It can be seen that AADT has the smallest average distance errors compared to the others. The average error of AADT was 50.2 mm, with a standard deviation of 87.7 mm. Besides the comparison of average distance errors among those algorithms, the researchers were also interested in observing the difference in average distance errors when the floor was detected (at a distance farther than 2,500 mm) and not detected. The aim is to find out whether AADT could finally resolve the drawbacks of AAT. It can be seen in Figure 7(b)

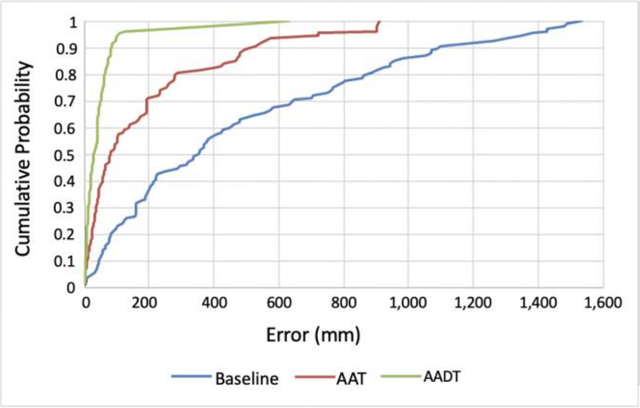


Figure 8. Cumulative distribution function (CDF) of distance calculation errors over many trials.

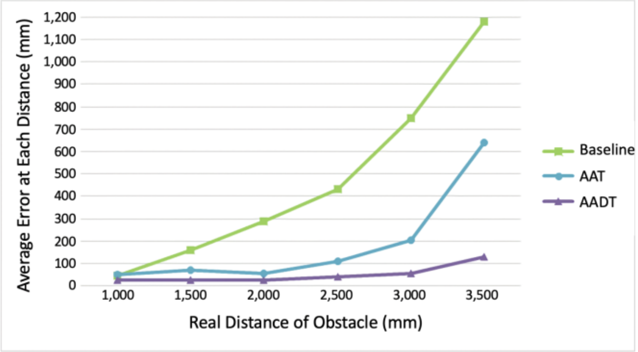


Figure 9. Robustness of the algorithm accuracy toward the increasing distance of the obstacle, along RGB-D camera coverage.

TABLE I
AVERAGE EXECUTION TIME OF BASELINE APPROACH, AAT, AND AADT

Algorithm	Average Execution Time (ms)	Standard Deviation
Baseline	7.41	1.19
AAT	7.05	1.22
AADT	9.50	3.61

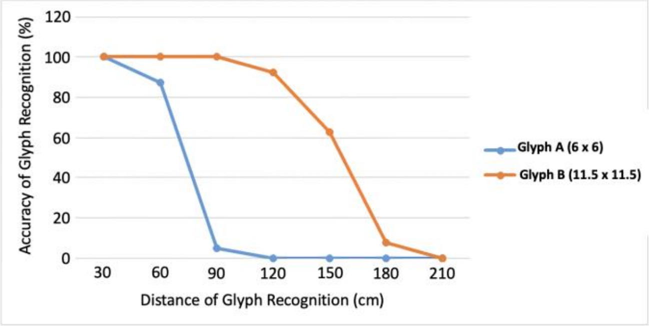


Figure 10. Comparison of average accuracy of glyph recognition between glyph A (6 x 6 cm) and glyph B (11.5 x 11.5 cm) for every distance in trials.

that when the floor was detected, all of the compared algorithms had worse distance calculations than when the floor was not detected. The baseline was taken from the previous paper [5]. It was calculated based on the average depth within each block. The baseline value will have large errors, especially when the values of nearby pixels are divergent.

AAT struggles when it cannot separate the object and the floor before it, especially when the floor is detected at a distance exceeding 2,500 mm. The average error of AADT was below 100 mm for both conditions, far better than the baseline approach and AAT.

In regard to precision, Figure 8 depicts the CDF of error calculation of baseline, AAT, and AADT algorithms. Based on Figure 7(a), the standard deviation of AADT was 87.7 mm, so the maximum value of error calculation considered normal was within the range of 0 to 137.9 mm.

However, it can be observed in Figure 8 that 93.5% of errors of AADT were below 100 mm. Thus, over 108 trials, most of them resulted in low error distance calculation. Even AAT only had a location precision of 53.7% in 100 mm (the CDF of distance error of 100 mm was 0.537). So, AADT is the most precise one.

The accuracy of the distance calculation algorithm often decreases when the distance of the observed object increases. It means that the value of errors in calculating nearby objects will be smaller than the farther one. Figure 9 depicts the robustness of the compared algorithm against this issue. Based on Figure 9, the accuracy of the baseline approach was still acceptable at a distance of 2,000 mm; beyond this distance, the average distance errors were too high. At a distance of 2,500 mm or less, AAT performed very well with average distance errors below 200 mm. However, if farther than that distance, the average distance errors became higher even reached more than 600 mm at a distance of 3,500 mm. While in AADT, from the closest distance to the coverage limit of RGB-D camera, the average distance errors were constantly below 150 mm. The range of AADT's average errors from the first to the last distance of measurement was only 103.3 mm. Therefore, the accuracy of AADT's distance calculation decreased by approximately 21.3 mm for every 500 mm of the increasing distance of the obstacle from the RGB-D camera.

The last performance indicator is execution time. Table I shows the comparison of execution time between the baseline approach, AAT, and AADT. The baseline approach and AAT had similar average execution times, approximately 7 ms, with a standard deviation of 1 to 1.5. While AADT had a slower average execution time of 9.5 ms, with a standard deviation of 3.61. It had only a slight difference compared to the other algorithms. Based on the described performance evaluation, it is concluded that AADT is better than the other algorithms regarding accuracy, precision, and robustness toward the increasing distance of obstacles. However, it is slower in execution time with only small differences.

2) GLYPH RECOGNITION

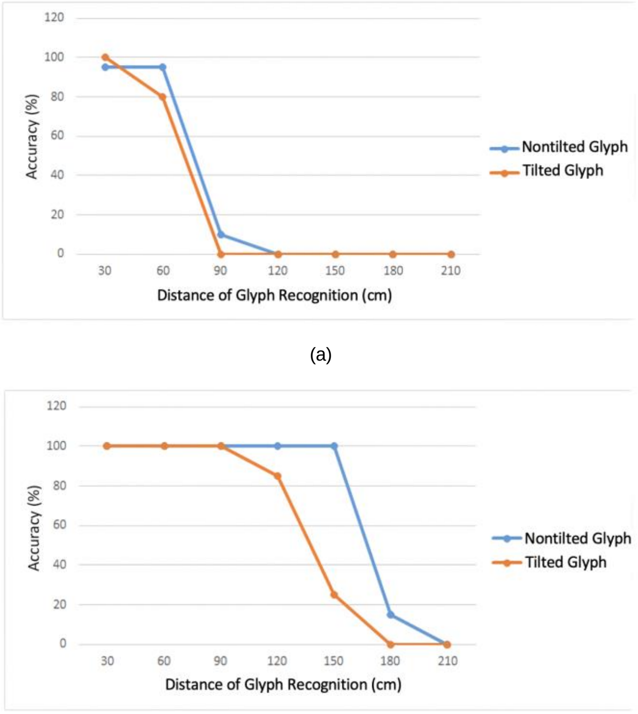
Figure 10 depicts a comparison of the average accuracy of glyph recognition between glyph A (6 × 6 cm) and glyph B (11.5 × 11.5 cm) for every distance in trials. Figure 10 shows that glyph B still had an average accuracy beyond 90% when the glyph distance from the RGB-D camera reached 120 mm. The average accuracy for glyph A at a distance of 120 mm was 0%, as no glyph could be detected. The accuracy of glyph A decreased significantly at a distance of 600 mm or beyond. In contrast, the accuracy of glyph B fell to unacceptable accuracy for persons with visual impairment at a distance of 150 mm or beyond.

Based on Figure 10, it can be concluded that:

- The accuracy of the glyph recognition is inversely proportional with the distance of glyph from the RGB-D camera. It means that the accuracy decreases when the distance increases.
- The size of the glyph is proportional with the accuracy of the glyph recognition. Thus, the larger the size of the

TABLE II
AVERAGE EXECUTION TIME OF GLYPH RECOGNITION BETWEEN GLYPH A AND GLYPH B

Information	**Glyph A (6 × 6 cm)**	**Glyph B (11.5 × 11.5 cm)**
Average execution time (ms)	82	68
The fastest execution time (ms)	53	41
The slowest execution time (ms)	138	90



(a)

(b)

Figure 11. Comparison of glyph recognition accuracy between tilted glyph and non-tilted glyph in (a) glyph type A (6 x 6 cm) and (b) glyph type B (11.5 x 11.5 cm).

glyph is, the better the accuracy of the glyph recognition for each trial distance.

- Glyph B is better used than glyph A because it has high accuracy at distances of 100 to 1,500 mm. Therefore, INVys implements sound notification at those distances. Thus, persons with visual impairment can recognize the glyph before colliding with it.
- Based on the third point, the average accuracy of glyph B from distance 300 to 1,500 mm, that is 91%. This outcome is far better than glyph A that could only achieve 38.5% average accuracy at same distance.
- With the result of the fourth point, the minimum glyph size that practically effective in the context of indoor navigation for persons with visual impairment, that is 11.5 × 11.5 cm or beyond.

Table II details the average recognition execution time for both glyphs A and B. The execution time of glyph B was faster than that of glyph A, despite the fact that both values were less than 100 ms. Therefore, glyph B is not only better recognized but also quicker. However, the difference is so minute as to be insignificant.

The glyph recognition analyzed the accuracy in both tilted and nontilted glyph types. Figure 11 depicts the comparison of glyph recognition accuracy between tilted glyphs and nontilted glyphs in both types of glyphs A and B. It indicates which condition that contributes more to decreasing the overall recognition accuracy. From Figure 11(a), a tilted glyph has lower recognition accuracy than nontilted glyph in most distances of glyph recognition trials. This result is readily apparent in Figure 11(b). At a distance between 1,200 mm and 1,800 mm from the RGB-D camera, the tilted glyph B had a lower accuracy than the non-tilted glyph B. Even the difference is quite significant. At a distance of 1500 mm, a nontilted glyph still had 100% accuracy, while a tilted glyph had only 30% accuracy. Tilted glyphs have a more significant impact on reducing overall glyph recognition accuracy than nontilted glyphs.

VII. CONCLUSION AND FUTURE WORKS

In this paper, INVys impairment using the RGB-D camera have been presented as an indoor navigation system for persons with visual impairment. INVys uses a depth camera to perform micronavigation function and utilizes color camera for macronavigation function. In micronavigation, INVys proposes AADT algorithm in detecting obstacle.

Experiments demonstrated that AADT outperformed other algorithms in terms of accuracy (50.2 mm distance errors), precision (95% errors in trials were below 100 mm), and robustness toward the increasing distance of obstacle (the average distance errors was constantly below 150 mm from the closer distance to the coverage limit of the RGB-D camera). The sole performance indicator that AADT performed worse on than other algorithms was execution time; however, the difference was so small (approximately 3 ms) so it can be ignored.

In regards of macronavigation function, INVys recognizes optical glyph as marker of interest points or navigational signs that is collocated in indoor environment using automatic glyph binarization method. Trials with two different sizes of glyph, namely glyph A (6 × 6 cm) and glyph B (11.5 × 11.5 cm), indicate that glyph B had an average accuracy of 91% from distance 300 to 1,500 mm, far better than glyph A that had an average accuracy of 38.5% at the same distance. For future development, it would be great to add a positioning system and path planning function that does not rely on any markers. It can be used to guide persons with visual impairment to walk step by step from one place to another place seamlessly.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest.

AUTHOR CONTRIBUTION

Conceptualization, Widyawan and Muhammad Risqi Saputra; methodology, Widyawan and Paulus Insap Santosa; software, Muhammad Risqi Saputra; validation, Widyawan; formal analysis, Widyawan and Paulus Insap Santosa; investigation, Widyawan and Muhammad Risqi Saputra; resources, Muhammad Risqi Saputra; data curation, Paulus Insap Santosa; writing—original draft, Widyawan and Muhammad Risqi Saputra; writing—review and editing, Paulus Insap Saputra; supervision, Widyawan and Paulus Insap Saputra.

REFERENCES

[1] Y. Wu, H.B. Zhu, Q.X. Du, and S.M. Tang, "A Survey of the Research Status of Pedestrian Dead Reckoning Systems Based on Inertial Sensors," *Int. J. Automat., Comput.*, Vol. 16, No. 1, pp. 65–83, Feb. 2019, doi: 10.1007/s11633-018-1150-y.

[2] H. Petrie, "User Requirements for a GPS-Based Travel Aid for Blind People," in *Proc. Conf. Orientat. Navig. Syst. Blind Persons*, 1995.

[3] W.C.S.S. Simões, *et al.*, "A Review of Technologies and Techniques for Indoor Navigation Systems for the Visually Impaired," *Sens.*, Vol. 20, No. 14, pp. 1–35, Jul. 2020, doi: 10.3390/s20143935.

[4] H. Takizawa, Y. Kuramochi, and M. Aoyagi, "Kinect Cane System: Recognition Aid of Available Seats for the Visually Impaired," *2019 IEEE 1st Glob. Conf. Life Sci., Technol. (LifeTech)*, 2019, pp. 189–193, doi: 10.1109/LifeTech.2019.8884061.

[5] A. Khan, F. Moideen, and J. Lopez, "KinDectect: Kinect Detecting Objects," in *Computers Helping People with Special Needs*, K. Miesenberger *et al.*, Eds., Heidelberg, Germany: Springer, 2012, pp. 588–595, doi: 10.1007/978-3-642-31534-3_86.

[6] B. Singh, "A Framework of Connected Smart Sensing Canes for Obstacle Detection and Avoidance," *2022 IEEE 10th Region 10 Humanit. Technol. Conf. (R10-HTC)*, 2022, pp. 76–80, doi: 10.1109/r10-htc54060.2022.9930047.

[7] M.A. Ikbal, F. Rahman, and M.H. Kabir, "Microcontroller Based Smart Walking Stick for Persons with Visual Impairment," *2018 4th Int. Conf. Elect. Eng., Inf., Commun. Technol. (iCEEiCT)*, 2018, pp. 255–259, doi: 10.1109/CEEICT.2018.8628048.

[8] H. Watanabe, M. Sumiya, and T. Terada, "Human-Machine Cooperative Echolocation Using Ultrasound," *IEEE Access*, Vol. 10, pp. 125264–125278, 2022, doi: 10.1109/ACCESS.2022.3224468.

[9] M.H.A. Wahab *et al.*, "Smart Cane: Assistive Cane for Visually-Impaired People," *Int. J. Comput. Sci*, Vol. 8, No. 4, pp. 21–27, Jul. 2011.

[10] Y. Zhao *et al.*, "Laser Based Navigation in Asymmetry and Complex Environment," *Symmetry*, Vol. 14, No. 2, pp. 1–18, Jan. 2022, doi: 10.3390/sym14020253.

[11] S. Real and A. Araujo, "Navigation Systems for the Blind and Visually Impaired: Past Work, Challenges, and Open Problems," *Sens.*, Vol. 19, No. 15, pp. 1–20, Aug. 2019, doi: 10.3390/s19153404.

[12] N.A. Kumar, Y.H. Thangal, and K.S. Beevi, "IoT Enabled Navigation System for Blind," *2019 IEEE R10 Humanit. Technol. Conf. (R10-HTC) (47129)*, 2019, pp. 186–189, doi: 10.1109/R10-HTC47129.2019.9042483.

[13] P.S. Farahsari, A. Farahzadi, J. Rezazadeh, and A. Bagheri, "A Survey on Indoor Positioning Systems for IoT-Based Applications," *IEEE Internet Things J.*, Vol. 9, No. 10, pp. 7680–7699, May 2022, doi: 10.1109/JIOT.2022.3149048.

[14] D.R. Bolla, S. Trisheela, P. Nagarathana, and H. Sarojadevi, "Object Detection in Computer Vision Using Machine Learning Algorithm for Persons with Visual Impairment," *2022 IEEE 2nd Mysore Sub Sect. Int. Conf. (MysuruCon)*, 2022, pp. 1–6, doi: 10.1109/MysuruCon55714.2022.9972494.

[15] R. Oktem and E. Aydin, "An RFID Based Indoor Tracking Method for Navigating Impaired People," *Turkish J. Elect. Eng., Comput. Sci.*, Vol. 18, No. 2, pp. 185–196, Mar. 2010, doi: 10.3906/elk-0904-3.

[16] S. Koley and R. Mishra, "Voice Operated Outdoor Navigation System for Visually Impaired Persons," *Int. J. Eng. Trends, Technol.*, Vol. 3, No. 2, pp. 153–157, Mar.–Apr. 2012.

[17] D. Ni *et al.*, "The Design and Implementation of a Walking Assistant System with Vibrotactile Indication and Voice Prompt for the Visually Impaired," *2013 IEEE Int. Conf. Robot., Biomimetics (ROBIO)*, 2013, pp. 2721–2726, doi: doi: 10.1109/ROBIO.2013.6739885.

[18] M.R.U. Saputra, Widyawan, G.D. Putra, and P.I. Santosa, "Indoor Human Tracking Application Using Multiple Depth-Cameras," *2012 Int. Conf. Adv. Comput. Sci., Inf. Sys. (ICACSIS)*, 2012, pp. 307–312.

[19] A. Kirillov (2011) "From Glyph Recognition to Augmented Reality," [Online], https://www.codeproject.com/Articles/258856/From-glyph-recognition-to-augmented-reality, access date: 12-Oct-2022.

[20] D.R. Bolgiano and E.D. Meeks, "A Laser Cane for the Blind," *IEEE J. Quantum Electron.*, Vol. 3, No. 6, pp. 268–268, Jun. 1967, doi: 10.1109/JQE.1967.1074528.

[21] D.I. Ahlmark, H. Fredriksson, and K. Hyyppä, "Obstacle Avoidance Using Haptics and a Laser Rangefinder," *IEEE Workshop Adv. Robot., Its Soc. Impacts (ARSO)*, 2013, pp. 76–81, doi: 10.1109/ARSO.2013.6705509.

[22] T. Hiramoto, T. Araki, and T. Suzuki, "Infrared Guided White Cane for Assisting the Visually Impaired to Walk Alone," *2022 Int. Conf. Mach. Learn., Cybern. (ICMLC)*, 2022, pp. 276–280, doi: 10.1109/icmlc56445.2022.9941336.

[23] E.B. Kaiser and M. Lawo, "Wearable Navigation System for the Visually Impaired and Blind People," *2012 IEEE/ACIS 11th Int. Conf. Comput. Inf. Sci.*, 2012, pp. 230–233, doi: 10.1109/ICIS.2012.118.

[24] J.M. Benjamin, "The Laser Cane," *Bull. Prosthet. Res.*, pp. 443–450, 1974.

[25] L.A. Guerrero, F. Vasquez, and S.F. Ochoa, "An Indoor Navigation System for the Visually Impaired," *Sens.*, Vol. 12, No. 6, pp. 8236–58, Jun. 2012, doi: 10.3390/s120608236.

[26] A. Riener and H. Hartl, "'Personal Radar': A Self-governed Support System to Enhance Environmental Perception," *Proc. 26th Annu. BCS Interact. Spec. Group Conf. People, Comput.*, 2012, pp. 147–156.

[27] Y. Wei and M. Lee, "A Guide-Dog Robot System Research for the Visually Impaired," *IEEE Int. Conf. Ind. Technol. (ICIT)*, 2014, pp. 800–805, doi: 10.1109/ICIT.2014.6894906.

[28] A. Aladren, G. Lopez-Nicolas, L. Puig, and J.J. Guerrero, "Navigation Assistance for the Visually Impaired Using RGB-D Sensor with Range Expansion," *IEEE Syst. J.*, Vol. 10, No. 3, pp. 922–932, Sep. 2016, doi: 10.1109/JSYST.2014.2320639.

[29] D. Dakopoulos and N.G. Bourbakis, "Wearable Obstacle Avoidance Electronic Travel Aids for Blind: A Survey," *IEEE Trans. Syst., Man, Cybern. Part C (Appl., Rev.)*, Vol. 40, No. 1, pp. 25–35, Jan. 2010, doi: 10.1109/TSMCC.2009.2021255.

[30] D. Bernabei *et al.*, "A Low-Cost Time-Critical Obstacle Avoidance System for the Visually Impaired," *2011 Int. Conf. Indoor Position., Indoor Navig. (IPIN)*, 2011, pp. 21–23.

[31] V. Mohandas and R. Paily, "Stereo Disparity Estimation Algorithm for Blind Assisting System," *CSI Trans. ICT*, Vol. 1, No. 1, pp. 3–8, Mar. 2013, doi: 10.1007/s40012-012-0004-y.

[32] R.G. Praveen and R.P. Paily, "Blind Navigation Assistance for Visually Impaired Based on Local Depth Hypothesis from a Single Image," *Procedia Eng.*, Vol. 64, pp. 351–360, 2013, doi: 10.1016/j.proeng.2013.09.107.

[33] C. Lee, Y. Su, and L. Chen, "An Intelligent Depth-Based Obstacle Detection System for Visually-Impaired Aid Applications," *2012 13th Int. Workshop Image Anal. Multimed. Interact. Serv.*, May 2012, pp. 1–4, doi: 10.1109/WIAMIS.2012.6226753.

[34] T.K. Chuan, M. Hartono, and N. Kumar, "Anthropometry of the Singaporean and Indonesian Populations," *Int. J. Ind. Ergonom.*, Vol. 40, No. 6, pp. 757–766, Nov. 2010, doi: 10.1016/j.ergon.2010.05.001.

[35] M.R.U. Saputra, Widyawan, and P.I. Santosa, "Obstacle Avoidance for Visually Impaired Using Auto-Adaptive Thresholding on Kinect's Depth Image," *2014 IEEE 11th Int. Conf. Ubiquitous Intell., Comput., 2014 IEEE 11th Int. Conf. Autonomic, Trusted Comput., 2014 IEEE 14th Int. Conf. Scalable Comput., Commun., Its Associated Workshops*, 2014, pp. 337–342, doi: 10.1109/UIC-ATC-ScalCom.2014.108.

[36] N. Otsu, "A Threshold Selection Method from Gray-Level Histograms," *IEEE Trans. Syst. Man, Cybern.*, Vol. 9, No. 1, pp. 62–66, Jan. 1979, doi: 10.1109/TSMC.1979.4310076.

[37] L. Chen, X. Nguyen, and C. Liang, "Object Segmentation Method Using Depth Slicing and Region Growing Algorithms," *Int. Conf. 3D Syst., Appl. Gen.*, 2010, pp. 4–7.

[38] F.Y. Shih, *Image Processing and Pattern Recognition: Fundamentals and Techniques*, 4th ed. Hoboken, USA: Wiley-IEEE Press, 2010.

[39] *Kinect for Windows, Human Interface Guidelines v1.7*, Microsoft, Redmond, WA, USA, 2013.

[40] X. Guo *et al.*, "A Survey on Fusion-Based Indoor Positioning," *IEEE Commun. Surv., Tut.*, Vol. 22, No. 1, pp. 566–594, 2020, doi: 10.1109/COMST.2019.2951036.

[41] J. Han, L. Shao, D. Xu, and J. Shotton, "Enhanced Computer Vision with Microsoft Kinect Sensor: A Review," *IEEE Trans. Cybern.*, Vol. 43, No. 5, pp. 1318–1334, Oct. 2013, doi: 10.1109/TCYB.2013.2265378.