Bulletin of Electrical Engineering and Informatics
Vol. 10, No. 5, October 2021, pp. 2811~2820
ISSN: 2302-9285, DOI: 10.11591/eei.v10i5.3170

Goal location prediction based on deep learning using RGB-D camera

Heba Hakim[1], Zaineb Alhakeem[2], Salah Al-Darraji[3]
[1]Department of Computers Engineering, Basrah University, Iraq
[2]Department of Communication Engineering, Iraq University College, Iraq
[3]Department of Computer Sciences, Basrah University, Iraq

Article Info

Article history:

Received May 26, 2021
Revised Jul 30, 2021
Accepted Aug 31, 2021

Keywords:

Computer vision
Deeep learning
Depth sensor
Object detection
Object recognition

ABSTRACT

In the navigation system, the desired destination position plays an essential role since the path planning algorithms takes a current location and goal location as inputs as well as the map of the surrounding environment. The generated path from path planning algorithm is used to guide a user to his final destination. This paper presents a proposed algorithm based on RGB-D camera to predict the goal coordinates in 2D occupancy grid map for visually impaired people navigation system. In recent years, deep learning methods have been used in many object detection tasks. So, the object detection method based on convolution neural network method is adopted in the proposed algorithm. The measuring distance between the current position of a sensor and the detected object depends on the depth data that is acquired from RGB-D camera. Both of the object detected coordinates and depth data has been integrated to get an accurate goal location in a 2D map. This proposed algorithm has been tested on various real-time scenarios. The experiments results indicate to the effectiveness of the proposed algorithm.

Corresponding Author:

Heba Hakim
Department of Computers Engineering
Basrah University, Iraq-Basrah
Email: hebah.hakem@gmail.com

1. INTRODUCTION

Person with healthy vision determines his orientation in the surrounding environment, moves from one place to another and distinguishes things and places without any difficult. Unfortunately, visually impaired person (VI) encounters many problems in his daily life. Therefore, numerous navigation assistive devices have been implemented to aim VI person and increase his self-confidence. [1]-[5]. The navigation system means the ability to find a current location, generating the optimal path to the desired destination. In order to achieve a full autonomous navigation system, full information must be provided such as current location, destination location and a map of the surrounding environment. Some navigation systems use simultaneous localization and mapping (SLAM) [6]-[9] approaches to construct 2D or 3D map of their surroundings. These approaches concern with constructing a map of unknown environment depending on the acquired data from the sensor and simultaneously compute the mobility system's position within a map. Some of SLAM algorithms depend on LiDAR (light detection and ranging) [10]-[12] while others based on RGB-D camera [13]-[15]. Since the coordinates of the goal in navigation system is needed, the object detection method with data of a RGB-D camera has been used in this work to achieve that. In recent years, convolution neural networks have been applied with major breakthrough success to recognition, detection, and segmentation of objects in images. Object detection is one of an important task in computer vision that

deals with recognizing objects of a certain class (i.e. car, animal, human) and localizing them in digital image. Now, object detection has widely been used in various real-time applications (i.e. robot vision, autonomous driving). There are many methods of object detection that uses convolution neural networks (CNNs) [16], [17] such as R-CNN [18], fast R-CNN [19], faster R-CNN [20], single shot multibox detector (SSD) [21], you only look once (YOLO) [22]. Figure 1 and Figure 2 provide the comparison on the common object detection methods accuracy and speed, respectively. As shown from these figures, YOLO v2 [23] approach provides a good tradeoff between the precision and speed. So, this approach has been used in our proposed algorithm.

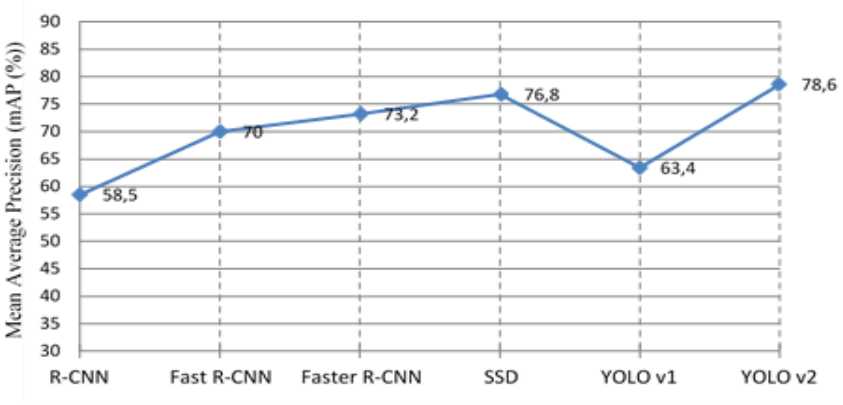


Figure 1. The accuracy of common object detection method on pascal VOC dataset

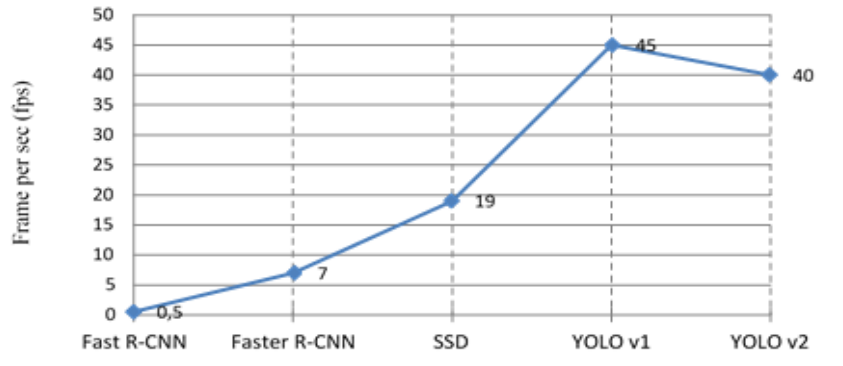


Figure 2. The speed of common object detection method on pascal VOC dataset

2. THE PROPOSED SYSTEM

Any navigation system required a 2D map, current location of user and final destination to generate a path that will be followed. Both of the current position of user and 2D map of unknown surrounding environment are not a focus of this paper, as there are many SLAM methods available for that. Therefore, to specify the coordinate of a desired goal that was selected by VI person, a proposed algorithm that combines both the results of object detection method based on deep-learning and depth information from RGB-D camera is implemented in this work.

The proposed algorithm depends on RGB-D camera mounted on a head as shown in Figure 3 that provides color image and depth data for each pixel in an image. The color image will be used as input to the object detection method (YOLO v2) in order to predict the class of the detected object and the location of the recognized object (i.e. bounding box that gives an object class coordinates) in an image. However, the depth information is used to obtain the distance between the recognized object and user. Both the depth information and coordinates of the recognized object in an image are integrated in order to obtain the location of an object (that is considered as a goal) in a 2D map that is used in the navigation system.

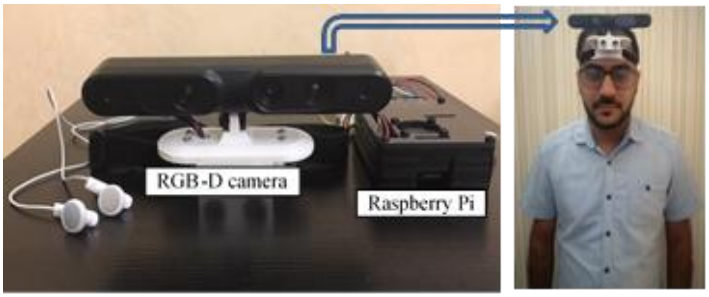


Figure 3. Assistive wearable device using RGB-D camera

The block diagram of proposed algorithm in Figure 4 starts with applying object detection method to recognize the object class. After that, the type of the recognized object is converted to voice command by text-to-speech (TTS) process (ESpeak TTS method) to notify visually impaired person with information of his surroundings. The visually impaired person selects his desired goal among the recognized objects on the voice command through the building microphone of the RGB-D camera. The speech recognition module is used to convert the VI person's voice to text in order to match the selected destination with any of the recognized object types. The coordinates of bounding box for the selected destination and its depth are taken to obtain a goal coordinate to the navigation system.

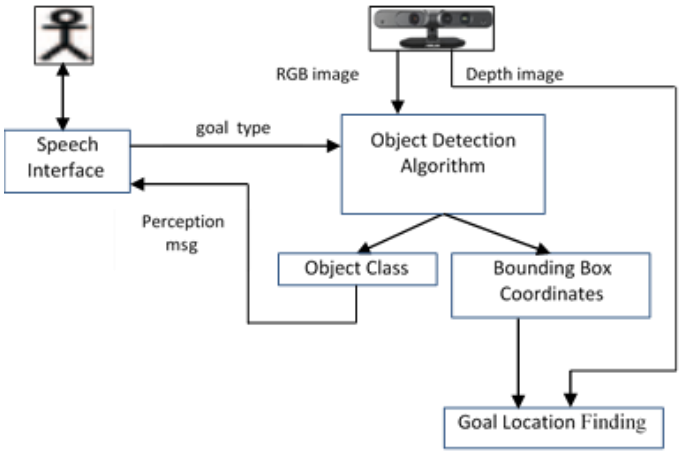


Figure 4. The block diagram of the proposed algorithm

2.1. RGB-D camera

Asus Xtion sensor is one of the most popular active 3D-camera which is introduced in 2012 by a company of PrimeSense in collaboration with Asus company. The first generation is Asus Xtion Pro that only provides a depth camera. After that, the Xtion Pro Live version has been released which provides RGB camera and depth ..data. It is a structured light camera where the principle of triangulation is used to measure the depth for each pixel. Asus Xtion Pro Live comprised of a VGA camera to capture RGB image, depth sensor measures object distance to the camera to provide a depth image, and 2 microphones. The specification of Asus Xtion Pro Live listed in Table 1.

Table 1. Asus Xtion Pro Live camera specification

Specifications	
RGB image/depth map Resolution	640 × 480 pixels
Vertical viewing angle	45°
Horizontal viewing angle	58°
Accurate distance range	0.8m – 3.5m
size	170 gm
weight	18×3.5×5 cm
cost	~ 150$

With this sensor, it is possible to acquire the color image with 3 channels (red, green, and blue), each channel is represented by 8 bits, and also depth information in 11 bits for each pixel in RGB-image. The depth data is expressed in millimeters where the value zero represents that there is no depth information at that corresponding pixel. The raw depth map is determined by an IR laser source and an IR camera (CMOS sensor). The IR laser source emits the IR light in the form of a known dot pattern (speckles pattern) by the diffraction grating on the scene in front of the camera and IR camera reads these reflected spackles. After that, the depth sensor processor receives the speckle pattern and computes the depth value by correlating the captured pattern with a stored reference pattern which is located on the plane with a known depth to the sensor. The depth map is the output of the depth sensor processor. To express the object points 3D coordinates, the coordinate system of a depth with its origin is considered at the perspective center of the IR camera as shown in Figure 5.

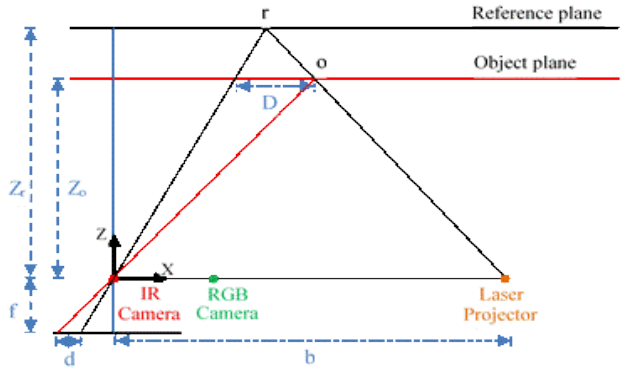


Figure 5. The schematic representation for relation of depth disparity

The Z-axis is perpendicular to an image plane in the direction of the object, the X-axis is orthogonal to Z-axis towards the baseline b which represents the distance between the laser projector and IR camera center, and Y-axis is perpendicular to Z and X making a right-handed coordinate system. Assume that the point o of an object is on the reference plane at a depth Z_r to the camera and a speckle on an object is captured on the IR camera image plane. The speckle location on the image plane is shifted in the direction of X once the object is displaced far away from/close to the sensor. This shift is measured as disparity (pixel offset) d in the image space. From the triangles similarity, the following can be calculated [24], [25]:

$$\frac{D}{b} = \frac{Z_r - Z_o}{Z_r} \tag{1}$$

$$\frac{D}{d} = \frac{Z_o}{f} \tag{2}$$

where Z_o represents the depth (distance) of a point o in an object space, f is focal length of IR camera, and D denotes the point o displacement in object space. Z_o is expressed in (3) by substituting D from (1) into (2):

$$Z_o = \frac{Z_r}{1 + \frac{Z_r}{fb} d_f} \tag{3}$$

The Asus Xtion Pro live sensor in our system works mostly for indoor environment, as the depth measurement influenced by the direct sunlight. In fact, this camera is not totally unused in outdoor environments, but it can be used for a cloudy day or a night scene. Asus Xtion Pro Live is with light weight and small size that allows it suitable to be mounted. Also, the USB 2.0 port of Raspberry pi board powers the camera by 5v. It transmits RGB-image and depth data per pixel to the system by using the open source library OpenNI2. OpenNI2 driver is used for sensor interfacing by providing wrappers to many languages including python. To visualize the RGB and depth image, OpenCV is used by opening ' frame-data'. The 2 microphones of Asus Xtion Pro live distributed on the both sides of the sensor. Each microphone operates in audio stream of 16-bit with a 16 kHz sampling rate. In this work, the camera's microphone is used to select a final goal among recognized objects by VI person.

2.2. You only look once v2 (YOLO v2)

YOLO v2 is an improved approach to YOLO v1 in which the advantage on speed is kept and the mAP (mean average precision) value of YOLO v1 (63.4%) is increased. In general, YOLO uses single convolution neural network to predict more than one box and probabilities of class for these boxes. It splits an image into grid cells of S×S. Each grid is responsible of detecting an object if the object center falls into this grid cell.

The output tensor of YOLO v2 is with size S×S×(B(5+C)), since each bounding box (BB) predicts: (1) confidence score which is used to weight this bounding box; (2) the center of bounding box (x,y) relative to the grid cell and dimension of bounding box (h,w); (3) C class probabilities. YOLO v2 is developed in order to improve the accuracy by adding Batch normalized in the convolution layer. Besides, it trains the classifier with images of 224×224 then fine tune the classifier with images of 448 × 448 using much fewer epochs instead of YOLO training that trains the classifier with images of 224 × 224 then the resolution is increased to 448 for detection. Also, the idea of anchor box is used to get the best shapes of anchor box that will be used in predicting. Moreover, the input image was shrunk to 416×416 to get odd number of locations in the map of feature (i.e. a single center grid cell).

In this work, object detection method depends on YOLO v2; (1) to allow VI person to classify and distinguish the objects appearing in his path; (2) to provide the location of the classified object that is represented as a final goal to the navigation system. Tiny YOLO v2 with weight that is pre-trained on PascalVOC dataset has been specifically adopted in our work. This method is the smallest version of the complicated YOLO v2. It is lite and faster than original algorithm since it consists of fewer layers that makes it suitable to apply on Raspberry Pi. Figure 6 shows Tiny YOLO v2 architecture that consists of 9 convolution layers with 6 pooling layers.

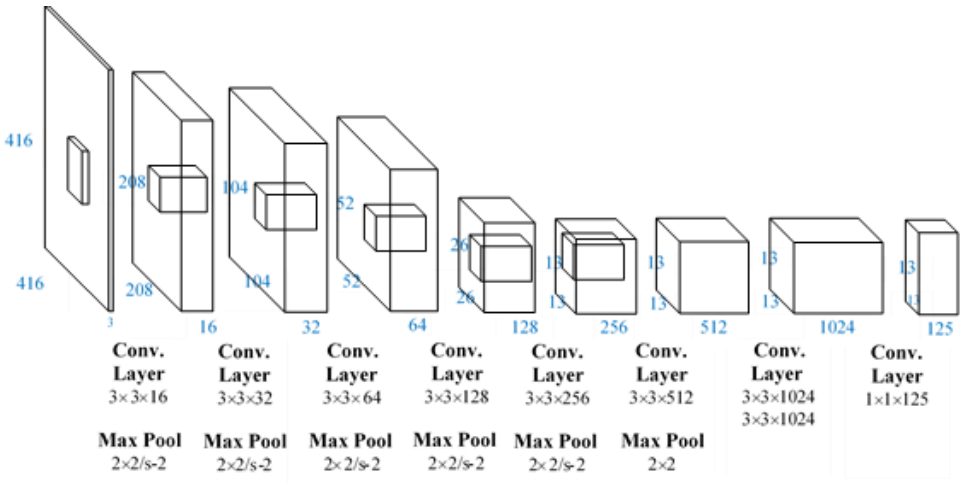


Figure 6. Tiny YOLO v2 archeitecture

2.3. The algorithm steps

This subsection illustrated the proposed algorithm of finding goal coordinates on 2D grid map (represents the 2D occupancy map of the surrounding environment). The proposed algorithm starts with acquiring the RGB-image in a front of the VI person and its depth raw data from RGB-D camera (Asus Xtion Pro Live). Firstly, the captured image is resized to (416×416) pixels since the Tiny YOLO v2 uses an image with this size as input. The value of threshold is set to 0.4 to determine if the detected object is true or not and pre-trained weight on Pascal VOC data set is loaded. After that, Tiny YOLO v2 is applied to detect the object category among 20 categories of the dataset (Pascal VOC) and make the bounding box around the object. The two outputs of the detection process as shown in Figure 7 are:

a. The category of the recognized object, which is converted to voice message by ESpeak TTS method to notify VI person with the enviroment information to select one of the recognized objects as a goal;
b. Set of coordinates for bounding box (BB) (x-top left, y-top left, x-bottom right, and y-bottom right).

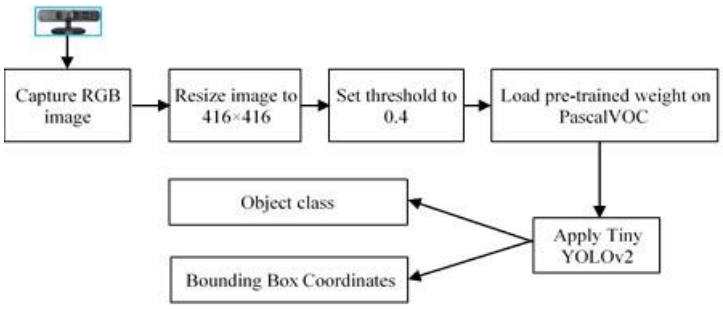


Figure 7. The process of YOLO v2 on an image

After this process, VI person tells the system where he wants to go (i.e. desired goal) via voice through the built-in microphone of a depth camera. The voice of VI person is converted by Google speech-recognition method to text to match the desired goal with one of the detected object classes. The y-top left coordinate of BB for the selected goal is taken to be used to know the direction of the object with respect to VI stand (i.e. object lies on the right or left of VI person). This is done by dividing RGB-image into two equal regions (right region and left region) as shown in Figure 8 and examined y-top left value where it lies. If this value is between 1 and 208, then the selected object lies on the left of VI person's current position. As opposite, when this value lies between 209 and 416 that mean the selected object is detected on the right of VI person standing. While, the minimum depth from depth raw data of this bounding box is used to know the distance between the VI person and the selected object. Since, the measured depth value is in millimeter (mm), the following is applied in order to convert it to pixel value on a 2D map:

$$Depth\ (in\ pixel) = \frac{Depth\ in\ m*1000}{2D\ map\ resolution} + origion \quad (4)$$

Where, 2D map resolution represents a grid cell length in (m/pixel).

Each grid cell has a probability of being occupied: (a) black color refers to an occupied cell; (b) white color refers to a free cell; (c) dark grey color cell refers to area that was not scanned. An obstacle (occupied cell) in a 2D map is with value 1 while a free cell with value 0.

Thus, all required information to know the coordinate of a selected goal have been available. The minimum depth value (in pixel) of bounding box for the selected object indicates to x-coordinate of the selected goal on 2D map. To know the y-coordinate of the selected goal on grid map, all pixels values of the row with the goal x-coordinate in the right or left of VI person' current position are examined as shown in Figure 9. It should be noted that the right area or left area relative to the current position of VI person is specified based on a previous defined area as described in Figure 8. The flowchart of the proposed algorithm has been described in details in Figure 10 that used the data from RGB-D sensor as input and outputs the desired goal coordinates on 2D occupancy grid map in real time.

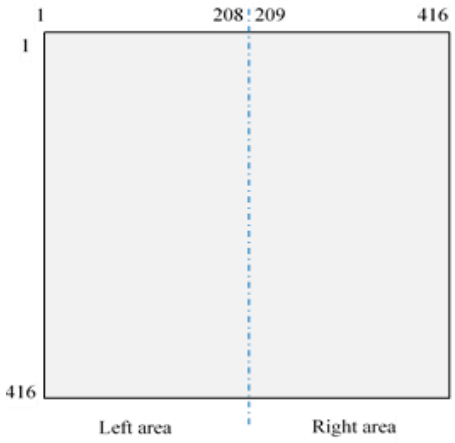


Figure 8. The spliting RGB-image into two areas

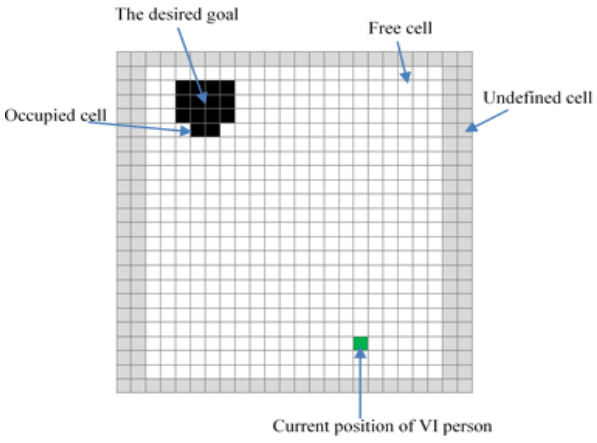


Figure 9. 2D grid map representation

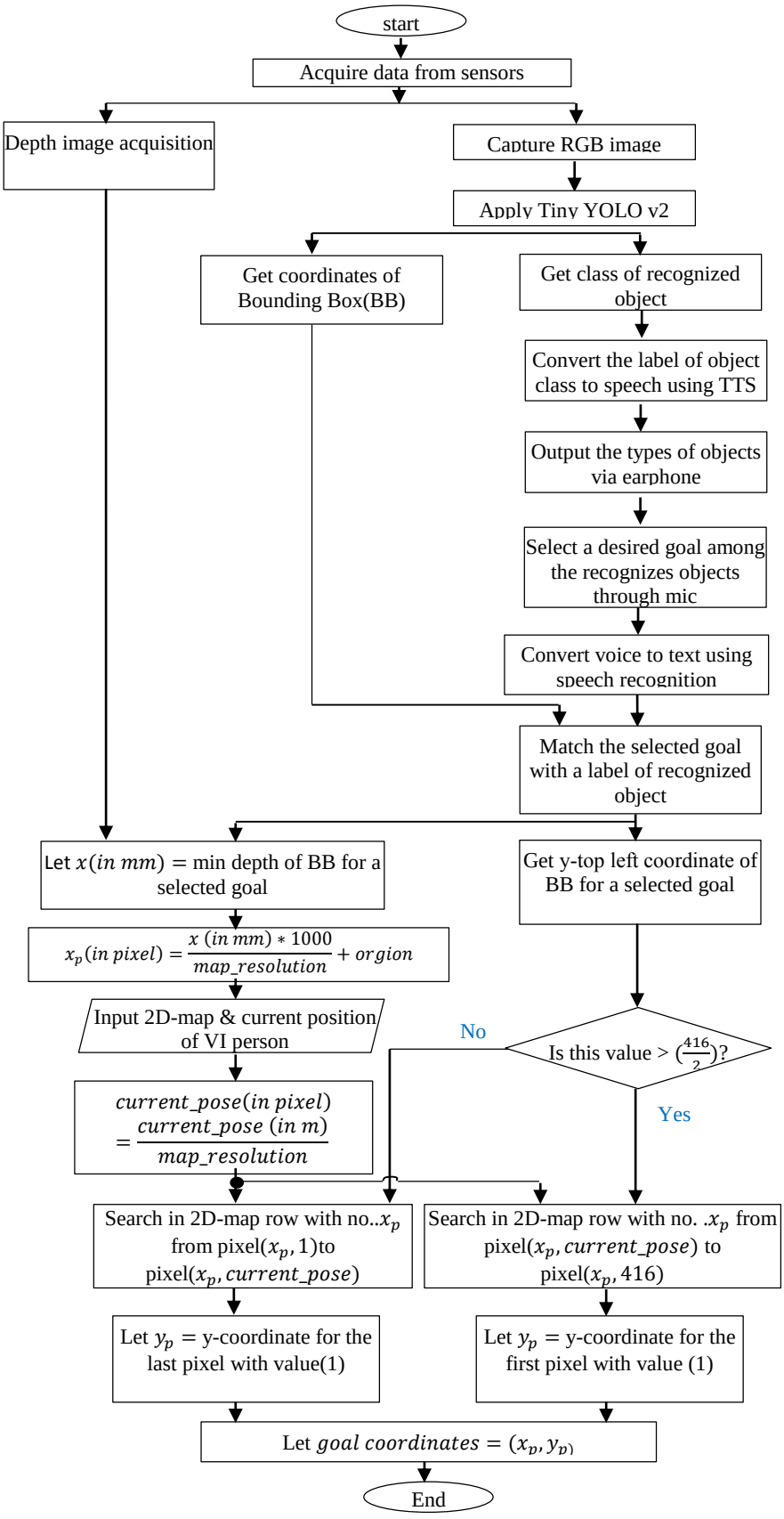


Figure 10. The proposed algorithm flowchart

3. THE EXPERIMENTS AND RESULTS

The proposed algorithm of finding goal coordinates on 2D-map has been evaluated in different real-time scenarios in the indoor environments. It implemented on Raspberry pi 3 B+ with python-programming language using Tensorflow1.4, OpenCV2, and OpenNI2 library by using RGB-D camera mounted on the head. The proposed algorithm is based on the depth sensor that used IR camera. Therefore, the light intensity of an environment may affect its performance. The measurement errors were plotted in Figure 11 shows the capability of RGB-D camera to work under which condition in an indoor environment. The RGB-D camera was placed in different distances away from the object under different light condition. With OpenNI2 library, it becomes easy to access to a depth value of each pixel as an array. These depth information were written to a csv file by python language in order to overview and evaluate the depth map obtained by depth sensor.

In Figure 11, the x-axis indicates to the actual distance from the sensor to the obstacle in (m) and y-axis referred to the maximum deviation value in (m). The deviation was recorded every 0.5 m. As seen from the overall measurements results, the deviation between the actual depth and calculated depth is very small. This makes Asus Xtion Pro live a quite useful for our work due to the astonishing precision comparing with its low cost.

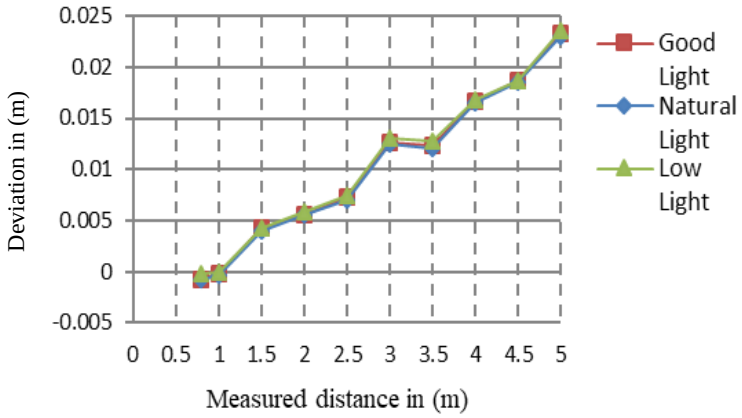
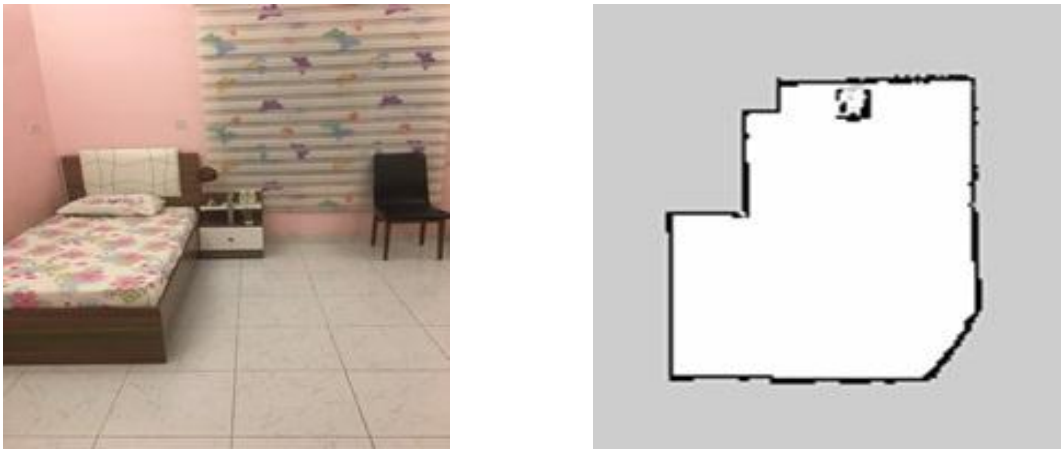


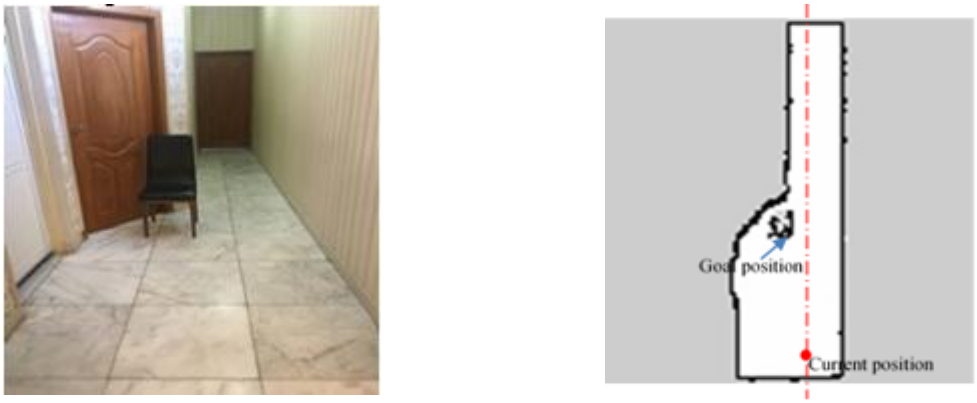
Figure 11. Lighting influence on depth data

In the real-time scenarios in Figures 12 and 13, the coordinates of a desired goal on 2D occupancy grid map has been determined after a VI person selects the chair as a desired goal. This goal coordinates will be used in the navigation system to guide VI person from his current position to the desired goal. A red node in Figure 12-b and 13-b represents the current position of VI person and a blue arrow pointed to a goal position. These nodes will be taken as start point and end point in path planning module of the navigation system.



(a) (b)

Figure 12. First scenario; (a) real image on 2D map, (b) goal position on 2D map



(a) (b)

Figure 13. Second scenario; (a) real image on 2D map, (b) goal position on 2D map

4. CONCLUSION

This paper presents a proposed algorithm of finding goal coordinates on a 2D occupancy map. This proposed algorithm has been implemented in order to use it in the indoor navigation system used to help visually impaired people reach to the desired destination within unknown indoor environments. The proposed algorithm depends on deep-learning based object detection method using RGB-D camera, which runs on a lightweight and low-cost main processing platform. Also, the used sensor has the characteristics of small size, lightweight, and low cost. Thus, it has great potential to be used in a wearable navigation system for a visually impaired person. The results of experimental verified that the proposed algorithm was effective on specifying a desired destination coordinates on a 2D map in a real time.

REFERENCES

[1] H. Hakim and A. Fadhil, "Indoor low cost Assistive device using 2D SLAM based on LiDAR for Visually Impaired People," *Iraqi Journal of Electrical and Electronic Engineering*, vol. 15, no. 2, pp. 115-121.

[2] M. M. Islam, M. Sheikh Sadi, K. Z. Zamli, and M. M. Ahmed, "Developing Walking Assistants for Visually Impaired People: A Review," *IEEE Sensors Journal*, vol. 19, no. 8, pp. 2814-2828, 2019, doi: 10.1109/JSEN.2018.2890423.

[3] H. Hakim and A. Fadil, "Navigation system for visually impaired people based on RGB-D camera and ultrasonic sensor," *ACM International Conference on Information and Communication Technology*, pp. 172-177, 2019, doi: 10.1145/3321289.3321303.

[4] S. Real and A. Araujo, "Navigation Systems for the Blind and Visually Impaired Past Work, Challenges, and Open Problems," *Sensors*, vol. 19, no. 15, pp. 1-20, 2019, doi: 10.3390/s19153404.

[5] H. Hakim and A. Fadil, "Indoor Wearable Navigation System Using 2D SLAM Based on RGB-D Camera for Visually Impaired People," *Springer series: Advances in Intelligent Systems and Computing*, vol. 1292, pp. 661-672, 2020.

[6] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: part I," *IEEE Robotics & Automation Magazine*, vol. 13, no. 2, pp. 99-110, June 2006, doi: 10.1109/MRA.2006.1638022.

[7] T. Bailey and H. Durrant-Whyte, "Simultaneous localization and mapping (SLAM): part II," *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 108-117, 2006, doi: 10.1109/MRA.2006.1678144.

[8] C. Cadena, *et al*., "Past, Present, and Future of Simultaneous Localization and Mapping: Toward the Robust-Perception Age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309-1332, 2016, doi: 10.1109/TRO.2016.2624754.

[9] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A solution to the simultaneous localization and map building (SLAM) problem," *IEEE Transactions on Robotics and Automation*, vol. 17, no. 3, pp. 229-241, June 2001, doi: 10.1109/70.938381.

[10] K. Konolige, G. Grisetti, R. Kümmerle, W. Burgard, B. Limketkai, and R. Vincent, "Efficient Sparse Pose Adjustment for 2D mapping," *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2010, pp. 22-29, doi: 10.1109/IROS.2010.5649043.

[11] B. Steux and O. E. Hamzaoui, "tinySLAM: A SLAM algorithm in less than 200 lines C-language program," *11th International Conference on Control Automation Robotics & Vision*, 2010, pp. 1975-1979, doi: 10.1109/ICARCV.2010.5707402.

[12] S. Kohlbrecher, O. von Stryk, J. Meyer and U. Klingauf, "A flexible and scalable SLAM system with full 3D motion estimation," *IEEE International Symposium on Safety, Security, and Rescue Robotics*, 2011, pp. 155-160, doi: 10.1109/SSRR.2011.6106777.

[13] F. Endres, J. Hess, J. Sturm, D. Cremers and W. Burgard, "3-D Mapping With an RGB-D Camera," *IEEE Transactions on Robotics*, vol. 30, no. 1, pp. 177-187, Feb. 2014, doi: 10.1109/TRO.2013.2279412.
[14] Mur-Artal, J. D. Tardós, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255-1262, 2017, doi: 10.1109/TRO.2017.2705103.
[15] M. Labbe, F. Michaud, "RTAB-Map As an Open-source Lidar and Visual Simultaneous Localization and Mapping Library for Large-scale and Long-term Online Operation," *Journal of Field Robotics*, vol. 36, no. 2, pp. 416-446, 2018, doi: 10.1002/rob.21831.
[16] H. Hakim A. Fadil, "Survey Convolution Neural Networks in Object Detectrion," *Journal of Physics: Conference Series*, no. 1804, pp. 1-19, 2021, doi: 10.1088/1742-6596/1804/1/012095.
[17] N. Aloysius and M. Geetha, "A review on deep convolutional neural networks," *International Conference on Communication and Signal Processing ICCSP*, 2017, pp. 0588-0592, doi: 10.1109/ICCSP.2017.8286426.
[18] R. Girshick and J. Donahue, T. Darrell, and J. Malik, "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," *IEEE International Conference on Computer vision an Pattern Recognition*, 2014, pp. 580-587.
[19] R. Girshick, "Fast r-cnn," *IEEE International Conference on Computer Vision ICCV*, 2015, pp. 1440-1448.
[20] S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1137-1149, 2017, doi: 10.1109/TPAMI.2016.2577031.
[21] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C. Fu, and A. C. Berg, "SSD: Single Shot MultiBox Detector," *European Conference on Computer Vision*, arXiv:1512.02325v5, 2016, pp. 21-37.
[22] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," *IEEE International Conference on Computer Vision and Pattern Recognition CVPR*, 2016, pp.779-788.
[23] J. Redmon and A. Farhadi, "YOLO9000: better, faster, stronger," *IEEE International Conference on Computer Vision and Pattern Recognition CVPR*, 2017, pp. 7263-7271.
[24] T. Jia, Z. Zhou, and H. Gao, "Depth Measurement Based on Infrared Coded Structured Light," *Hindawi Publishing Corporation Journal of Sensors*, vol. 2014, ID. 852621, pp. 1-9, 2014, doi: 10.1155/2014/852621.
[25] K. Khoshelham and S. Elberink, "Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications," *Sensors*, vol. 12, no. 2, pp. 1437-1454, 2012, doi: 10.3390/s120201437.