Autonomous Indoor Pathfinding using Neural Network in Complex Scenes

Vignesh Vasudevan, Guojun Yang and Jafar Saniie

Department of Electrical and Computer Engineering
Illinois Institute of Technology
Chicago, Illinois, United States

***Abstract*— Navigation to a specific destination indoors can be a challenge due to different reasons such as visual impairment, unknown environments, etc. There has been much work done to solve this issue such as indoor positioning systems, navigation using sensors and even using a robotic guide. In this paper, a novel and straightforward method of path planning (including object avoidance) is presented as a way of navigating to a desired location within a complex environment. The system proposed uses the combination of depth information from an RGB-D camera and the object information from a Neural Network based object identification technique, to efficiently calculate and plan a path in real-time, to a pre-specified destination. Persons to be helped are identified using object detection, and the most practical path to the desired destination is calculated. The path information would be sent to the handheld device of the person being helped in the suitable form of interface, such as visual, audio, etc. The surveillance type nature of the system enables it to help multiple persons in the same area. The model was tested in a controlled environment with one individual person being guided to nearby specified locations. While the testing showed promising results, strong conclusions are yet to be made with the current system.**

I. INTRODUCTION

Navigation and General Positioning systems have come a long way after the advent of the Information age. Smarter and reliable technological innovations have helped in making such systems more accurate, the GPS can pinpoint the location of a person to within 5 meters, and more accessible, anyone can access and locate their positions by just using a handheld device. General outdoor navigation has become trivial due to the development of such technologies.

Although the objective may be similar, indoor navigation poses a different kind of challenge. Enclosed environments, such as buildings, makes it more difficult to obtain information about the exact layout of the structure and locations of each specified destination. Traditional positioning systems such as GPS are highly inefficient in such situations, including the fact that they are not able to properly determine the elevation of a person's current location, making it ineffective in a multistoried building. Targeted users are also to be considered while designing such systems. Indoors navigation can be an obstacle to persons with disabilities. This necessitates the user accessibility of the system and widens the scope of factors to be considered while developing it.

There has been much work done to put in place a ubiquitous system, as well received and adapted as the GPS system, but none have made it so far. One of the most popular methods has been a Wi-Fi Localization system [1], which exploits the ever-present facility of Wi-Fi to determine a user's position. Yet another method is with a smartphone, using its onboard sensors and accessing a modelled indoor map through the web to achieve its purpose [2]. The exciting field of Machine Vision, enabling machines to make smart decisions based on its visual information, has also opened up possibilities of a more intuitive approach for navigation, especially in the field of robotics.

This paper presents a system which can be incorporated into surveillance cameras, and the navigation information can be obtained on a handheld device. For experimental purposes of the system, it was implemented using an RGB-D camera (Microsoft Kinect) and PC. In tandem with a Convolutional Neural Network (CNN) based object recognition method (YOLOv2 [3]), this system was able to model a general outline of the observed environment and determine a dynamic path from the person to the desired location, which was visualized in real-time on an OpenGL rendering window to help with error-correction and debugging.

Section II briefly discusses some of the various other works done in addressing the issue of indoor navigation, while considering their targeted users and the results achieved by their systems. In comparison, the proposed method is outlined and briefly discussed. Section III delves into the overall structure of the proposed system and its components. Section IV presents the experimental results and discussions on them. Section V concludes the paper and mentions the possible applications of the system.

II. DEVELOPMENT OF INDOOR NAVIGATION SYSTEMS

A. Non-Visual Methods

Section I briefly mentions and brings attention to the various non-visual methods of localization and consequently, navigation. Exploiting the correlation of the Wi-Fi signals from routers in different positions in a building can provide an ambiguity area where a user is likely to be found. Recursively applying this function can gradually zero-in on the user's

978-1-5386-5398-2/18/$31.00 ©2018 IEEE

location [1]. It also proposes the idea for creating a semantic floor-map of the entire building to make the process of navigation more efficient. F. Shayganfar *et al.* [4] put forward a rigorous data-based approach where entire layouts of the surrounding environments are stored on an online database. Their goal was to automatically generate semantic Building Information Models (BIM) to achieve a smart and efficient way of navigation.

B. *Visual Methods*

Vision-based methods of pathfinding and navigation are being popularized as they can provide a real-time assessment of the user's environment, broadens the scope of targeted users to include individuals with disabilities such visual impairments, etc. S. Verma *et al.* [2] used indoor maps, generated by taking 360^0 panoramic pictures and stored in a web database, in tandem with a camera smartphone using the Dead-reckoning algorithm to localize the user. While the method is effective, it still requires the pre-obtained information about the environment. The challenge of low mobility of an RGB-D camera was addressed and adeptly overcome by using a single camera and running the computationally heavy Structure from Motion (SfM) algorithms to generate a 3D map of the environment to make navigation possible in [5].

T. Gonnot and J. Saniie [6] designed a system specifically for the navigation and guidance of visually impaired people. It was complete with a camera glass for obtaining video feed, a smartphone for processing information, and communication interfaces such as audio outputs and a vector vibration belt for added benefit. G. Yang and J. Saniie [7] proposed a stand-alone system which used unique Augmented Reality (AR) markers to achieve localization and navigation. By using an initial known ID as the origin, the system was able to visualize the locations of the different IDs within the real-world on a virtual renderer. R. Jiang *et al.* [8] designed a RGB-D and CNN based navigation system which provided auditory information for the user. The bulky personal set-up provided clear cut results but raised questions regarding the mobility of the system.

C. *Proposed Model*

The model proposed in this paper is vision-based, and furthermore, the fixed camera style approach addresses the problem of mobility. This also endows a multi user interface feature to the system. Customized navigational information can be sent to multiple persons individually if a proper prioritization system is implemented.

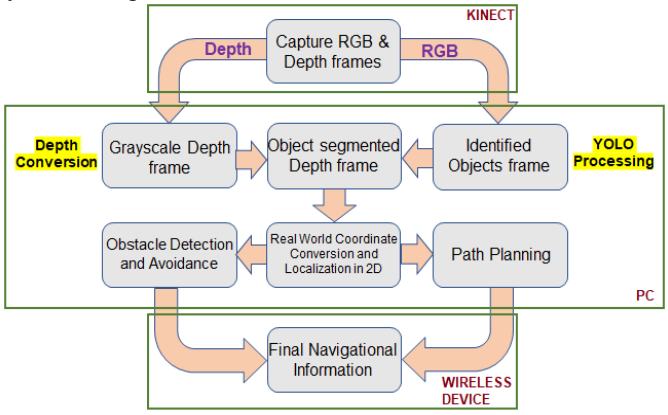

Figure 1. System Architecture

Figure 1. describes the process flow of the system. The essential component of the design is the RGB-D camera. The proposed model uses a Microsoft Kinect, capable of capturing 640x480 pixel frames at a rate of 30 Hz for both the RGB feed and IR depth feed. The processing and path-planning is done on a PC with the Nvidia GeForce GTX 1070 GPU for added computational power as running a real-time Convolutional Neural Network (CNN) Object detection is computationally expensive. The state-of-the-art method You Only Look Once (YOLO) [9] is used for object recognition. The final navigational information can be sent to a handheld of the user as audio or video information.

III. RGB-D Assisted Neural Network

In this section, the different components of the system are elaborated, and the workflow is explained.

A. *RGB-D Information*

The accuracy of the Kinect in capturing the depth information (Z coordinate) is a major advantage: it gives an average error of about a few millimeters to 4 cm for objects within its range of 0.7 m – 5 m. For the general localization and navigation of a typical human being, error of a few centimeters would be trivial. The raw depth information obtained from the Kinect was needed to be converted to real world measurements, here it was converted into millimeters. The conversion is done with the help of OpenKinect, an open source module for the management and operation of the Kinect. The conversion formula stated in their resources is given in Eq. (1). [10]

$$Distance = 0.1236 * \tan\left(\frac{rawDisparity}{2842.5 + 1.1863}\right) m \qquad (1)$$

The depth information was converted to an 8-bit grayscale image to get a better understanding of the distance of the objects from the camera. The closer pixels are assigned a lower intensity value, while farther pixels are given a higher value. The working range of the system is set from 0.5 m to 5.5 m from the camera, and all the values are calculated accordingly.

The RGB frame information is used to detect objects in the scene and obtain the necessary details of the individual objects for localization and visualization. The object detection is done by the CNN-based method YOLO discussed in Section III(*B*).

B. *CNN-based Object Detection*

For object detection, different algorithms were taken into consideration. It was determined that the algorithm You Only Look Once (YOLO) [3] was the apt method for the system. Its relatively low storage requirements, less complex architecture and consequently, faster processing speed is suitable for real-time applications. In YOLO, the input image is divided into S×S grid and each grid cell predicts B bounding boxes, their confidence scores, and detection of a class within it. The confidence score for each box is given by Eq. (2). [9]

$$\begin{aligned}\Pr(Class_i|Object) \times \Pr(Object) \times IOU_{pred}^{truth} \\ = \Pr(Class_i) \times IOU_{pred}^{truth}\end{aligned} \qquad (2)$$

where IOU_{pred}^{truth} is the intersection over union for the ground truth and the predicted class. The overall variables to be predicted can be represented as a S × S × (B × 5 + C) tensor.

The YOLO model is based on CNN architecture and the network has 24 convolutional layers and 2 fully connected layers as shown in Figure 2. The loss function of the model depends on the *x, y, w, h* of the predicted bounding boxes, and the probabilities of classes detected. For the system proposed, the pre-trained YOLO weights using the Microsoft COCO [11] database was used.

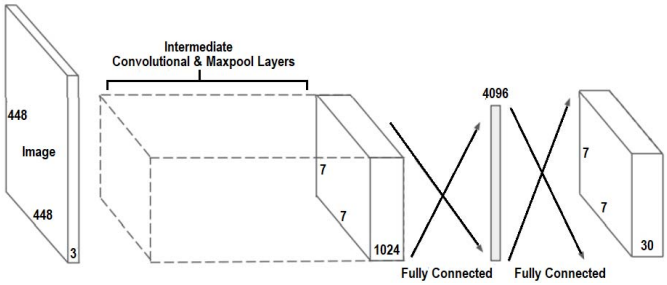

Figure 2. YOLO architecture

The detection data is combined with the depth information frame to obtain the person segmented frame and the depth and positional information of all the objects in the scene. This data is used to localize all the objects in the observed scene.

C. Localization, Path Planning and Obstacle Avoidance

The first step for achieving navigation is to localize all the objects and persons in the scene in terms of real world coordinates. The center of the Kinect is chosen to be the origin from which all other positions are calculated. For the current system, all the objects are localized to a 2D plane as the environment is constrained to an enclosed area with no changes in elevation. The pixel coordinates of the observed objects and their respective depth information are correlated to determine its real-world positions. The selection of the pixel location and depth information for each object is done by selecting the pixel in the object segmented image which strongly represents the entire features of the object. The conversion factors were roughly determined by placing certain markers in front of the camera at different depths to ascertain the distance at which the measured length of the marker in pixels was the same as the measure length in millimeters. Eq. (3) shows the obtained factors for conversion:

$$X_{mm} = \frac{Z_{mm}}{930} \times X_{px} \quad \& \quad Y_{mm} = \frac{Z_{mm}}{930} \times Y_{px} \qquad (3)$$

where Z_{mm} is the corresponding depth information of that pixel in millimeters. Using this X-Y data and its respective depth information, persons and objects are localized, and given a unique identity. In the case of a visual interface as is implemented here, persons are represented by circles and objects by pentagons. For an auditory interface, unique names containing the features of the object can be assigned. In this paper, visual interface is the primary focus.

The first person to be detected is determined to be the user and the destination is set by the user by selecting any one of the identified objects. The shortest path between the person and the destination is calculated and displayed. There are two cases for path determination: clear and blocked paths. In a clear path, since there are no objects blocking the path, the final navigational directions are straightforward, as depicted in Figure 3. Whereas in a blocked path, a blocking object makes complications for path determination, requiring the application an obstacle avoidance algorithm to obtain a skewed final path, as shown in Figure 3.

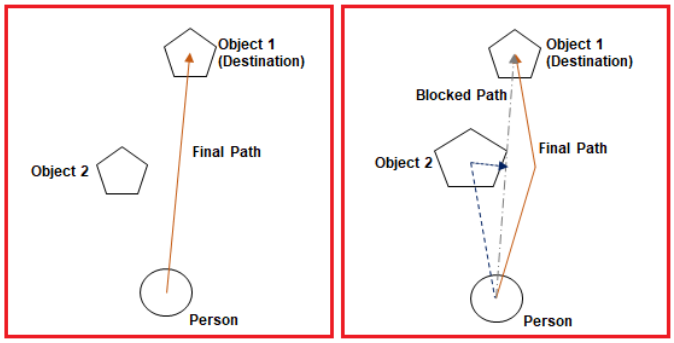

Figure 3. Clear path (left) and Blocked path (right)

For obstacle avoidance, a method using the vector from the center of the blocking object pointing towards the blocked path and perpendicular to it, is used to push the calculated blocked path to a new skewed clear path. The length of the vector determines the strength of the push; nearer objects have a shorter vector length and therefore, will have a stronger push, resulting in a more skewed bypass. As shown in the Blocked path of Figure 3, the perpendicular vector from the blocking Object 2 pushes the previously determined path further out to arrive at the Final Path. The magnitude of the vector is determined by the following steps:

- First, the distances between the Person and all other Objects in the scene, and length of the Blocked path are determined by using the localization data.
- The projections of the Person-Object distances on the Blocked path is calculated. If the length of a projection is greater than the Blocked path, or its direction is opposite, the affiliated Object is non-blocking.
- The magnitude is then calculated using the Pythagoras' theorem where the Person-Object distances and their respective projection lengths are known.

Objects with vector magnitude higher than a threshold value are considered non-blocking. The vector magnitudes are compared quantify the strength of their push, and then the final path is determined.

IV. Results and Discussion

The proposed system was tested in an enclosed room with sufficient lighting and adequate number of identifiable objects. The Kinect was properly oriented in rotation and placed on a horizontal elevated plane, high enough to clearly discern the major feature of different objects. The center of the Kinect camera was the origin from which both the depth and lateral distances were calculated. For experimental purposes, the 2D map of the scene was displayed on an OpenGL window on the PC by which the user selected the desired destination. Persons were denoted by circles and objects by pentagons. The calculated path was displayed as a line drawn from the person to the destination, and the irregularities in the line were determined by the number of active blocking objects. The following sequence of images (Figure 4, 5, 6, 7 and 8) shows the intermediate frames and the final results.

Figure 4 shows the objects identified by the YOLO object detection technique applied on the RGB image obtained from the Kinect. The threshold confidence level for detection was chosen to be around 16% ~ 20%.

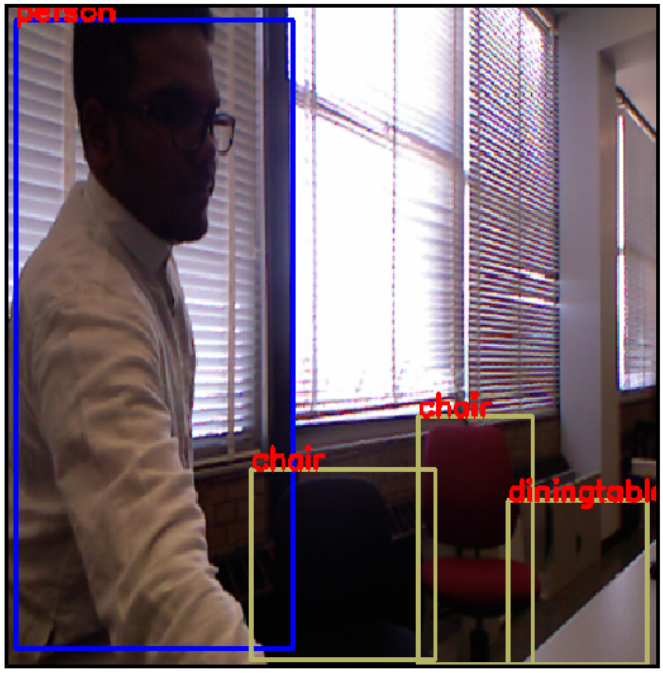

Figure 4. YOLO Identified Objects Frame
(Person, chair, chair, dining table)

Figure 5 is the corresponding depth image of the RGB frame. This was obtained by converting the depth data in millimeters into 8-bit intensity values for each pixel.

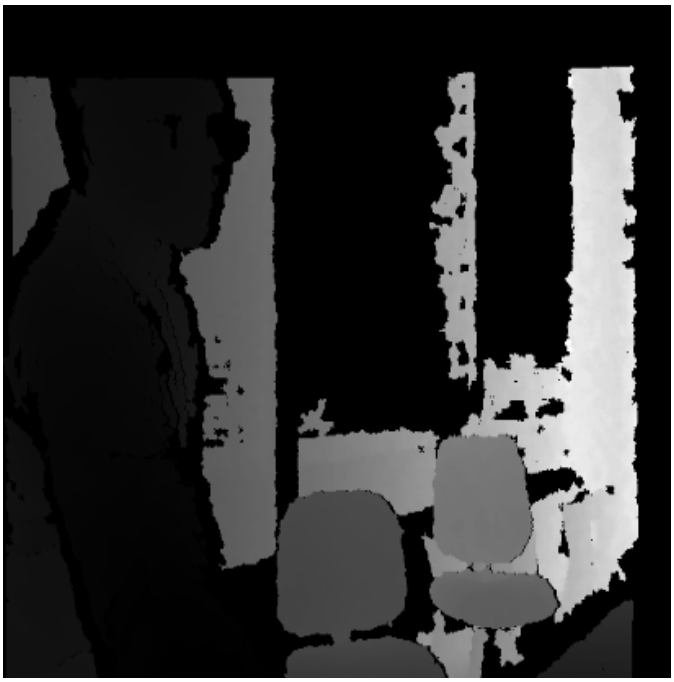

Figure 5. Kinect Depth Frame

Figure 6 shows the binary image of a segmented single user. This was done by applying a pixelwise AND operation using both the YOLO frame (Figure 4) and Depth frame (Figure 5), on just the areas of interest, i.e., only bounding boxes containing persons.

Figure 6. Person Segmented Depth Frame

When the destination was set to be the nearby chair, there were no obstacles in the path. The results were obtained as shown in Figure 7.

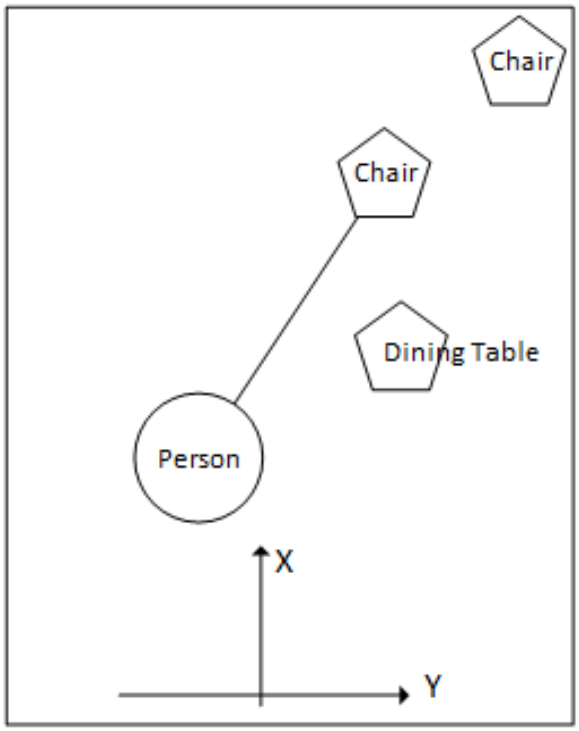

Figure 7. Clear Path

Figure 8 shows the blocked path scenario, where the destination selected was the chair in the distance. A final twisted line path was obtained.

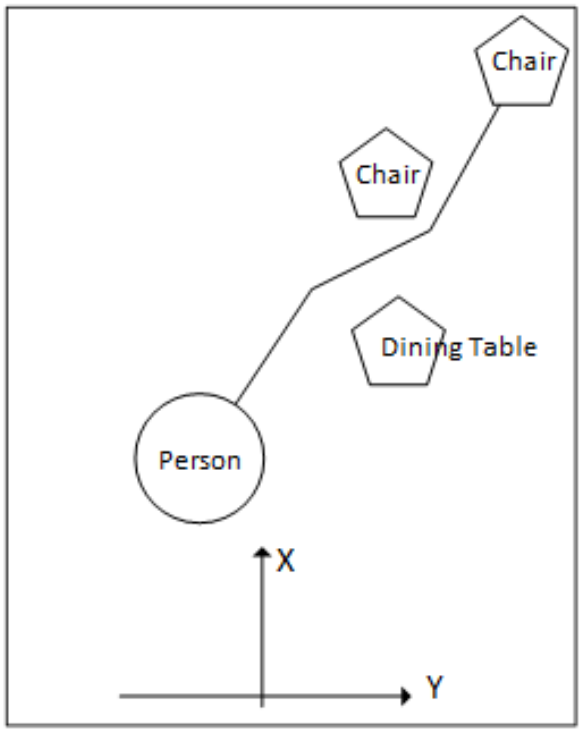

Figure 8. Blocked Path

The results of the test were satisfactory. Objects were clearly discerned, detected and localized to a 2D plane, where the path from the user's current location to the user defined destination was successfully calculated in both cases, clear path and blocked path.

The use of a CNN based object detection technique was advantageous in complex scenes, as proved by the results. Target objects were properly identified in the given complex scene among all other objects such as windows, cabinets, etc. Localization of the detected objects mapped to the visualized 2D plane was accurate. As observed in the given RGB image in Figure 4, the corresponding OpenGL maps (Figure 7 and 8) were able to visualize the scene faithfully. Pathfinding was dynamic and without ambiguity. The system registered the movements of the user and recalculated the path to the destination accordingly, in real-time.

V. Conclusion

In this paper, a novel method of autonomous indoor navigation was introduced combining depth perception and object identification to achieve fast and accurate real-time efficiency. The preliminary testing in a controlled environment displayed encouraging results. Advancements to the system can be made by adding wireless devices for user interface, visual or audio, expanding the system to 3D localization covering multiple rooms and floors, and a token system for helping multiple detected users. The possible applications of this system are vast, ranging from personal navigation to pathfinding for autonomous robots.

References

[1] M. Youssef, M. Elzantout, R. Elkhouly and A. Lotfy, "Ubiquitous indoor localization and worldwide automatic construction of floor plans," arXiv preprint arXiv:1204.3328, 2012.

[2] S. Verma, R. Omanwar, V. Sreejith and G. Meera, "A smartphone based indoor navigation system," in *2016 28th international conference on microelectronics (ICM)*, Giza, Egypt, 2016.

[3] J. Redmon and J. Farhadi, "YOLO9000: Better, Faster, Stronger," arXiv preprint arXiv:1612.08242, 2016.

[4] F. Shayeganfar, A. Anjomshoaa and A. M. Tjoa, "A Smart Indoor Navigation Solution Based on Building Information Model and Google Android," *ICCHP. Springer,* pp. 1050-1056, 2008.

[5] G. Yang, Z. Zhou, T. Gonnot and J. Saniie, "Design Flow of Motion Based," in *2015 IEEE International Conference on Electro/Information Technology (EIT)*, Dekalb, IL, USA, 2015.

[6] T. Gonnot and J. Saniie, "Integrated machine vision and communication system for blind navigation and guidance," in *2016 IEEE International Conference on Electro Information Technology (EIT)*, Grand Forks, ND.

[7] G. Yang and J. Saniie, "Indoor navigation for visually impaired using AR markers," in *2017 IEEE International Conference on Electro Information Technology (EIT)*, Linoln, NE, USA, 2017.

[8] R. Jian, Q. Lin and S. Qu, "Let Blind People See: Real-Time Visual Recognition with Results Converted to 3D Audio," Stanford University, 2016.

[9] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You only look once: Unified, real-time detection," arXiv preprint arXiv:1506.02640, 2015.

[10] S. Magnenat, "Imaging Information," OpenKinect.org, 2010. [Online]. Available: https://openkinect.org/wiki/Imaging_Information.

[11] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona , D. Ramanan, P. Dollr and C. Zitnick, "Microsoft COCO: Common objects in context," in *ECCV*, 2014.