Journal of Research and Applications in Mechanical Engineering
ISSN: 2229-2152 (Print); 2697-424x (Online)
(2024) Vol. 12, No. 1, Paper No. JRAME-24-12-014
[DOI: 10.14456/jrame.2024.14]

*Research Article*

# Semantic Mapping and Voice User Interface Based on ORB-SLAM and YOLO for Navigating Visually Impaired Person

**Q. Wang**[1]
**Y. Shikanai**[2]
**K. Mima**[3]
**K. Tobita**[2,*]
[1] *Department of System Engineering, Graduate School, Shizuoka Institute of Science and Technology, Shizuoka 4378585, Japan*
[2] *Department of Mechanical Engineering, Faculty of Science and Technology, Shizuoka Institute of Science and Technology, Shizuoka 4378585, Japan*
[3] *Department of Electrical and Electronic Engineering, Faculty of Science and Technology, Shizuoka Institute of Science and Technology, Shizuoka 4378585, Japan*

**Received** 26 September 2023
**Revised** 16 November 2023
**Accepted** 25 November 2023

**Abstract:**
*As the world's population grows and life expectancy increases, the number of visually impaired people is increasing. We developed a visual navigation map for visually impaired people to solve their life problems. This map combines the navigation map generated by Visual SLAM with the semantic information of the landmark detected by the YOLO object-detection algorithm to create a map that can be used for voice navigation and other purposes. To help visually impaired people find what they want in their daily lives, we have also developed a voice user interface based on YOLO object detection, which is a relatively lightweight voice recognition system that can help visually impaired people solve problems in their lives.*

**Keywords:** *Visual SLAM, Object detection, Semantic map, Voice user interface*

## 1. Introduction

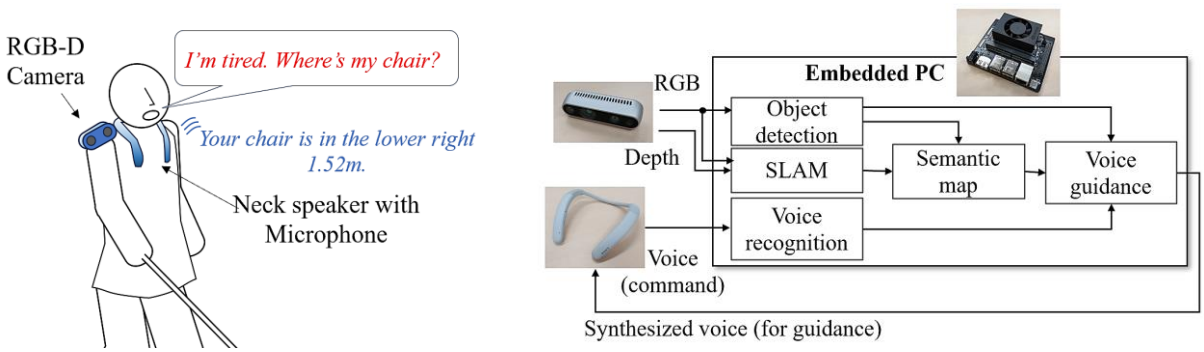
Humans have a significant and valuable ability called visual perception, which gives them access to a wealth of detailed information about their surroundings, including the position and characteristics of various things. The major method for gathering environmental data is visual sensors because of their low cost and superior scene identification capabilities, which have been made possible by advancements in hardware and graphics optimization technologies. As the study of the Visual SLAM algorithm continues to advance, it has yielded numerous beneficial outcomes. Davison's MonoSLAM was the first monocular visual SLAM system [1] that can create continuous sparse maps online based on the extended Kalman filter framework, which opens up new areas in the direction of visual SLAM. However, sparse feature points are easily lost, and the system is not sufficiently stable. The parallel tracking mapping algorithm proposed in [2] is the earliest visual SLAM algorithm based on a graph optimization framework. It divides the system into two threads, mapping and positioning, which improve the execution efficiency of the algorithm and can stably perform attitude estimation and map construction. In [3], an SVO algorithm was proposed, which is faster and has lower computational requirements than other algorithms, but it is only suitable for planar motion. In [4], the SVO algorithm was added to a neural network to predict the depth prediction results of the network through the depth of a single image, which improves the accuracy of SVO. In 2015, [5] proposed the ORB-SLAM algorithm based on the parallel tracking mapping algorithm, which divides the system into three independent threads: feature point tracking, local map construction, and closed-loop detection, thus improving the positioning accuracy.

* Corresponding author: K. Tobita
E-mail address: tobita.kazuteru@sist.ac.jp

In 2017, the literature [6] further improved it and proposed the ORB-SLAM2 algorithm, which added monocular, binocular, and depth camera modes to ensure real-time performance and improve positioning accuracy with a wider application range. In 2017, [7] proposed Semantic Fusion, a semantic map construction method based on convolutional neural networks. This method uses convolutional neural networks and depth maps to generate probability maps and then uses the Bayesian method to combine the recognition results with SLAM to generate dense semantic maps. The indoor effect is good but not suitable for large-scale scenes. CNN-SLAM proposed in the literature [8] used a trained convolutional neural network for the first time to predict the depth map of a single image, simultaneously perform semantic segmentation, and integrate the existing global scene depth map to overcome the limitation of monocular visual SLAM lacking scale information. In 2020, Stanford University proposed a metric-semantic SLAM system, Kimera [9], which uses a camera and inertial navigation to build a semantic 3D environment grid that can achieve accurate state estimation and global trajectory estimation.

In recent years, many high-performance deep learning detection methods have emerged in the field of target detection and recognition. For example YOLO (You Only Look Once), RCNN (Regions with CNN features), SSD (Single Shot MultiBox Detector) and so on. Among them, YOLO algorithm [10] adopts the regression method and uses a convolutional neural network to achieve end-to-end target detection. Compared with the RCNN and other Two-Stage methods that first search for borders and then classify them, the detection speed is faster, and it is suitable for automatic driving and other application scenarios that require very high real-time performance.

This study aimed to create a visual navigation map that provides semantic information to visually impaired people by embedding landmark information detected by the YOLOv5 target detection algorithm into a map generated by visual SLAM (ORB-SLAM). By combining voice systems and object recognition, a voice user interface is created to help visually impaired people find the desired items. The final concept for guiding visually impaired people is shown in Fig. 1. A semantic map is created based on the acquired data of the RGB-D camera. Based on the user's voice commands and object detection results, the system provides the distance and direction to the target object. This paper first describes the principle of semantic mapping and its experiments. Next, the voice user interface and its experiments are described and finally summarized.




(a) Concept (b) System diagram

**Fig. 1.** Conceptual figure used by visually impaired people.

## 2. Building Visual Navigation Maps

### *2.1 Summary of System Configuration*

The system consists of a PC and a depth camera. The PC uses Intel NUC. It is equipped with Core i5-1135G7 core with eight four-core lines and a clock frequency of 4.2GHz. The size is 110 mm x 110 mm and weighs 0.5 kg. An Intel RealSense D435i depth camera (Fig. 2) was used as the sensor. The RealSenseD435i camera is equipped with two infrared sensors, an infrared laser transmitter, a color sensor, and a BMI055 inertia measurement unit, with global image shortcuts and broad-view, which can effectively capture and stream depth data of moving objects. This can provide color, depth, and infrared video streams with six degrees of freedom of movement, with a minimum distance depth of 200 mm and a maximum distance of 10m, with some performance parameters as shown in Table 1.

**Fig. 2.** Intel RealSense D435i.

**Table 1:** Some performance parameters of RealSense D435i.

| Project | Performance parameter |
|---|---|
| Depth range | 0.2m-10m |
| The depth error is less than 2% | 2m |
| Depth image resolution | 1280×720@30fps, 848×480@90fps |
| RGB image resolution | 1920×1080@30fps, 1280×720@90fps |
| Depth field Angle | 86°×57° |
| Color image | 2MP/64°× 41° |
| RGB wide Angle | 69.4° |
| IMU | Support |

Fig. 3 depicts the process of creating a visual navigation map. Based on depth information, a grid map is created, and landmarks are identified using RGB data, projected onto the grid map, and semantic information is added.

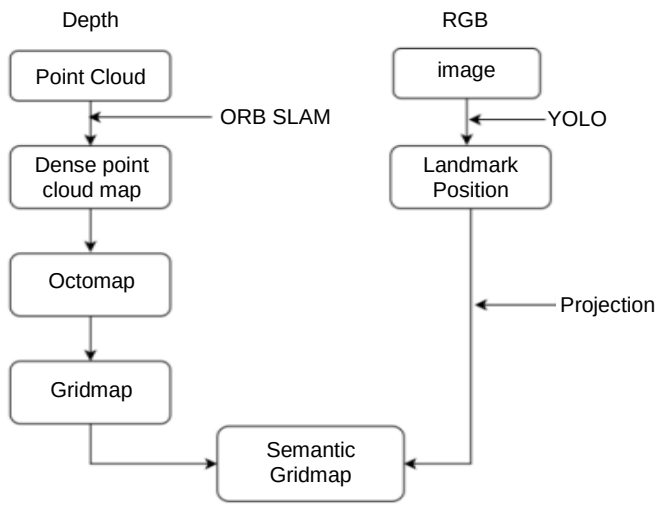



**Fig. 3.** System flowchart.

*2.2 Calibration of Internal and External Parameters of Depth Camera*

The main purpose of camera calibration is to obtain the internal and external parameters of the camera, and to determine the distortion parameters to eliminate the influence of transverse distortion and radial distortion. There are many calibration methods, through discussion, we use Kalibr tool in this study.

First, RealSense D435i was adjusted to the best working state, the camera frequency was reduced to 4Hz, and the calibration board was rotated up and down and left and right to record camera data, with a recording resolution of 640*480, as shown in Fig. 4.

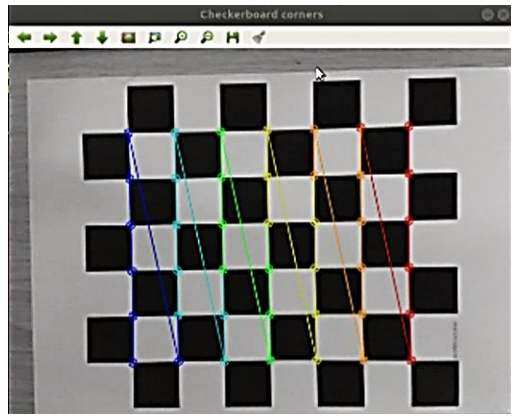



**Fig. 4**. Calibration plate.

Camera data was made as bag packet in ROS system. Kalibr was used to read the data stream of RGB camera and infrared camera from bag packet, and the internal parameters and distortion parameters of RGB camera and infrared camera were calculated. The data analysis of calibration results is shown in Fig. 5.

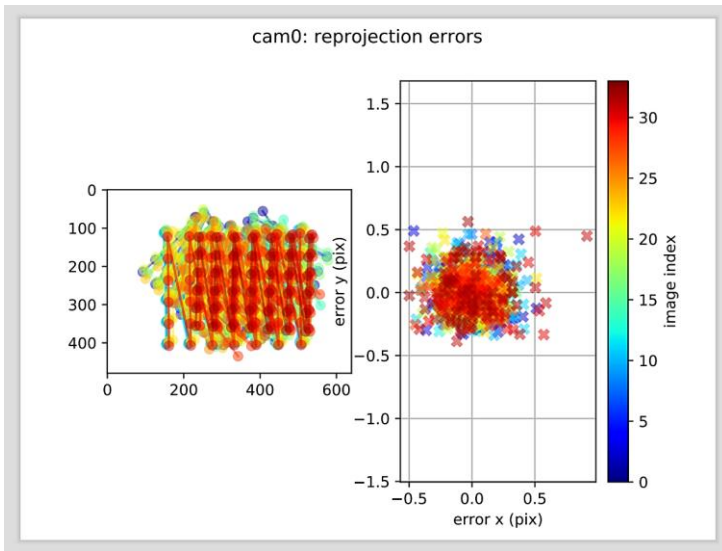



**Fig. 5.** Reprojection error.

From the results of this calibration, the reprojection error is about 0.5 pixels, which can verify that the calibration is successful. The results of calibration are shown in Table 2.

**Table 2:** Calibration results of RGB camera and IF camera

| **Parameter** | **RGB camera** | **Infrared camera** |
|---|---|---|
| Intrinsic parameter | $f_x$ = 665.54<br>$f_y$ = 665.50<br>$C_x$ = 313.22<br>$C_y$ = 247.66 | $f_x$ = 604.34<br>$f_y$ = 605.54<br>$C_x$ = 212.14<br>$C_y$ = 232.43 |
| Distortion parameter | $K_1$ = 0.3452<br>$K_2$ = 1.6657<br>$P_1$ = -8.7483<br>$P_2$ = 13.2314 | $K_1$ = 0.3412<br>$K_2$ = 1.6514<br>$P_1$ = -8.7354<br>$P_2$ = 13.2453 |

According to Table 2, the RGB camera's distortion parameter $k_1$ = 0.3452, $k_2$ = 1.6657, $P_1$ = -8.7483, $P_2$ = 13.2314. The internal parameter matrix is shown in formula 1.

$$\begin{bmatrix} fx & 0 & Cx \\ 0 & fy & Cy \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 665.54 & 0 & 313.22 \\ 0 & 665.50 & 247.66 \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

Infrared camera distortion parameter $k_1$ = 0.3412, $k_2$ = 1.6514, $P_1$ = -8.7354, $P_2$ = 13.2453. The internal parameter matrix is shown in formula 2.

$$\begin{bmatrix} fx & 0 & Cx \\ 0 & fy & Cy \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 604.34 & 0 & 212.14 \\ 0 & 605.54 & 232.43 \\ 0 & 0 & 1 \end{bmatrix} \quad (2)$$

*2.3 Generating Dense Map with ORB-SLAM*

First, we position the camera anywhere in the testing area, and then a map is created using that location as the origin coordinate. The ORB-SLAM2 algorithm is used to extract features from the testing environment, and a sparse point-cloud map is created. Figs. 6 and 7 depict the results of the feature point extraction and point cloud map construction.

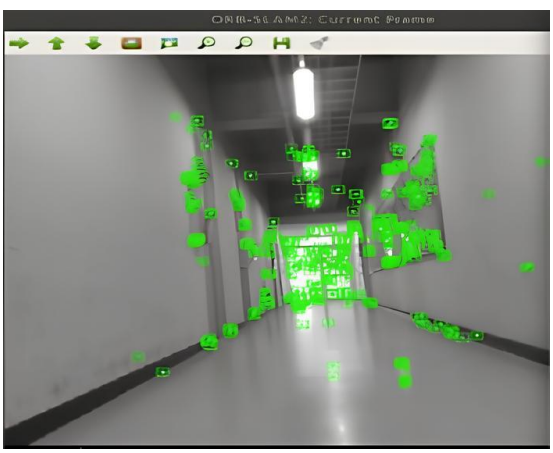


**Fig. 6.** Feature point extraction.

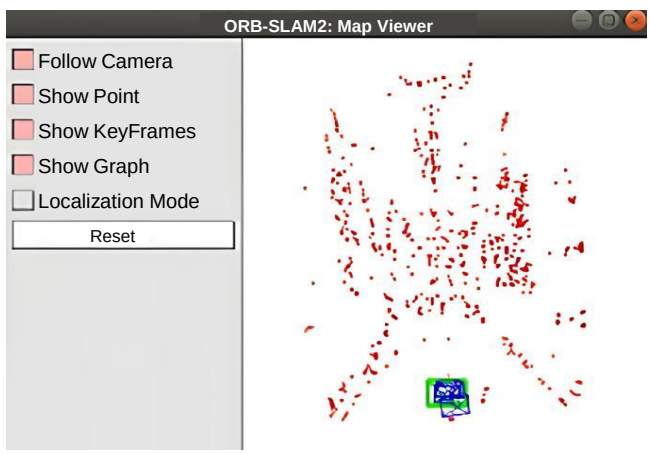



**Fig. 7.** Sparse point cloud map.

Although sparse point-cloud maps can be generated using ORB-SLAM2 to represent the experimental environment, they cannot be used for robot navigation owing to the discontinuous distribution of feature points in space. By connecting each frame of the dense point clouds, a dense point cloud map can be created based on the precise bit estimation results of ORB-SLAM2. Fig. 8 shows the construction outcomes.

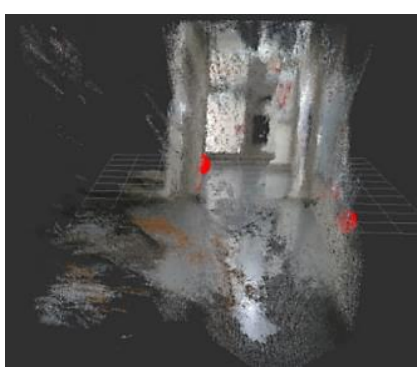


**Fig. 8.** Dense point cloud map.

*2.4 Convert Dense map to Octo map and Grid map*

The experimental environment can be accurately described by a dense point-cloud map created with the upgraded ORB-SLAM2; however, Dense map has a huge storage size and requires considerable storage space. Because Dense map operates in real-time, the amount of storage required increases quickly. Additionally, Dense map contains a large amount of extraneous information that is not required in an experimental setting, such as information on the ceiling. A dense point-cloud map cannot be utilized for navigation or route planning because it lacks information on the relationships between spatial points. Octo map, which is based on Octree [11], is used to modify Dense map. Fig. 9 shows the construction outcomes. The memory footprint and processing speed per frame of Octo map are listed in Table 3. The memory size was significantly smaller than that with Dense map, and the processing time for each frame also decreased.

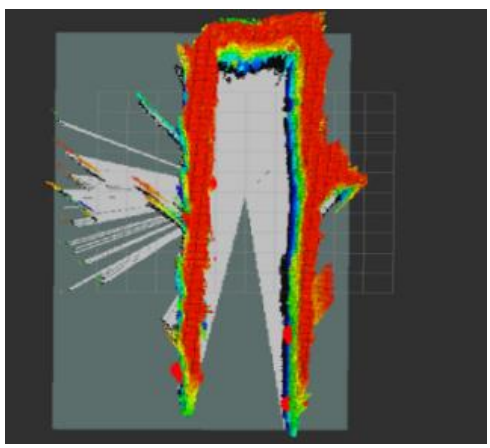


**Fig. 9.** Octree map of experimental environment.

**Table 3:** Point cloud and octree parameter comparison.

| **Parameter** | **Value of number** |
|---|---|
| Point cloud memory | 183.49 MByte |
| Octree memory | 3.06 MByte |
| Average processing time per frame of point cloud | 0.218 s |
| Average processing time per frame of octree | 0.137 s |

Robotic route planning and navigation can be performed using Octo map. However, the calculation is relatively high, and the real-time performance is poor if Octree id directly utilized. The 3D information on a map contains information that is not crucial for navigation, because robots typically navigate and avoid obstacles on a two-dimensional plane. In this study, we use a three-dimensional oblique projection transformation to translate the Octree map acquired above into a two-dimensional Grid map. Fig. 10 depicts the fundamental diagram of oblique projective transformation. Here, point P1 is a voxel's Octo map coordinate, point P2 is a diagonal projection coordinate of Grid map, and point P3 is the voxel's orthogonal projection coordinate.

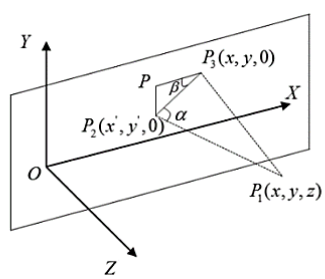



**Fig. 10.** Oblique projection principle.

Equation (1) is used to derive the coordinates projected on the ground, as shown in Fig. 7.

$$x' = x - z\cot\alpha\cos\beta$$
$$y' = z - z\cot\alpha\sin\beta \quad (3)$$

$\alpha$ is the angle between $P_1$, $P_2$, and $P_3$ and $\beta$ is the angle between $P_1$, $P_2$, and $P_3$.

Based on Eq. (1), Octo map uses oblique projective transformation to project a voxel that is between 0.1 and 2.0 meters above the ground onto a two-dimensional plane. By continuously updating the occupied state of the lattice created during the oblique projection, a Grid map of the experimental environment is created. Fig.11 depicts the impact of the construction.

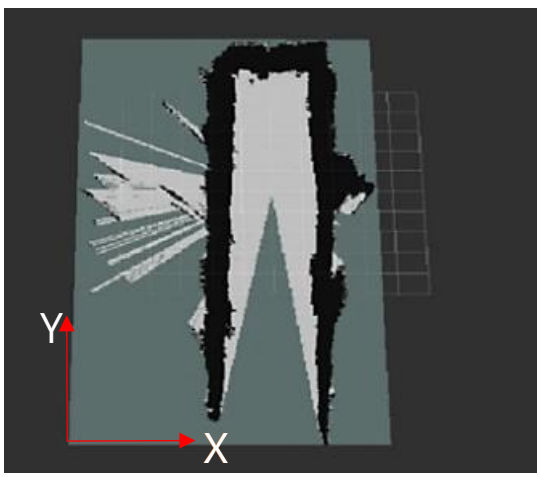



**Fig. 11.** Grid map of experimental environment.

*2.5 Landmark Detection with YOLO*

Training and test sets must be generated to recognize items according to the actual experimental environment before training and testing the YOLOv5s network. We turned on the depth camera RealSense D435i in the hallway and used the rosbag program to record the test environment dataset. The parameters of the dataset are listed in Table 4.

**Table 4:** Experimental environment data set division.

| **Parameter** | **Numerical value (sheet)** |
|---|---|
| Total data sets | 120 |
| Training set | 100 |
| Test set | 20 |

A dataset must be labeled once it has been created. Labeling can be performed in a variety of ways; however, in this study, we utilized Roboflow, a program that is suggested on the official website. We labeled the dataset of the experimental setting for doors and fire extinguishers using labeling tools, as illustrated in Fig. 12. Images from the test set were used to test the weight files saved during each training session, and it was determined that each landmark was recognized, as shown in Fig. 13. Fig. 14 presents the outcomes of the training, after discussion, the model meets the experimental requirements.

**Fig. 12.** Data set annotation.

**Fig. 13.** Recognition effect of road sign.

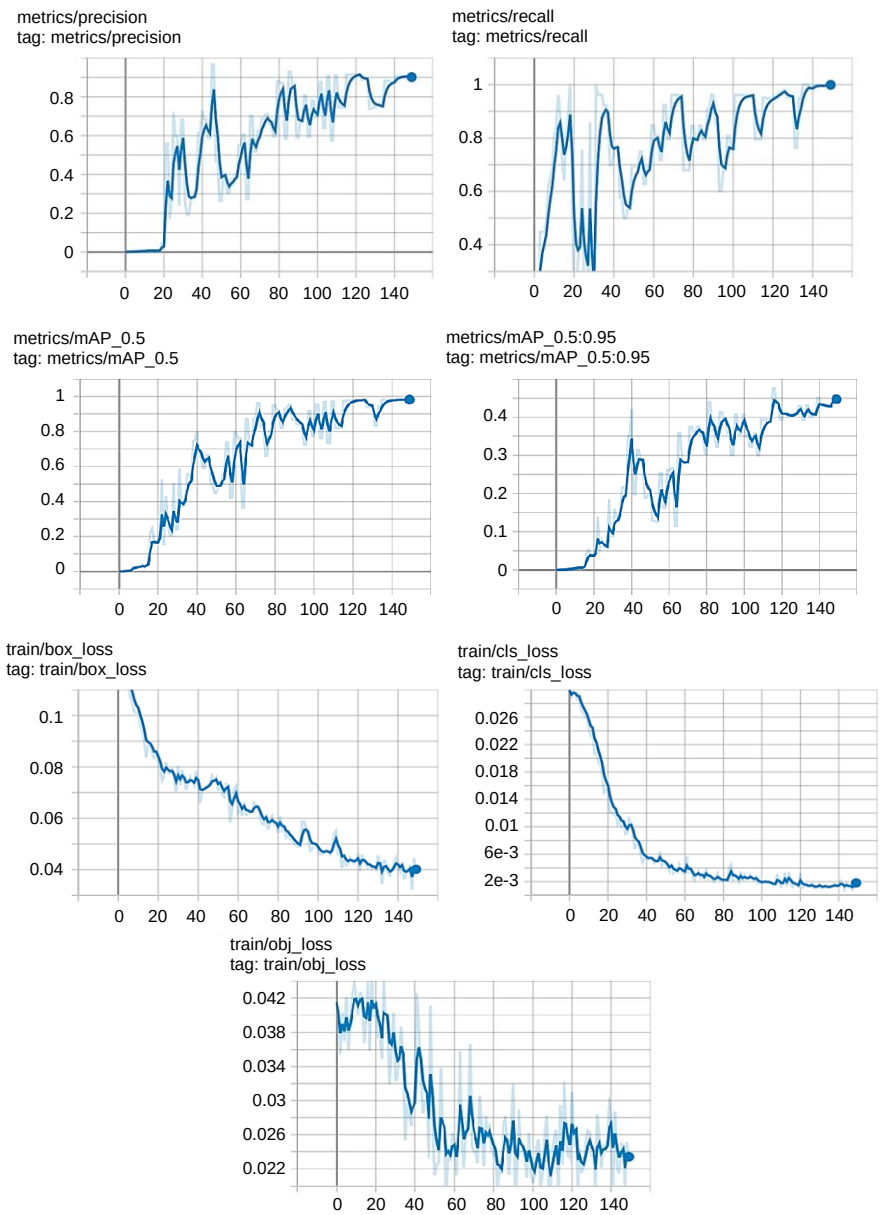



**Fig. 14.** Model training results.

## 3. Principles and Experiments of Semantic Mapping

### *3.1 Principle of Semantic Mapping*

To obtain and recognize landmarks in the test environment, the semantic mapping system concurrently employs ORB-SLAM2 to build a map and activates the YOLOv5 object identification thread. In this work, we annotate the terms “door” and “fire extinguisher” as the landmarks. The center of the landmark is formed at the point where the diagonal lines connecting its four vertices meet. The center point $P(x, y)$ represents the landmark’s central location, whereas points $P_1$ ($x_1$, $y_1$), $P_2$ ($x_1$, $y_2$), $P_3$ ($x_2$, $y_1$), and $P_4$ ($x_2$, $y_2$) are the coordinates of the landmark’s four pixel vertices. Equation (4) provides the calculation formula.

$$x = \frac{x_1 + x_2}{2}$$
$$y = \frac{y_1 + y_2}{2} \tag{4}$$

When the object recognition thread spots a specific landmark, it is chosen in the identification frame and uses Eq. (2) to determine the coordinates of the landmark's center point. The location of the landmark's central coordinates in the coordinate system of the camera can be determined according to the internal parameters. Using the external characteristics of the camera, it is also possible to determine the locations of the landmarks in the global coordinate system. Visual SLAM and object identification are combined, allowing for superimposition of the location of the recognized landmark on the map in the world coordinate system and its semantics. Fig. 15 displays item recognition in the test setting. The experimental parameters are listed in Table 5. Fig. 16 displays the results of semantic mapping. The door was marked with red signposts, and the fire extinguisher was marked with green signs.


**Fig. 15.** Experimental road sign recognition effect.

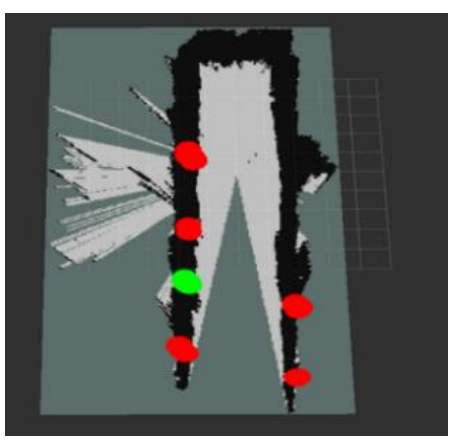


**Fig. 16.** Road sign map diagram.

**Table 5:** Parameters of the experimental.

| Parameter | Numerical value |
|---|---|
| Length of corridor | 10m |
| Corridor width | 1.5m |
| The movement speed when generating the map | 0.4m/s |
| Camera height | 0.15m |
| Average processing time per frame of semantic gridmap | 0.135s |

*3.2 The Experiment on the Error of Semantic Mapping*

To verify the accuracy of the position of the landmark, the position of the landmark in the two-dimensional grid was calculated through the internal and external parameter matrix of the camera, and the actual position of the landmark was compared; the grid size was 10x10; we set the lower left corner of the grid as the origin of the coordinates, and set the distance of each grid to 1 m. The experiment was carried out by changing the positions of the landmark ten times. Fig. 17 shows the detected and actual landmark coordinates' relative coordinates as errors. The X-axis and Y-axis represent the error in the horizontal and vertical directions, respectively. When measuring the deviation from the real coordinates in terms of the distance from the origin, the average and standard deviation were 0.154 m and 0.0165 m, respectively. The red brake line indicates the average of these distance. The results show that the semantic mapping error is about one-tenth to one-fifth of the grid, which verifies the accuracy and feasibility of the proposed mapping method.

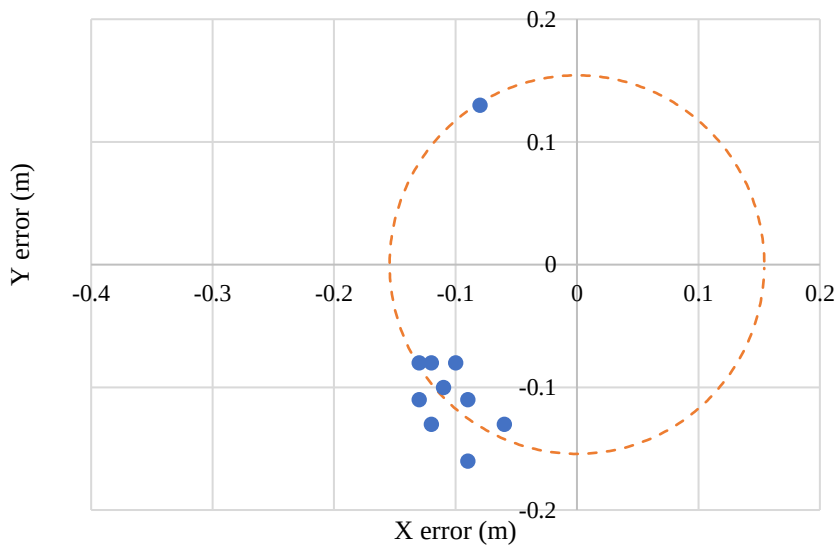



**Fig. 17.** Error display of semantic mapping.

## 4. Building Voice User Interface

*4.1 Summary of System Configuration*

The primary sensor is also the D435i depth camera, the voice device we chose was sony SRS-NB10 neck Bluetooth speaker. As shown in Fig. 18, the RGB information from the camera is sent to the YOLOv5 object identification system to identify the category of the object, after which the system uses the depth data input to determine the three-dimensional coordinates of the object. The object category's position is then communicated through speech. In addition, for the ease of use for visually impaired individuals, we have developed voice recognition features that permit users to locate particular objects and receive information on their whereabouts through analyzing their instructions.

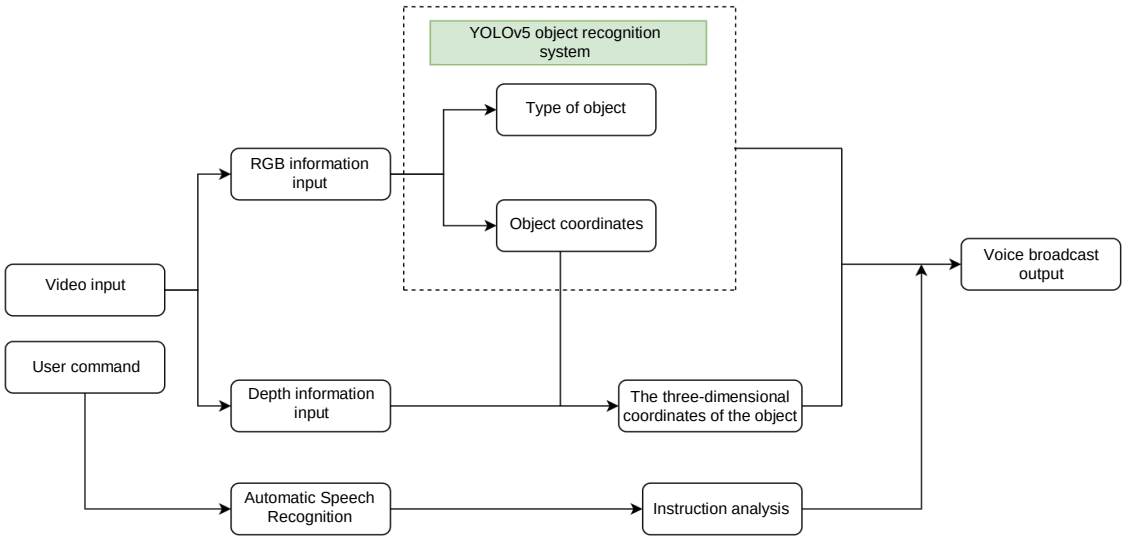



**Fig. 18.** System flowchart.

*4.2 Determine the Type of Object to Detect and Model Training*

Before constructing the YOLO detection model, it is essential to determine the types of items that will be recognized. We have selected eleven distinctive objects as detection targets for the YOLO model, which are frequently used by visually impaired individuals. These items include "person, sink, microwave, cell phone, couch, chair, oven, bed, hair dryer, door, and window". To enhance identification accuracy, we extensively searched for appropriate data on the coco data set website and subsequently processed it. Ultimately, we trained the model based on the data parameters illustrated in Table 6.

**Table 6:** Data set division.

| **Parameter** | **Numerical value (sheet)** |
|---|---|
| Total data sets | 69197 |
| Training set | 61483 |
| Test set | 7714 |

Fig. 19 presents the outcomes of the training. After each completion of YOLO's training, an expX directory (where X denotes the number of results generated) is produced under the run directory. The results include Box, Objectness, Classification, Precision, Recall, and mAP parameters. YOLOV5 adopts GIOU loss as the bounding box loss function, with higher accuracy of the box corresponding to lower mean GIoU loss values. Objectness's speculation is based on the average of the target detection. The lower the average, the greater the accuracy of the objective detection. Prefix "val" refers to the validation set. A validation set refers to a limited amount of data that is set apart from the training data set to assess a model's performance on new data. The model undergoes parameter updating and optimization in line with the training set, and its performance is subsequently assessed using the validation set to fine-tune hyperparameters or choose the most fitting model. By examining the assessment outcomes of the dataset, we can assess if the model is overfitting or underfitting and apply appropriate modifications. Precision is the level of accuracy (all positive classes have been identified) while recall is the accuracy of identifying positive instances (how many positive examples were found). The recall rates of various categories will be computed to determine a comprehensive score. Accordingly, the model's accuracy is deemed satisfactory for the research requirement after careful deliberation.

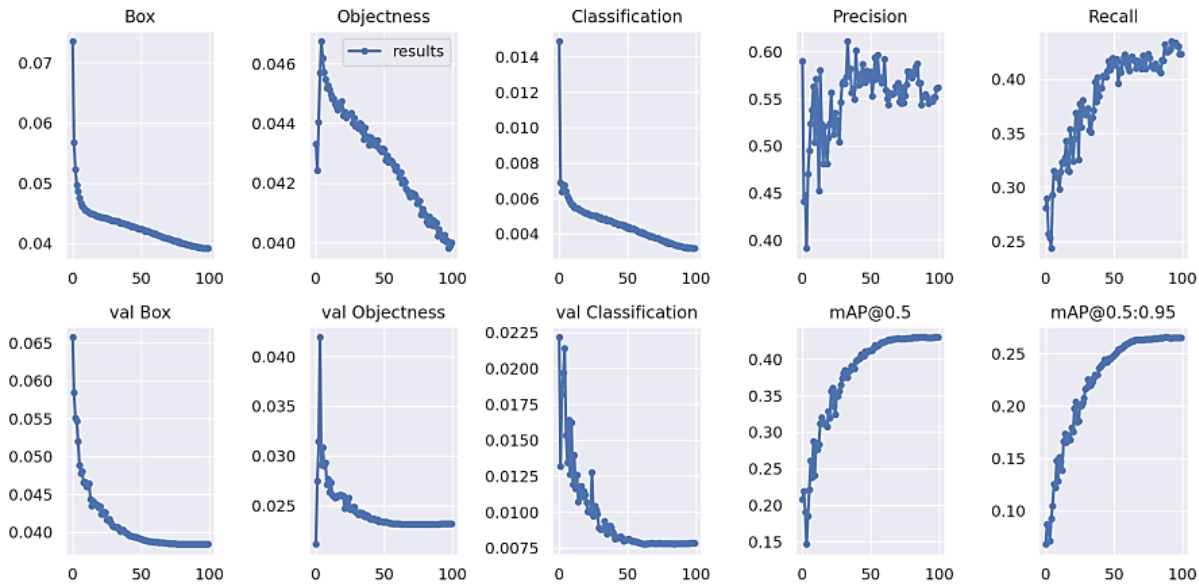



**Fig. 19.** Model training results.

Since the focus of this study is on individuals with visual impairments, it is necessary to provide information on the depth and position of objects. The image is segmented into four regions, as depicted in Fig. 20, with designations of "upper left," "lower left," "upper right," and "lower right."
The programme's effectiveness is presented in Fig. 21.

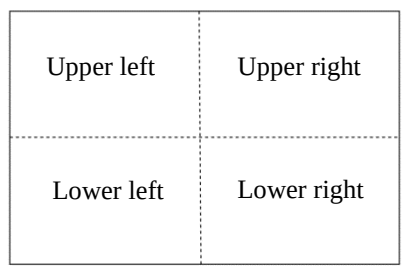



**Fig. 20.** Zoning diagram.


**Fig. 21.** Detection effect.

*4.3 Automatic Speech Recognition*

Speech recognition technology is capable of converting speech data into text [12]. Thanks to the continuous improvement of artificial intelligence, voice recognition technology has become highly advanced over time. The principle of speech recognition is shown in Fig. 22.

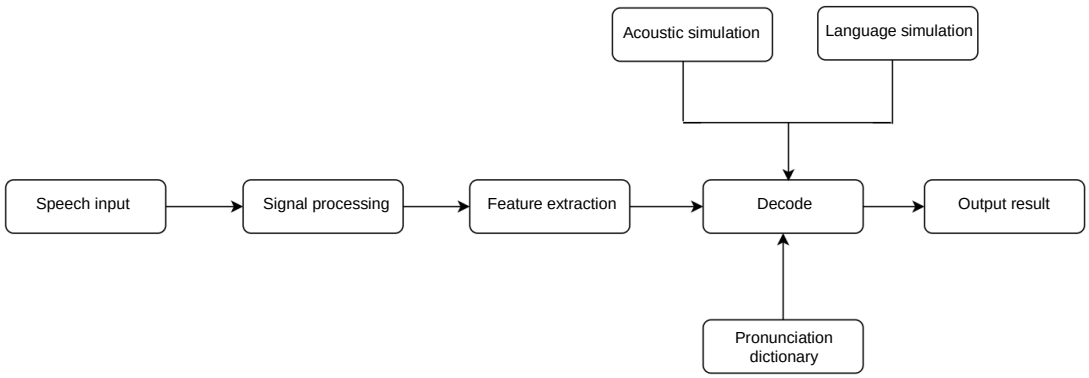



**Fig. 22.** The principle of Speech Recognition.

*4.4 Voice Input Control*

As object recognition and voice broadcasts occur in real-time and may be affected by the surrounding environment, the keyboard spaces were chosen to control the input and termination of voice. Additionally, each speech input and output is displayed in the terminal, as shown in Fig. 23.

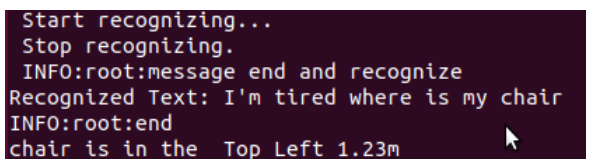



**Fig. 23.** Speech input and output results.

*4.5 Experiments on Detection Accuracy*

To determine the accuracy of the system, experiments were conducted to measure the deviation between the actual object position and the position detected by YOLO. A 3-metre long road was created using yellow tape, with markers placed every 0.1m. Fig. 24 displays the experimental environment. For testing, a mobile phone was selected as the object due to its regular shape and small size. Then, we position the camera at the beginning of the road and, as the D435i camera has a minimum detection depth of 0.2m, we move the phone from 0.3m to 3m, increasing the distance by 10cm each time. The camera, phone, and road remained aligned in a straight line, and the camera's angle is shown in Fig. 25.

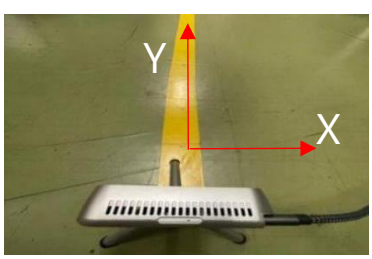



**Fig. 24.** Experimental environment.


**Fig. 25.** Camera angle.

*4.6 Analysis of Experimental Results*

Finally, we recorded the experimental results and calculated the detection error. The relationship between detection distance and detection error can be seen in Fig. 26.

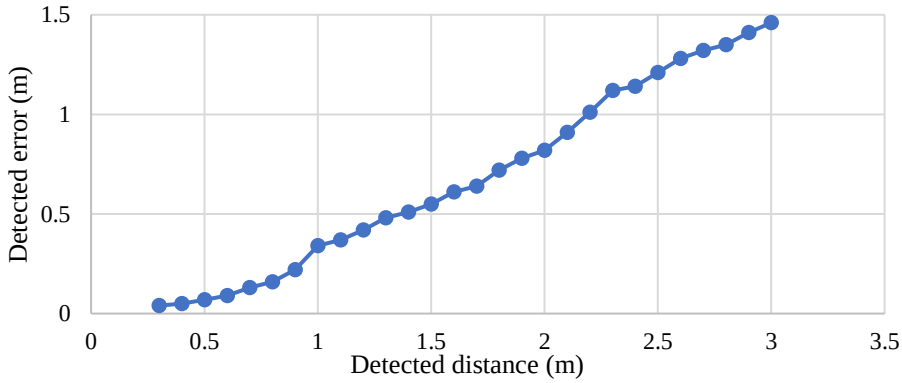



**Fig. 26.** The relationship between detection distance and detection error.

Through the experimental results, it is evident that the error increases as the detection distance increases. This is due to the selection box continuously jumping during real-time detection. The experimental results were obtained by taking the average value, which also contributes to the increase in error. However, in practical application, if the object is at a considerable distance from the user, it may be beneficial to guide the user to a general location, which could result in an accuracy variation of one to two meters. It is my contention that the precise location of the object can then be determined once the approximate position has been established again.

## 5. Conclusion

By integrating environmental landmarks detected by the object detection algorithm YOLOv5 into a map created using Visual SLAM (ORB-SLAM), we built a semantic navigation map. We suggested a semantic mapping technique and displayed the outcomes of our experiments. This approach addresses the lack of meaning in the traditional grid map and lowers the hardware cost of the SLAM technology. In applications, it is also possible to send landmark information to the user in conjunction with speech nodes and enhance the precision of navigation as an assist function. I believe that adding this map to guide robots will improve navigation accuracy and enhance human-computer interaction. In addition, we proposed a lightweight system suitable for daily use by visually impaired people, which adds depth information to traditional YOLO and an integrated speech recognition system, and verified the feasibility of this system through experiments. However, there are still some shortcomings in this study, and we will continue to improve these in future work.

## References

[1] Davison AJ, Reid ID, Molton ND, Stasse O. MonoSLAM: real-time single camera SLAM. IEEE Trans Pattern Anal Mach Intell. 2007;29(6):1052-1067.

[2] Klein G, Murray D. Parallel tracking and mapping for small AR workspaces. 2007 6th IEEE and ACM International Symposium on Mixed and Augmented Reality; 2007 Nov 13-16; Nara, Japan. USA: IEEE; 2007. p. 225-234.

[3] Forster C, Zhang Z, Gassner M, Werlberger M, Scaramuzza D. SVO: Semidirect visual odometry for monocular and multicamera systems. IEEE Trans Robot. 2017;33(2):249-265.

[4] Loo SY, Amiri AJ, Mashohor S, Tang SH, Zhang H. CNN-SVO: improving the mapping in semi-direct visual odometry using single-image depth prediction. 2019 International Conference on Robotics and Automation (ICRA); 2019 May 20-24; Montreal, Canada. USA: IEEE; 2019. p. 5218-5223.

[5] Mur-Artal R, Montiel JMM, Tardos JD. ORB-SLAM: a versatile and accurate monocular SLAM system. IEEE Trans Robot. 2015;31(5):1147-1163.

[6] Mur-Artal R, Tardos JD. ORB-SLAM2: an open-source SLAM system for monocular, stereo, and RGB-D cameras. IEEE Trans Robot. 2017;33(5):1255-1262.
[7] McCormac J, Handa A, Davison A, Leutenegger S. SemanticFusion: Dense 3D semantic mapping with convolutional neural networks. 2017 IEEE International Conference on Robotics and Automation (ICRA); 2017 May 29 - Jun 3; Singapore. USA: IEEE; 2017. p. 4628-4635.
[8] Tateno K, Tombari F, Laina I, et al. CNN-SLAM: real-time dense monocular SLAM with learned depth prediction. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2017 Jul 21-26; Honolulu, USA. USA: IEEE; 2017. p. 6565-6574.
[9] Rosinol A, Abate M, Chang Y, Carlone L. Kimera: an open-source library for real-time metric-semantic localization and mapping. 2020 IEEE International Conference on Robotics and Automation (ICRA); 2020 May 31 - Aug 31; Paris, France. USA: IEEE; 2020. p. 1689-1696.
[10] Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: unified, real-time object detection. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR); 2016 Jun 27-30; Las Vegas, USA. USA: IEEE; 2016. p. 779-788.
[11] Hornung A, Wurm KM, Bennewitz M, Stachniss C, Burgard W. OctoMap: an efficient probabilistic 3D mapping framework based on octrees. Auton Robot. 2013;34(3):189-206.
[12] Oruh J, Viriri S, Adegun A. Long short-term memory recurrent neural network for automatic speech recognition. IEEE Access. 2022;10:30069-30079.