Research and Simulation of A Home Robot Navigation System Based on YOLO-SBA and ROS2

Liang Zhou

College of Information Science and Technology, Northwest University, Xi'an, China
2022117530@stumail.nwu.edu.cn

Abstract. With the rapid advancement of smart homes and assistive care technologies, the application scope of domestic service robots in daily life has been continuously expanding. To ensure these robots can efficiently execute tasks and adapt to complex environments, their autonomous navigation capabilities have become a critical challenge. In home environments, robots must navigate dynamic obstacles, narrow passages, and variable environmental factors, which impose higher demands on their path planning and real-time decision-making abilities. To address these challenges, this study proposes an efficient navigation system for domestic service robots, integrating the YOLO-SBA object detection algorithm with the ROS2 robotic operating system. This research employs the ROS2 operating system for robot modeling and simulation, utilizing the COCO128 dataset for model training. Compared to the baseline YOLOv8n model, the YOLO-SBA model achieves a 2.1% increase in precision, a 10.2% improvement in recall, a 9.9% enhancement in mAP50, and a 9.3% boost in mAP50:95. The system's effectiveness was validated through laser SLAM mapping experiments conducted on the Gazebo simulation platform. The results demonstrate that the proposed system can efficiently perform object detection and SLAM mapping tasks in complex home environments.

Keywords: Domestic Service Robots; YOLO-SBA; ROS2; COCO128 Dataset; Laser SLAM Mapping

1 Introduction

With the rapid development of smart homes and assistive care technologies, domestic service robots have gradually become an indispensable part of modern household life [1]. These robots are not only capable of performing daily household tasks, such as cleaning and object transportation, but also provide assistive care services for the elderly or individuals with limited mobility, significantly enhancing convenience and safety in daily life. However, the complexity and diversity of home environments pose significant challenges to the autonomous navigation capabilities of robots. Dynamic obstacles (e.g., moving pets or furniture), narrow passages (e.g., doorways or corridors), and variable environmental factors (e.g., lighting changes or temporarily placed objects) require robots to possess efficient real-time perception, environmental mapping, and path-planning capabilities.

J. Iqbal (ed.), *Proceedings of the 2025 2nd International Conference on Mechanics, Electronics Engineering and Automation (ICMEEA 2025)*, Advances in Engineering Research 272,
https://doi.org/10.2991/978-94-6463-821-9_48

Autonomous navigation technology is the core of enabling domestic service robots to achieve efficient task execution and environmental adaptability [2]. Traditional navigation methods perform well in structured environments but often fail to meet the requirements of real-time performance and robustness in complex and dynamic home environments. Therefore, developing a navigation system capable of stable operation in dynamic and uncertain environments has become a critical research direction. Object detection [3] and Simultaneous Localization and Mapping (SLAM) technology [4], as key components of robot perception and navigation, directly determine the adaptability of robots in complex environments. In recent years, significant progress has been made in related fields. For example, Cai et al. proposed a deep learning-based multi-sensor fusion SLAM method, significantly improving the localization accuracy of robots in dynamic environments [5]; Wang et al. achieved efficient object detection in complex scenes by enhancing the YOLOv5 model [6]; additionally, Cheng et al. developed an adaptive path-planning algorithm by integrating ROS2 and reinforcement learning, further improving the navigation efficiency of robots in unstructured environments [7]. These studies provide important theoretical and technical support for the development of domestic service robots.

To address the aforementioned challenges, this study proposes an efficient navigation system for domestic service robots based on the YOLO-SBA object detection algorithm and the ROS2 robotic operating system. The YOLO-SBA model, through optimization and improvement of YOLOv8n, significantly enhances the precision and recall of object detection, providing reliable support for real-time perception in complex environments. Furthermore, by leveraging the efficient communication and scheduling capabilities of the ROS2 operating system and the laser SLAM mapping validation on the Gazebo simulation platform, this study successfully achieves efficient navigation and task execution in complex home environments.

This study constructs an efficient navigation system for domestic service robots based on YOLO-SBA and the ROS2 operating system, and validates its performance through simulation experiments. The main contributions of this paper are as follows:

(1) Proposing an improved YOLOv8 object detection algorithm—YOLO-SBA—to enhance the robot's perception capabilities in dynamic environments;

(2) Implementing multi-sensor fusion based on the ROS2 framework and constructing a robot simulation model;

(3) Utilizing the Gazebo simulation platform for system mapping, testing, and optimization, verifying its SLAM mapping performance in complex home environments.

2 Preparatory Work

This section primarily introduces the preparatory work required for this study, focusing on the establishment of the robot model and the construction of the simulation environment. Firstly, the core functionalities of the ROS2 operating system and its advantages in robot modeling and simulation are elaborated in detail. As an open-source robotic operating system, ROS2 provides efficient communication mechanisms and modular design, effectively supporting perception, decision-making, and control tasks in robotics. Secondly, the COCO128 dataset used for model training in this study is introduced. As a subset of the COCO dataset, COCO128 contains a wide range of object categories and scenarios, offering diverse sample support for model training and ensuring the model's generalization capability in complex home environments. Through in-depth analysis and application of the ROS2 operating system and the COCO128 dataset, a solid foundation is laid for subsequent model optimization and experimental validation.

2.1 Robot Operating System 2

Robot Operating System 2 (ROS 2) is an open-source software development framework designed to provide modular, scalable, and cross-platform support for the design, development, and control of robotic systems [8]. As the next-generation version of ROS, ROS 2 has undergone significant improvements in architecture, performance, and functionality to address the complex demands of modern robotic applications. ROS 2 adopts a distributed computing architecture, enabling collaborative operations in multi-robot systems. Its core design principles include modularity, loose coupling, and scalability. Compared to ROS 1, ROS 2 utilizes the Data Distribution Service (DDS) for communication, an industry-standard middleware that offers efficient and reliable message delivery mechanisms. DDS supports real-time performance and Quality of Service (QoS) configuration, making ROS 2's communication mechanisms more flexible and adaptable to diverse network environments and hardware platforms.

The core components of ROS 2 include Nodes, Topics, Services, Actions, and the Parameter Server. Nodes are the fundamental computational units in ROS 2, each capable of independently executing tasks and communicating with other nodes via Topics, Services, or Actions. Topics facilitate message passing through a publish-subscribe model, while Services support request-response interactions. Actions provide state tracking and feedback mechanisms for complex, long-running tasks. Additionally, ROS 2 supports multiple transport protocols, such as TCP, UDP, and shared memory, to accommodate different network environments and performance requirements.

ROS 2 offers extensive cross-platform support, enabling it to run on various operating systems, including Linux, Windows, and macOS, as well as embedded platforms like RTOS. This versatility allows ROS 2 to be widely applied in scenarios ranging from desktop simulations to real-world robot deployments. Furthermore, ROS 2 supports multiple programming languages, such as C++, Python, and Java, enhancing its cross-platform capabilities. In its design, ROS 2 places particular emphasis on real-time performance and security. By integrating real-time operating systems (RTOS) and supporting real-time scheduling strategies, ROS 2 meets the demands of time-sensitive robotic applications. Additionally, ROS 2 provides secure communication mechanisms, including encryption and authentication, to ensure the safety of robotic systems in complex network environments.

ROS 2 also offers a rich suite of development tools, including simulation tools (e.g., Gazebo), visualization tools (e.g., RViz), and debugging tools (e.g., rqt), which significantly simplify the development, testing, and debugging processes of robotic systems. Moreover, ROS 2 benefits from a large open-source community and ecosystem, providing a vast array of software packages and libraries that cover various aspects, from perception and navigation to control. ROS 2 has been widely adopted in academic research, industrial automation, and service robotics. Its flexibility and scalability make it suitable for a wide range of applications, from simple mobile robots to complex multi-robot systems, demonstrating significant potential in fields such as autonomous driving, drones, smart manufacturing, and medical robotics [9].

2.2 Dataset

To address the challenge of recognizing diverse objects in home navigation scenarios, this paper employs the COCO128 dataset [10] as the experimental dataset, with sample images illustrated in Fig 1. The COCO128 dataset is a simplified version of the COCOtrain2017 dataset, consisting of the first 128 images and encompassing 80 object categories. These categories include common household items such as furniture, electronic devices, kitchen utensils, and vehicles, which are highly relevant to objects typically found in home environments. This makes the dataset particularly suitable for supporting object recognition tasks in domestic navigation scenarios. The streamlined nature of the COCO128 dataset ensures a balance between category diversity and representativeness while significantly reducing the computational costs associated with data processing and model training. As a result, it serves as an ideal choice for rapid prototyping and small-scale experiments. Inheriting the extensive category coverage and high-quality annotations of the original COCO dataset, the COCO128 dataset also minimizes computational resource consumption by reducing the data volume, thereby providing an efficient experimental platform for algorithm development and model optimization.

Fig. 1. Sample Images from the COCO128 Dataset

2.3 Selective Boundary Attention Module

The Selective Boundary Attention (SBA) module [11] serves as the core component of YOLO-SBA, representing an innovative attention mechanism designed to enhance feature fusion capabilities. The SBA module focuses on the efficient integration of boundary information and semantic information within feature maps. By adaptively calibrating multi-scale features, it significantly improves the model's sensitivity to object boundaries, thereby enhancing both target localization and classification accuracy. Compared to traditional upsampling and cropping modules, the SBA module demonstrates superior performance in fine-grained object detection tasks within complex scenes. Particularly in tasks requiring the recognition of diverse objects and precise localization, the SBA module exhibits exceptional performance due to its ability to reinforce boundary information. In the context of home navigation tasks, the introduction of the SBA module significantly optimizes the model's ability to recognize various objects in domestic environments. Whether handling dynamic targets, small objects, or complex occlusion scenarios, the SBA module demonstrates remarkable adaptability and precision.

The mathematical expressions of SBA module can be written as follows:

Given an input feature map $F \in R^{C\times H\times W}$, the information is extracted from the input feature map using an edge detection operation as shown in equation (1).

$$F_{edge} = Edge(F) \tag{1}$$

where F_{edge} represents the boundary features.

Attention weights for boundary and semantic features are computed using 1 x 1 convolutional layers followed by a sigmoid activation function as shown in equation (2).

$$\begin{cases} W_{edge} = \sigma\left(Conv_{1\times 1}\left(F_{edge}\right)\right) \\ W_{semantic} = \sigma\left(Conv_{1\times 1}(F)\right) \end{cases} \tag{2}$$

where W_{edge} and $W_{semantic}$ are the boundary and semantic weights, respectively, and σ(·) denotes the sigmoid function.

The calibrated feature map is obtained by fusing the weighted boundary and semantic features as shown in equation (3).

$$F_{calibrated} = W_{edge} \odot F_{edge} + W_{semantic} \odot F \tag{3}$$

where ⊙ represents elements-wise multiplication.

The calibrated features from multiple scales(P2,P3,P4,P5) are fused into the final output feature map as shown in equation (4).

$$F_{output} = Fuse\left(F^{P2}_{calibrated}, F^{P3}_{calibrated}, F^{P4}_{calibrated}, F^{P5}_{calibrated}\right) \tag{4}$$

where $Fuse(.)$ denotes the multi-scale fusion operation.

3 Visual recognition algorithm

YOLOv8 was selected as the baseline model for this study due to its robust object recognition capabilities, which demonstrate higher accuracy and more stable recognition performance in navigation and object detection tasks within home environments.

The overall architecture of YOLO-SBA is illustrated in Fig 2. Compared to the original YOLOv8, this model optimizes the Feature Pyramid Network (FPN) by replacing the traditional upsampling and cropping modules with the Selective Boundary

Attention (SBA) module. This module excels in multi-scale feature fusion, significantly enhancing the detection capability for targets in complex scenes, particularly in small object recognition and precise boundary localization. By integrating YOLOv8n with the SBA module, the proposed model achieves more accurate target recognition in home navigation tasks, improves environmental perception, and maintains stable detection performance under varying lighting conditions and occlusions. These advancements collectively elevate the overall intelligence level of the system.

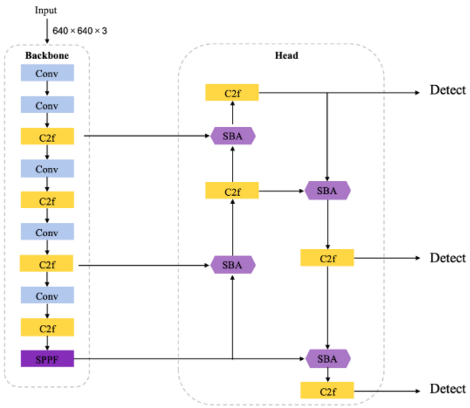

Fig. 2. The structure of YOLO-SBA (Picture credit: Original)

4 Experiment

4.1 Experimental Environment

To compare the recognition performance of various network models on the self-collected dataset, it is necessary to establish a corresponding training environment for model training and validation. The training environment constructed in this study, along with the associated training hyperparameter settings, are detailed in Table 1 and Table 2, respectively.

Table 1. Hardware and Software Experimental Environment Configuration

		Name	Software Experimental Environment
Hardware Environment	Experimental	GPU	NVIDIA GeForce RTX4090D
		CPU	18 vCPU AMD EPYC
		Memory	60GB
		Graphics Memory	30GB
Software Environment	Experimental	Operation System	Ubuntu20.04

Python	3.8
Pytorch	2.0.0
CUDA	11.8

Table 2. Hyperparameter setting

Parameter name	Hyperparameter setting information
Image Size	640 × 640 × 3
Epochs	500
Batch Size	128
Optimizer	AdamW
Lr0	0.001
Lrf	0.01
Momentum	0.937
Weight Decay	0.0005
Learning strategy	Cosine Annealing

4.2 Result of Experiment

The training results of YOLO-SBA are illustrated in Fig 3. It can be observed that around 500 epochs, the box loss, cls loss, and dfl loss exhibit convergence behavior, while precision, recall, mAP50, and mAP50-95 reach their peak values during training. The corresponding recognition performance is demonstrated in Fig 4.

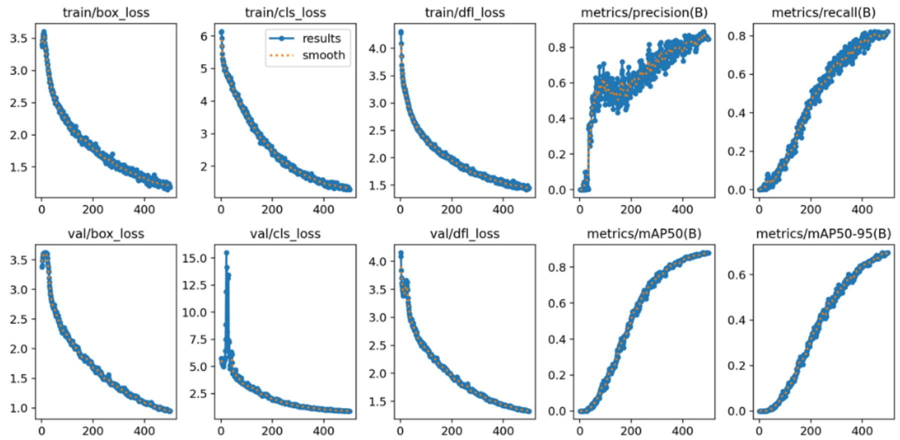

Fig. 3. Training Results of YOLO-SBA (Picture credit: Original)

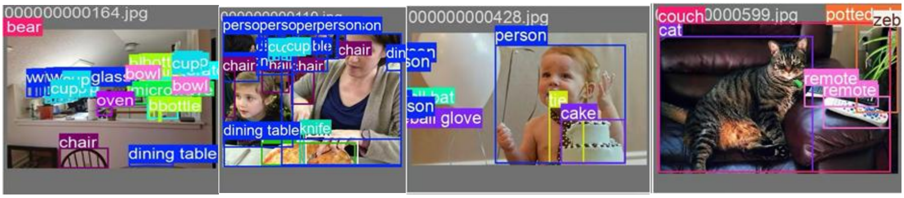

Fig. 4. Recognition Performance of YOLO-SBA (Picture credit: Original)

Furthermore, to validate the performance of the improved detection algorithm, a comparative experiment was conducted between the YOLO-S network model and the YOLOv8n object detection algorithm. The specific experimental results are presented in Table 3. The data in Table 3 indicate that, overall, the YOLO-S model demonstrates improvements across all metrics compared to YOLOv8n and other YOLO models. Specifically, compared to the baseline model YOLOv8n, the YOLO-S model achieves a 2.1% increase in precision, a 10.2% improvement in recall, a 9.9% enhancement in mAP50, and a 9.3% boost in mAP50:95.

Table 3. Comparison Experiments For YOLO-SBA

Models	Precision	Recall	mAP50	mAP50:95
YOLO-SBA	88.2%	80.1%	88%	69.7%
YOLOv8n	86.1%	69.9%	78.1%	60.4%

5 Simulation Experiment

In this section, a robot model was first created using the Xacro format [12]. Subsequently, the Gazebo simulation platform was utilized to construct the system's map, and the robot model was loaded into the map to simulate the LiDAR sensor. Finally, the mapping performance in complex home environments was validated through laser SLAM (Simultaneous Localization and Mapping) technology.

5.1 Robot Modeling

In ROS, URDF (Unified Robot Description Format), an XML-based file format, is widely used to define the visual and physical structure of robot models. Although URDF provides basic structural description capabilities through tags such as <robot>, <joint>, and <link>, its limitations become apparent when dealing with complex robot structures, such as multi-wheeled vehicles. Specifically, each wheel requires independent coding, which not only results in a large amount of code but also increases maintenance complexity.

To address these challenges, Xacro (XML Macros) was introduced as an enhancement to URDF, aiming to improve code modularity and maintainability. This study adopted the Xacro format to construct a simulation model of a domestic service robot. The model features a two-wheel differential drive structure and integrates multiple key components, including a chassis, drive wheels, a LiDAR, a depth camera, and an IMU sensor. The structural diagram of the model is illustrated in Fig 5.

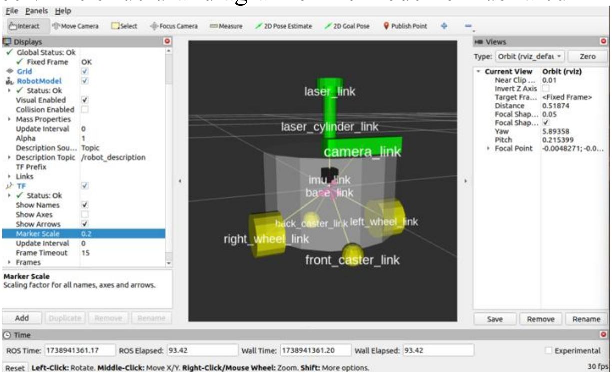

Fig. 5. Robot Structural Diagram (Picture credit: Original)

Moreover, in practical simulations, collisions are inevitable when the robot interacts with other objects, such as the contact between the wheels and the ground, which can also be considered a type of collision. Therefore, to ensure more accurate and safe interactions between the robot and other objects, it is necessary to add collision properties to the robot's components, as illustrated in Fig 6.

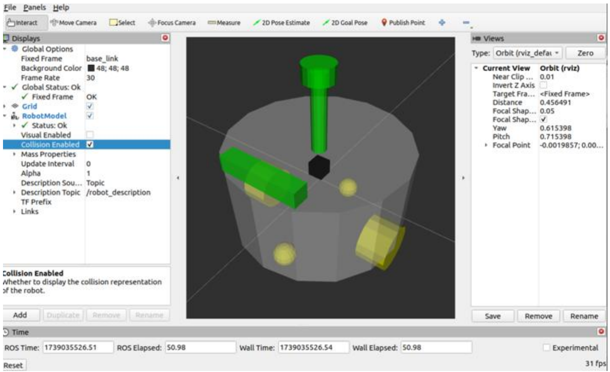

Fig. 6. Collision Visualization Effect (Picture credit: Original)

Real-world robot components possess mass, which introduces inertia during motion. Therefore, during simulation modeling, it is essential to assign mass and inertia properties to the robot's components, as illustrated in Figs 7 and 8.

Fig. 7. Robot Mass Configuration View (Picture credit: Original)

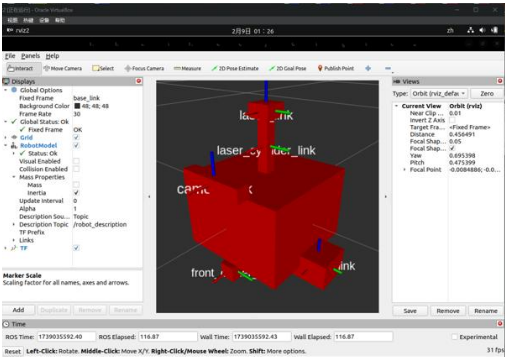

Fig. 8. Robot Inertia Configuration View (Picture credit: Original)

5.2 SLAM mapping simulation

5.2.1 Map Construction.

Firstly, the experimental environment for the robot was constructed based on the Gazebo simulation platform. By utilizing Gazebo's Editing Model function and employing precise dimension modeling methods, a complete layout structure of a four-room and one-living-room apartment was created (as shown in Fig 9). Additionally, considering the practical requirements of robot navigation, various obstacles were strategically placed in the passage and room areas to ensure that the simulation environment not only accurately reflects the spatial characteristics of a home scenario but also provides comprehensive testing conditions for subsequent SLAM mapping.

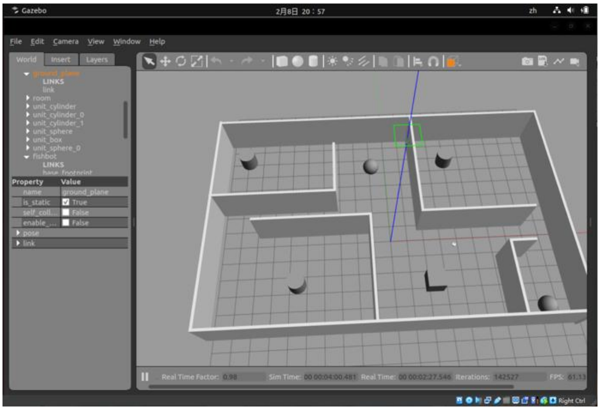

Fig. 9. Map Construction Visualization (Picture credit: Original)

5.2.2 Lidar simulation.

LiDAR (Light Detection and Ranging) is an advanced sensor based on laser ranging technology, which precisely measures the distance to target objects by emitting laser beams and receiving reflected signals. This sensor is capable of generating high-precision and high-frequency three-dimensional point cloud data of the surrounding environment, providing robots with rich spatial perception information. In the Gazebo simulation environment, LiDAR can be conveniently simulated by loading the LiDAR plugin, as illustrated in Fig 10.

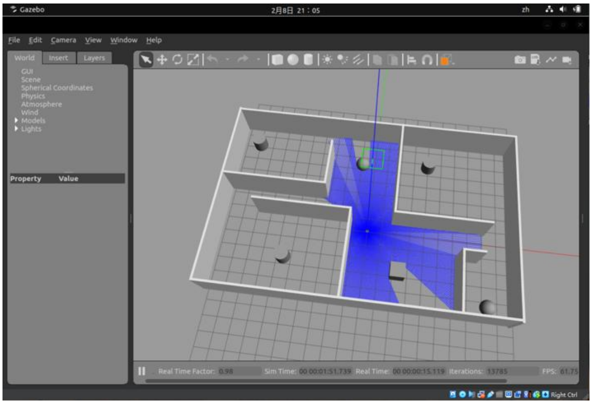

Fig. 10. LiDAR Simulation Visualization (Picture credit: Original)

5.2.3 SLAM Simulation.

Finally, by launching the Gazebo simulation platform and the map construction launch file, the system successfully loaded the simulation environment and the SLAM algorithm. In the Rviz visualization tool, the laser SLAM mapping process could be observed in real time. The initial state, as shown in Fig 11, indicates that the map information was not yet fully constructed. Subsequently, the keyboard control node was activated to manually control the robot's omnidirectional movement within the simulation environment, allowing the LiDAR to thoroughly scan environmental features. As the robot moved, the map information gradually improved, and the final environmental map is presented in Fig 12. By comparing the experimental results with the pre-created standard map, it is evident that the map constructed based on laser SLAM exhibits high consistency with the actual environmental model in terms of contour features, obstacle locations, and spatial layout. This fully validates the accuracy and reliability of the laser SLAM algorithm.

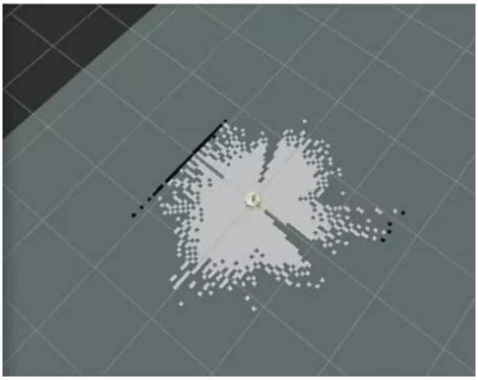

Fig. 11. Initial status of SLAM mapping (Picture credit: Original)

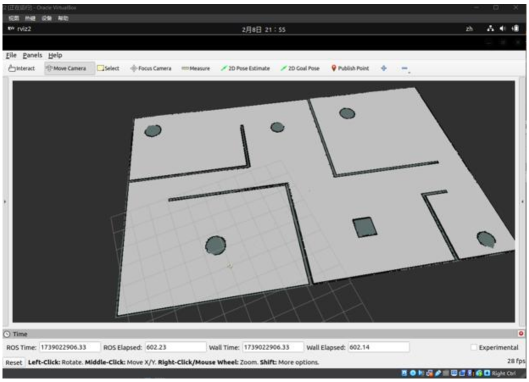

Fig. 12. Renderings of SLAM mapping (Picture credit: Original)

6 Conclusion

The YOLO-SBA object detection algorithm, integrated with the ROS (Robot Operating System) framework, was applied to the navigation system of a home service robot, with its effectiveness verified through the Gazebo simulation platform. First, the YOLO-SBA model was proposed by optimizing the YOLOv8n model and incorporating the SBA (Selective Boundary Attention) module to enhance the accuracy and robustness of object detection. Experimental results demonstrate that the proposed model outperforms YOLOv8n on several key metrics, with precision improving by 2.1%, recall increasing by 10.2%, mAP50 rising by 9.9%, and mAP50:95 increasing by 9.3%.

Second, a multi-sensor fusion system based on the ROS framework was implemented, and a robot simulation model was constructed to improve the robot's perception capabilities in complex home environments. Finally, laser SLAM mapping experiments were conducted using the Gazebo platform, revealing that the system could efficiently perform both object detection and SLAM tasks in intricate household settings, with the generated map showing a high degree of similarity to the real environment. Overall, this study enhances the autonomous navigation capabilities of home service robots and provides significant insights for the development of smart home robotics technology.

References

1. Gonzalez-Aguirre, J. A., et al.: 'Service robots: Trends and technology', Appl. Sci., 2021, 11, (22), pp. 10702
2. Bai, Y., et al.: 'Vision-based navigation and guidance for agricultural autonomous vehicles and robots: A review', Comput. Electron. Agric., 2023, 205, pp. 107584
3. Kaur, R., Singh, S.: 'A comprehensive review of object detection with deep learning', Digit. Signal Process., 2023, 132, pp. 103812
4. Barros, M., et al.: 'A comprehensive survey of visual SLAM algorithms', Robotics, 2022, 11, (1), pp. 24
5. Cai, Y., Ou, Y., Qin, T.: 'Improving SLAM techniques with integrated multi-sensor fusion for 3D reconstruction', Sensors, 2024, 24, (7), pp. 2033
6. Wang, H., et al.: 'YOLOv5-Fog: A multiobjective visual detection algorithm for fog driving scenes based on improved YOLOv5', IEEE Trans. Instrum. Meas., 2022, 71, pp. 1–12
7. Cheng, P. D. C., et al.: 'Dynamic path planning of a mobile robot adopting a costmap layer approach in ROS2', Proc. 2022 IEEE 27th Int. Conf. Emerg. Technol. Factory Autom. (ETFA), IEEE, 2022
8. Maruyama, Y., Kato, S., Azumi, T.: 'Exploring the performance of ROS2', Proc. 13th Int. Conf. Embedded Softw., 2016
9. Al-Batati, A. S., Koubaa, A., Abdelkader, M.: 'ROS 2 Key Challenges and Advances: A Survey of ROS 2 Research, Libraries, and Applications', Preprint, 2024
10. coco128: 'Kaggle', accessed 19 October 2022, https://www.kaggle.com/datasets/ultralytics/coco128
11. Tang, F., et al.: 'DuAT: Dual-Aggregation Transformer Network for Medical Image Segmentation', in Liu, Q., et al. (Ed.): Pattern Recognition and Computer Vision, PRCV 2023, Lecture Notes in Comput. Sci., vol. 14429, Springer, Singapore, 2024, pp. 1–12, https://doi.org/10.1007/978-981-99-8469-5_27

12. Xing, S.: 'ROS2 Robot Development: From Beginner to Practice', Electronics Industry Press, 2021