Efficient Relative Coordinate Inference for Dynamic SLAM Exploiting Monocular Cameras

Seung-Chan Yu
Dept. of Applied Artificial Intelligence
Major in Bio Artificial Intelligence
Ansan, Republic of Korea
mario1470@hanyang.ac.kr

Ji-Sung Park
Dept. of Applied Artificial Intelligence
Major in Bio Artificial Intelligence
Ansan, Republic of Korea
jsdms316@gmail.com

Dong-Ho Lee
Dept. of Applied Artificial Intelligence
Major in Bio Artificial Intelligence
Ansan, Republic of Korea
dhlee72@hanyang.ac.kr

***Abstract*—In the development of autonomous driving robots and related technologies, the successful implementation of SLAM (Simultaneous Localization and Mapping) is crucial. To this end, it is very important to estimate the current position and, orientation of the robot while efficiently constructing a map of the environment. Various algorithms have been proposed so far, utilizing LiDAR, graph-based methods, and inertial systems. However, these methods share common issues, such as high computational and resource costs for map construction, as well as limitations when operating in dynamic environments with numerous moving objects, such as inside large shopping mall. To address these challenges, we propose an algorithm for relative coordinate inference using distance measurement/trilateration, which is applicable even with monocular cameras.**

***Index Terms*—SLAM, Localization, YOLO, Object detection, Distance estimation, Trilateration**

I. Introduction

Autonomous driving technology is being applied in various fields. In order to ensure efficient localization and path planning operation, the application of SLAM (Simultaneous Localization and Mapping) technology is often utilized [1]–[5]. SLAM allows a robot to create a map of its surroundings and track its real-time position within that map. In the context of indoor environments and without access to GPS, SLAM operates as follows.

First, the robot moves through the environment, collecting sensor data and uses it to create a map of the surroundings. Various techniques, including occupancy grids, feature-based maps, and point clouds, can be employed for map generation. After mapping the environment, next the robot estimates the current position and orientation (pose) of the robot. The robot utilizes visual information from cameras, data from distance measurement sensors like LiDAR, or other sources to estimate its position. After finishing localization, the robot uses the generated map to decide where to move next and plan paths. This is particularly useful for autonomous exploration or performing tasks in an environment. Over time, errors may accumulate in the estimated robot position and the generated map. Loop closure identifies revisited locations and corrects these errors. By recognizing previously visited places, the robot can refine the map and adjust the estimated trajectory.

However, when applying SLAM in specific environments, several issues arise. Errors in the process of location estimation and loop closure system from the following weaknesses inherent in SLAM: (1) Environments with high human traffic or dynamic elements, such as hospitals or complex shopping malls, pose challenges for SLAM algorithms. Existing algorithms struggle to cope with such dynamic conditions. (2) SLAM inherently requires significant computation and computing resources. Real-time exploration, map generation, and position estimation tasks demand substantial processing power, leading to increased hardware costs. (3) SLAM's performance heavily relies on sensor availability and quality. In many cases, high-performance sensors such as depth cameras, inertial sensors, and radar systems are required, adding to the overall system cost.

To address these challenges, this paper proposes an algorithm to enhance the position estimation of SLAM. By assuming scenarios where the robot does not deviate significantly from a specific location, a distance estimation and trilateration technique using relatively inexpensive sensors like monocular cameras is employed. Learn the size and positions of static elements that persistently exist in specific locations and do not change its size or position. This learned information can then be utilized to enable location estimation using only a monocular camera, ensuring efficient localization and path planning. This approach offers the following contributions:

- Robust Localization Method for Special Environments: An implementation of robust localization methods suitable for environments with high human traffic or indoor limitations, such as hospitals or shopping malls.
- Cost Reduction in System Implementation: By enabling position estimation with monocular cameras, system implementation costs can be reduced.
- Simplified Computation with a Basic Algorithm: Improved calculation speed is achieved through a simple distance-based position estimation approach between simple objects and agents using YOLO-based object

This work was partly supported by the Institute of Information communications Technology Planning Evaluation (IITP) grant funded by the Korea government (MSIT) (No.RS-2022-00155885, Artificial Intelligence Convergence Innovation Human Resources Development (Hanyang University ERICA)), the National Research Foundation of Korea(NRF) grant funded by the Korea government (MSIT) (No.NRF-2022R1F1A1073208) and Korea Institute for Advancement of Technology (KIAT) grant funded by the Korea Government (MOTIE) (P0008691, HRD Program for Industrial Innovation).

979-8-3503-1327-7/23/$31.00 ©2023 IEEE

recognition.

II. Related Works

Visual SLAM [6] is a type of SLAM technology that utilizes visual information to simultaneously estimate the robot's position and create a map of the environment. Visual information is primarily obtained by analyzing images captured through cameras. When analyzing images, it can be categorized into pixel-based SLAM, which analyzes images at the pixel level, and feature-based SLAM, which extracts and analyzes key features. Recent state-of-the-art SLAM techniques predominantly use feature-based SLAM [7]. Visual SLAM offers advantages such as low sensor dependence and suitability for real-time processing.

As a branch of Visual SLAM, this approach involves representing data and constraints in the form of a graph when estimating the robot's pose or exploring the surrounding environment. It enables consistent and accurate pose estimation and map creation in various environments, and facilitates the complex integration of multiple sensors [8].

LiDAR SLAM is a method that utilizes laser light sensors to measure distances to surrounding objects and construct maps. It offers high precision and strong obstacle detection capabilities. Additionally, its fast distance measurement speed makes it suitable for real-time operations [9], [10].

ORB-SLAM has long been at the forefront of SLAM technology [11]. It utilizes features such as FAST [12] and BRIEF [13] to detect image features. Subsequent research has improved both accuracy and speed by incorporating inertial sensors [14] and implementing process parallelization [15].

However, as mentioned in previous section, there exists a common weaknesses in these SLAM algorithm, which is struggling with dynamic conditions and requiring high computational cost. We aim to propose an economically viable localization algorithm that can address these weakness.

III. Background and Methods

The proposed algorithm assumes continuous usage in the same location. The elements used for localization are stationary elements existing in that location, and a learning process is required for each machine to recognize these elements. It would be suitable for a continuously operating guidance robot in a single complex shopping mall or a hospital, for instance.

A. Yolov8

YOLO predicts bounding boxes and classes for all objects in the image in a single pass analysis. Unlike traditional object detection methods, it processes the entire image at once without dividing it into grids, enabling real-time processing [16]–[20].

B. Distance estimation

Distance measurement with monocular camera [21] in such environments is achieved through distinctive landmark features that are fixed and uniquely identifiable. If a landmark is N and the camera is C, the distance D_{NC} between these two elements can be calculated as follows:

$$D_{NC} = \frac{w_{rn} \times FL}{w_{sn}} \tag{1}$$

Here, w_{rn} represents the actual width of element N, and w_{sn} denotes the width of element N in the camera view. In the experimental process of this paper, the units for these values were set to centimeters (cm) and pixels, respectively. FL refers to the camera's focal length. In the case of the *Realsense D455i* camera used in the experiment, the focal length is set to 1.95mm. However, as shown in equation (1), the focal length need to be converted to a value in pixel units to calculate distance D_{NC} in centimeters. Therefore, the focal length was computed based on imagery captured at a fixed distance D as follows:

$$FL = \frac{w_{sn} \times D}{w_{rn}} \tag{2}$$

C. Trilateration with considering errors

When three or more fixed elements are detected on the screen, the current coordinates of the robot are calculated through trilateration. Let $K(x, y)$ denote the current position of the desired camera, and let $N_i(X_i, Y_i)$ represent the fixed coordinates for the detected ith object. If D_i represents the measured distance between object N_i and camera K, the following equation can be obtained:

$${D_i}^2 = (X_i - x)^2 + (Y_i - y)^2 \tag{3}$$

In general, during the process of performing trilateration, the following equation is derived and used:

$$\mathbf{x} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} \tag{4}$$

where the desired current camera's position vector x and constants A, b are defined as follows [22]:

$$\mathbf{x} = \begin{bmatrix} x \\ y \end{bmatrix} \tag{5}$$

$$\mathbf{A} = \begin{bmatrix} 2(X_1 - X_0) & 2(Y_1 - Y_0) \\ 2(X_2 - X_1) & 2(Y_2 - Y_1) \\ \vdots & \vdots \\ 2(X_n - X_{n-1}) & 2(Y_n - Y_{n-1}) \\ 2(X_0 - X_n) & 2(Y_0 - Y_n) \end{bmatrix} \tag{6}$$

$$\mathbf{b} = \begin{bmatrix} {D_0}^2 - {D_1}^2 + {X_1}^2 - {X_0}^2 + {Y_1}^2 - {Y_0}^2 \\ {D_1}^2 - {D_2}^2 + {X_2}^2 - {X_1}^2 + {Y_2}^2 - {Y_1}^2 \\ \vdots \\ {D_{n-1}}^2 - {D_n}^2 + {X_n}^2 - {X_{n-1}}^2 + {Y_n}^2 - {Y_{n-1}}^2 \\ {D_n}^2 - {D_0}^2 + {X_0}^2 - {X_n}^2 + {Y_0}^2 - {Y_n}^2 \end{bmatrix} \tag{7}$$

where n represents the number of fixed elements used for position estimation, and is typically set to the dimension specified plus 1 in order to determine the elements. However, there is a potential issue with this calculation method. In cases where errors are easily introduced, such as with monocular camera-based position estimation, even slight errors can result in significant deviations in the estimated position. Therefore, this paper applies error bounds to the measured distances to account for this issue and calculate the estimated region. Unlike the traditional trilateration method, this approach allows for more robust position calculation in the presence of errors. The estimated region can be defined as the set K which is satisfying following equation:

$$K = \{(x, y) | \forall i (D_i - err)^2 < (x - X_i)^2 + (y - Y_i)^2 < (D_i + err)^2\} \tag{8}$$

Here, the error range err determines the thickness of each ring in Fig. 1 example, and the calculated set of points *K* represents the shaded area with blue dots. If no region satisfying the above equation exists, the value of err can be increased and the algorithm can be rerun to find a solution. The proposed algorithm involves finding all points that is included in set *K* satisfy equation (8), and then calculating in which region the highest density of points exists. The user can adjust and control the density of points and the size of the regions according to their preferences. However, this may come at the cost of increased computational effort. By excluding information about the current position each time and utilizing only detected fixed elements to estimate the region, the problem of error accumulation can be mitigated.

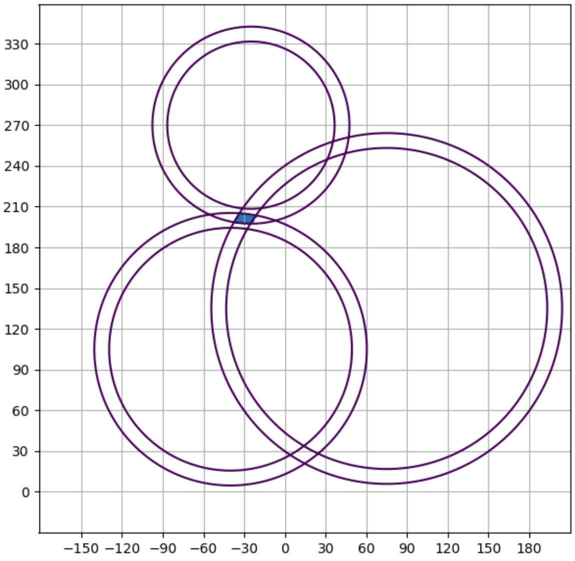


Fig. 1. Example of calculating the range for the expected position using error bounds. The thickness of each ring means amount of error to regard. The region colored with blue dots is the estimated camera's position.

IV. Evaluation

In this paper, a newly generated dataset was utilized during the experimental process. Given the nature of the proposed system, the dataset must adhere to specific criteria, where each label accurately corresponds to a unique object. This requirement varies depending on the applied context. To create a dataset that satisfies these conditions, various images of internal elements within different hospitals available on the web were used.

A. Environment

1) Robot Environment
   - ROS2 Foxy
   - Turtlebot3 - Waffle Pi model
   - Realsense Camera D455i
2) Computer Environment
   - Operating System: Ubuntu 20.04.6 LTS 64-bit
   - CPU: 11th Gen Intel® Core™ i9-11900 @ 2.50GHz × 16
   - GPU: NVIDIA GeForce RTX 3090 × 2
   - RAM: 96GB

B. Dataset

The dataset contains a total of 10 labels, and there are 560 data samples in the dataset. For training, 90% (504 samples) were used, and for validation, 10% (56 samples) were utilized. The training results are as follows. The training and validation results are presented in Fig. 2.

Fig. 3 depicts the training results of yolov8 utilizing the dataset. The pre-trained model yolov8l was employed, with 20 epochs and a batch size of 16 during training. In Fig. 3 (b), the validation results demonstrate accurate differentiation of most fixed elements. However, given the relatively limited dataset size, there is a potential risk of overfitting. Therefore, for actual application, it is recommended to use a larger volume of data.

C. Section estimation evaluation

The experimental objectives and evaluation criteria are as follows.

1) Place fixed elements and cameras in arbitrary locations and compare the detection capabilities and distance measurement accuracy in YOLOv8.
2) Utilize the detected fixed elements to estimate the current region and compare the accuracy.

These objectives outline the goals of the experiments, and the evaluation criteria focus on comparing detection abilities, distance measurement accuracy, and region estimation accuracy under different conditions and scenarios.

The experiments were conducted by moving the camera's position based on fixed elements and predicting the region in which the moved camera is located. In one experiment, the positions of fixed elements were altered to observe the impact of such changes, while assessing the algorithm's robustness. Table 1 presents a subset of the experimental results. Each *Fixed element* is used as object N_i in equation (1). Experiment (A), (B), and (C) have correct result for estimating even if there are some errors between estimated distance and real distance between camera and each fixed element. Fig. 4 illustrates the distances and regions calculated by the algorithm for each experiment in Table 1. In graph (B), the thickness of each ring

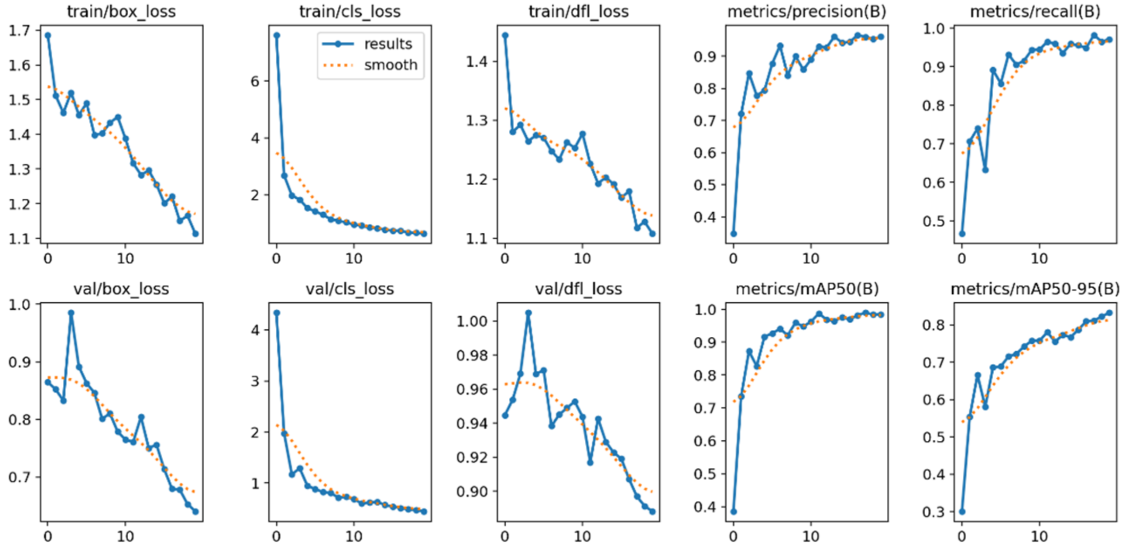
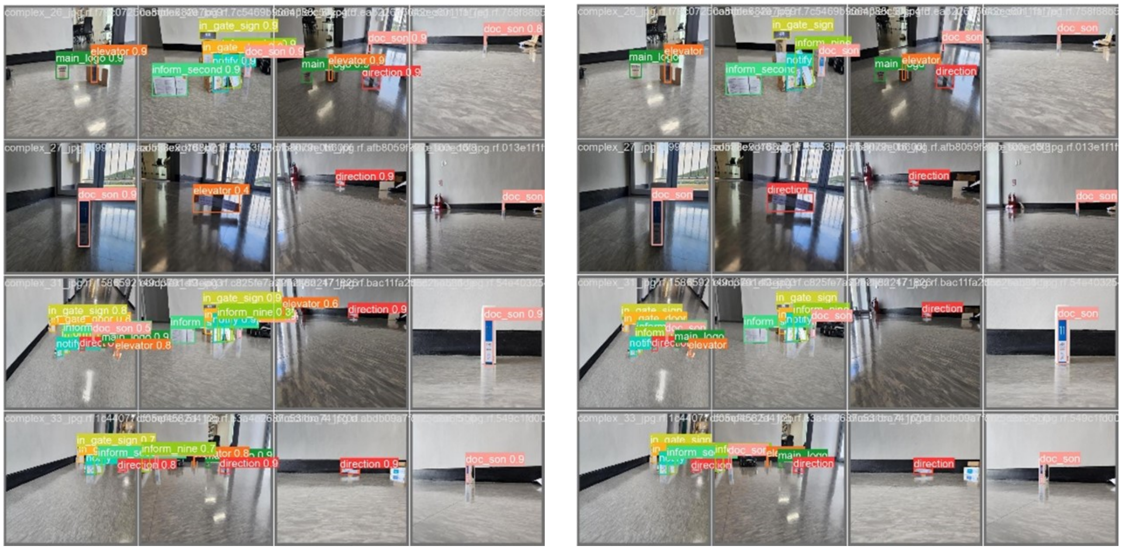


Fig. 2. The results of fine-tunning YOLOv8 using the generated dataset.



(a)

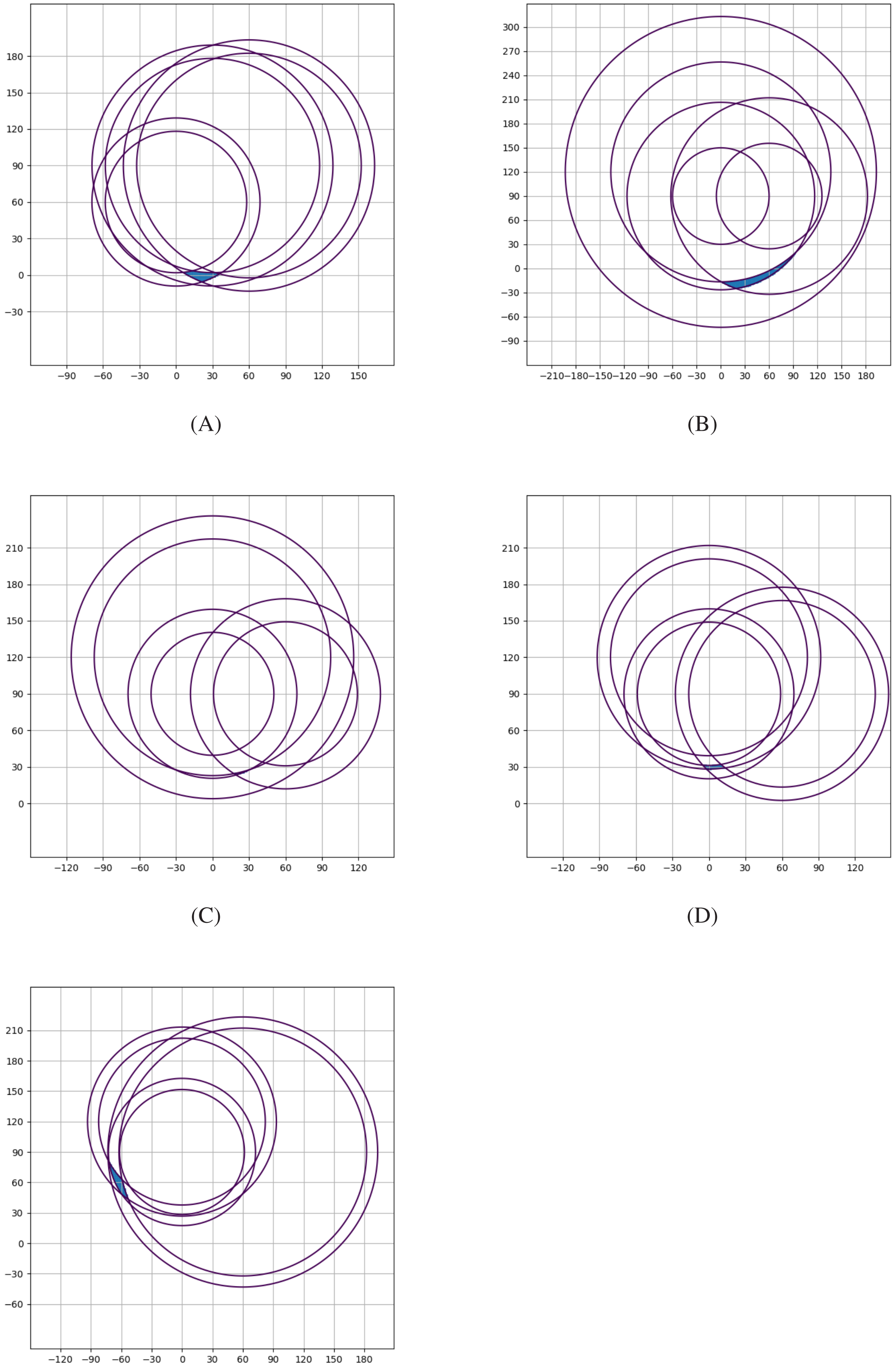
(b)

Fig. 3. The results of fine-tunning YOLOv8 using the generated dataset. (a) Ground-truth of validation set. (b) Prediction of validation set.

TABLE I
SAMPLES OF EXPERIMENTAL RESULTS.

Camera position		**Fixed element 1**			**Fixed element 2**			**Fixed element 3**			**Estimated Section**
		Real position	*Estimated distance*	*Real distance*	*Real position*	*Estimated distance*	*Real distance*	*Real position*	*Estimated distance*	*Real distance*	
(A)	**(30, 0)**	**(0, 90)**	**88.20**	**94.87**	**(60, 90)**	**93.8**	**94.87**	**(0, 120)**	**164.80**	**123.70**	**(15 ∼ 45, -15 ∼ 15)**
(B)	**(30, 30)**	**(0, 90)**	**61.13**	**67.08**	**(60, 90)**	**67.69**	**67.08**	**(0, 120)**	**104.60**	**94.87**	**(15 ∼ 45, 15 ∼ 45)**
(C)	**(30, 30)**	**(30, 60)**	**49.00**	**30.00**	**(60, 90)**	**69.64**	**67.08**	**(0, 120)**	**97.12**	**94.87**	**(15 ∼ 45, 15 ∼ 45)**
(D)	(0, 0)	(0, 90)	96.47	90.00	(60, 90)	109.43	108.17	(0, 120)	205.25	120.00	(15 ∼ 45, -45 ∼ -15)
(E)	(-30, 30)	(0, 90)	67.11	67.08	(60, 90)	127.67	108.17	(0, 120)	87.73	94.87	(-75 ∼ -45, 45 ∼ 75)

[a] The unit of given table is centimeter (cm). The rows with bold entries are the correct estimations



(A)

(B)

(C)

(D)

(E)

Fig. 4. The results of fine-tunning YOLOv8 using the generated dataset.

shape is greater compared to the other graphs, which signifies a larger range of error. Interestingly, it was observed that even with this increased error range, a reasonably accurate estimation of the regions is achievable. However, when observing the graphs for (D) and (E), it's evident that despite having a narrower error range, the results are inaccurate. A common factor in these two experiments is the significant distortion caused by the angle between the camera and the objects. It can be concluded that dealing with errors induced by this angle-related distortion proved to be challenging.

V. Conclusion

This paper proposes an algorithm to enhance the localization and map generation in fixed environments for autonomous robots. The experimental results confirm the effectiveness of the proposed approach in achieving robust localization and region estimation in various settings. Furthermore, the use of a relatively inexpensive monocular camera demonstrates the potential to achieve accurate position estimation while reducing implementation costs.

However, since there are some limitations to this study, we suggest the following subsequent research to address these issues:

- A significant source of error in distance measurement to fixed elements is the size of the field of view. Especially in areas like corridors, where front-facing views of fixed elements are scarce, there is a need for distance estimation methods that can account for this limitation. To resolve this issue, research like storing object recognition information by rotating the camera in a panoramic fashion, or storing information about short-distance movements of the robot could be conducted.
- The current distance measurement algorithms are susceptible to distortions such as object rotation or objects positioned at the camera's edges. Especially for the same object, the size of the object detection bounding box can vary depending on the viewing angle. This can lead to significant errors in distance measurement. To address these challenges, it may be beneficial to apply algorithms that estimate depth from images through learning in monocular cameras. This could potentially enhance the accuracy of distance calculations.

References

[1] A. Singandhupe and H. M. La, "A review of slam techniques and security in autonomous driving," in 2019 third IEEE international conference on robotic computing (IRC), (IEEE, 2019), pp. 602–607.

[2] H. Lategahn, A. Geiger, and B. Kitt, "Visual slam for autonomous ground vehicles," in 2011 IEEE International Conference on Robotics and Automation, (IEEE, 2011), pp. 1732–1737.

[3] T. Mill, A. Alt, and R. Liias, "Combined 3d building surveying techniques–terrestrial laser scanning (tls) and total station surveying for bim data management purposes," J. Civ. Eng. Manag. 19(Supplement_1), S23–S32 (2013).

[4] V. Krauß, A. Boden, L. Oppermann, and R. Reiners, "Current practices, challenges, and design implications for collaborative ar/vr application development," in Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, (2021), pp. 1–15.

[5] Yang Zhao, Haotian Yu, Kai Zhang, Yucheng Zheng, Yi Zhang, Dongliang Zheng, and Jing Han, "FPP-SLAM: indoor simultaneous localization and mapping based on fringe projection profilometry," Opt. Express 31, 5853-5871 (2023)

[6] Sharafutdinov, D., Griguletskii, M., Kopanev, P. et al. Comparison of modern open-source Visual SLAM approaches. J Intell Robot Syst 107, 43 (2023). https://doi.org/10.1007/s10846-023-01812-7

[7] Walter MR, Eustice RM, Leonard JJ. Exactly Sparse Extended Information Filters for Feature-based SLAM. The International Journal of Robotics Research. 2007;26(4):335-359. doi:10.1177/0278364906075026

[8] G. Grisetti, R. Kümmerle, C. Stachniss and W. Burgard, "A Tutorial on Graph-Based SLAM," in IEEE Intelligent Transportation Systems Magazine, vol. 2, no. 4, pp. 31-43, winter 2010, doi:

[9] W. Hess, D. Kohler, H. Rapp and D. Andor, "Real-time loop closure in 2D LIDAR SLAM," 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 2016, pp. 1271-1278, doi: 10.1109/ICRA.2016.7487258.R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[10] K. Liu and M. Cao, "DLC-SLAM: A Robust LiDAR-SLAM System With Learning-Based Denoising and Loop Closure," in IEEE/ASME Transactions on Mechatronics, doi: 10.1109/TMECH.2023.3253715. 10.1109/MITS.2010.939925.I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[11] R. Mur-Artal, J. M. M. Montiel and J. D. Tardós, "ORB-SLAM: A Versatile and Accurate Monocular SLAM System," in IEEE Transactions on Robotics, vol. 31, no. 5, pp. 1147-1163, Oct. 2015, doi: 10.1109/TRO.2015.2463671

[12] Newman, M. E. 2004. "Fast Algorithm for Detecting Community Structure in Networks." Physical Review E 69 (6): 066133. doi:10.1103/PhysRevE.69.066133

[13] Calonder, M., Lepetit, V., Strecha, C., Fua, P. (2010). BRIEF: Binary Robust Independent Elementary Features. In: Daniilidis, K., Maragos, P., Paragios, N. (eds) Computer Vision – ECCV 2010. ECCV 2010. Lecture Notes in Computer Science, vol 6314. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-15561-1_56

[14] Mur-Artal, Raúl, and Juan D. Tardós. "Visual-inertial monocular SLAM with map reuse." IEEE Robotics and Automation Letters 2.2 (2017): 796-803.

[15] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel and J. D. Tardós, "ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM," in IEEE Transactions on Robotics, vol. 37, no. 6, pp. 1874-1890, Dec. 2021, doi: 10.1109/TRO.2021.3075644.

[16] Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.

[17] Jiang, Peiyuan, et al. "A Review of Yolo algorithm developments." Procedia Computer Science 199 (2022): 1066-1073.

[18] C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," 2022, arXiv:2207.02696.

[19] Terven, Juan, and Diana Cordova-Esparza. "A comprehensive review of YOLO: From YOLOv1 to YOLOv8 and beyond." arXiv preprint arXiv:2304.00501 (2023).

[20] Lou, Haitong, et al. "DC-YOLOv8: Small-Size Object Detection Algorithm Based on Camera Sensor." Electronics 12.10 (2023): 2323.

[21] Yamaguti, Naoki, Shunichiro Oe, and Kenji Terada. "A method of distance measurement by using monocular camera." Proceedings of the 36th SICE annual conference. International session papers. IEEE, 1997.

[22] F. Thomas and L. Ros, "Revisiting trilateration for robot localization," in IEEE Transactions on Robotics, vol. 21, no. 1, pp. 93-101, Feb. 2005, doi: 10.1109/TRO.2004.833793.M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.