2025 IEEE/SICE International Symposium on System Integration (SII)
January 21-24, 2025, Munich, Germany

Accurate and Cost-Scalable Panorama Visual-Geometric-Matching based Localization System for Robot Navigation

Takuma Nakao[1] and Junji Takahashi[1]

***Abstract*— To operate a large number of autonomous robots, an accurate and low-cost localization technology is recommended. To meet this demand, we have proposed an original localization approach, Visual-Geometric-Matching (VGM), using a monocular RGB camera and a pre-build map, and have developed it as the client-server localization system. In this paper, we propose a panorama-VGM method, which achieves efficient memory usage and efficient image matching. By applying a cylindrical panorama transform to the template images, the redundant information is eliminated, and the VRAM memory usage is significantly reduced. The panorama transform is also applied to query image from a client, and both images are matched on panoramic projection surface. Furthermore, we integrated the results of panorama-VGM with the odometry data by Extended Kalman Filter and utilize for a mobile robot navigation. Experimental results show that the proposed panorama-VGM is accurate, robust and practical localization system. Specifically: (1) the median error is 0.11 [m] in the single algorithm accuracy evaluation, (2) the median error is 0.12 [m] in the mobile robot navigation experiment compared with LiDAR + EMCL, (3) the travel distance is 2,590 [m] during a continuous 77-minute durability test.**

I. Introduction

The LiDAR localization approach has been studied for a long time, and a many effective SLAM[1] methodologies including Monte Carlo Localization (MCL)[2], point cloud matching such as ICP and NDT, have been proposed. However, LiDAR sensors are generally expensive and the cost can be a disadvantage in applications where many robots are used. Therefore, the Visual SLAM approach[3], assuming the use of low-cost RGB camera has been proposed. The relative distance of continuous frames is estimated by comparing image futures. This approach makes it difficult to map frames in a low-texture environment. Yang proposed Pop-Up SLAM[4], witch can be used in low-texture environments; however few cases have been reported where it has been applied to robot navigation.

Assuming a precise prior-map, the robot localization problem would be easier. Some studies focus on the application of Building Information Modeling (BIM) in robotic technology[5][6]. Debaditya proposed BIM-PoseNet[7] that localize RGB camera in 3D indoor model using CNN trained by CG images generated from the model. Chen proposed A2L[8] that localizes a camera by aligning a photogrammetric point cloud with a BIM-referenced point cloud. In either method, the accuracy is on the order of 1 meter, and further improvements are required for use in robot navigation.

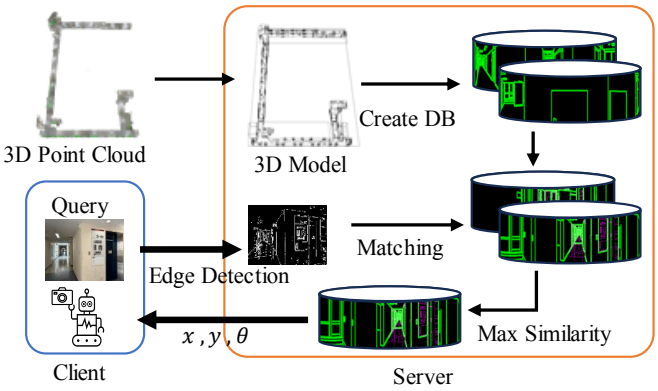


Fig. 1: Workflow of panorama-VGM.

We have proposed an original approach using Visual-Geometric-Matching (VGM) so as to provide accurate localization for practical multiple robot navigation at a reasonable cost. The VGM method localizes an image taken by RGB camera within a pre-build 3D model and performs as a positioning infrastructure system by being integrated into the server-client system[9]. Although this system has the potential to provide localization with sufficient accuracy for robot navigation, it requires a large amount of VRAM memory making it difficult to apply to large-scale environment.

In this paper, we propose a panorama-VGM method which achieves efficient memory usage and efficient image matching (Fig. 1). The cylindrical panorama transformation is applied to both template images and a query image in order to realize an efficient matching process. We also develop a sensor fusion system to merge the results of both panorama-VGM and odometry data of a mobile robot, based on Extended Kalman Filter (EKF).

In Section II, our proposing panorama-VGM method is described after brief explanation of VGM. In Section III, the sensor fusion method for applying panorama-VGM to robot navigation is described. Section IV presents experiments of the single performance evaluation of panorama-VGM. Section V presents experiments of the robot navigation. Section VI provides a summary of this paper.

II. Panorama Visual-Geometric-Matching

A. Overview of VGM

Our proposed method named "Visual-Geometric-Matching" requires a 3D model of the target environment as a pre-build map. If available, the data of Building

This work has been partially supported by the New Energy and Industrial Technology Development Organization (NEDO) JPNP20004

[1]Takuma Nakao and Junji Takahashi are with Department of Mechanical Engineering, Graduate School of Engineering, Toyohashi University of Technology, 1-1 Hibarigaoka, Tempaku, Toyohashi, Aichi, 441-8580, JAPAN nakao.takuma.un@tut.jp takahashi@me.tut.ac.jp

Information Modeling (BIM) can be used as the 3D model; otherwise, the 3D model can be made by the process including, scanning by precise laser sensor (Focus, FARO) and modeling by a software (ClassNK-PEERLESS, ARMONICOS CO., LTD.). For matching with 2D query image from clients, 2D template images with 3-DoF camera pose are generated by projection from the 3D model.

In the matching process, every pixel of query image is compared with every pixel of template images in the database one by one, and the similarity between two images is evaluated based on whether the pixels of line-segment are overlapped or not. As a result, the pose of template image that provides the maximum similarity is output. Thanks to the GPU computing, the process is speedy.

In VGM, the accuracy of position estimation and the range of position estimation are limited by the number of template images. In the conventional method, multiple template images are created by dividing the yaw angle into small segments. For example, if the images are created in 1-degree increments, 360 template images are created for each x,y position. However, this approach leads to most of the images containing duplicated data. Therefore, we propose panorama-VGM, which reduces the number of images by replacing them with a single panoramic image.

B. Panorama transformation

We define 4 types of images:

- Basic template image: $\boldsymbol{P}_{bt}$ (P_{bt}^{row}, P_{bt}^{col})
- Basic query image: $\boldsymbol{P}_{bq}$ (P_{bq}^{row}, P_{bq}^{col})
- Panorama template image: $\boldsymbol{P}_{pt}$ (P_{pt}^{row}, P_{pt}^{col})
- Panorama query image: $\boldsymbol{P}_{pq}$ (P_{pq}^{row}, P_{pq}^{col})

where row and col represent the height and width positions in pixels of the image, respectively. In Fig. 2 (a) ϕ_{bq}, w_{bq}, and f_{bq} represent the angle of view, the width, the focal length of the basic query image, respectively. R'_{pq} represents the calculational radius of the cylinder onto which the panoramic query image is projected, and w'_{pq} represents the calculational arc length of the panoramic query image. Then, they have a relation:

$$w'_{pq} = \phi_{bq} R'_{pq}. \tag{1}$$

To project all pixels of the basic query image onto the panoramic query image, the projection surface must be located outer to the basic image, that means,

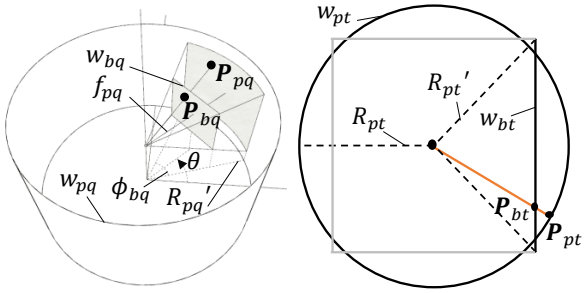
$$\frac{w_{bq}}{2\sin\left(\frac{\phi_{bq}}{2}\right)} \leq R'_{pq}. \tag{2}$$

To keep image expansion and contraction as small as possible during the panoramic projection, it is desirable for the radius of the cylinder to satisfy the following conditions:

$$f_{bq} = \frac{w_{bq}}{2\tan\left(\frac{\phi_{bq}}{2}\right)} \leq R'_{pq} \leq \frac{w_{bq}}{2\sin\left(\frac{\phi_{bq}}{2}\right)}. \tag{3}$$

From Eq. 2 and Eq. 3, R'_{pq} becomes,

$$R'_{pq} = \frac{w_{bq}}{2\sin\left(\frac{\phi_{bq}}{2}\right)}. \tag{4}$$



(a) Query image case (b) Template image case

Fig. 2: Parameters for panorama conversion.

Using Eq. 1, Eq. 4 and the integer function $\mathrm{INT}()$, the width of the panorama query image is derived:

$$w_{pq} = \mathrm{INT}\left(\phi_{bq} R'_{pq}\right). \tag{5}$$

after w'_{pq} quantization, the radius of the panoramic query image becomes,

$$R_{pq} = \frac{w_{pq}}{\phi_{bq}}. \tag{6}$$

For easier implementation, we set that the height of the panoramic query image is equal to that of the basic query image,

$$h_{pq} = h_{bq}. \tag{7}$$

Using above equations, the projection from the basic query image to the panoramic query image is given as follows:

$$P_{pq}^{col} = \frac{w_{pq}}{2\pi} \tan^{-1}\left(\frac{P_{bq}^{col}}{\alpha_{pq}\cos\left(\frac{\phi_{bq}}{2}\right)}\right), \tag{8}$$

$$P_{pq}^{row} = \frac{P_{bq}^{row}}{\sqrt{\alpha_{pq}^2\cos^2\left(\frac{\phi_{bq}}{2}\right) + {P_{bq}^{col}}^2}}, \tag{9}$$

$$\alpha_{pq} = \frac{R'_{pq}}{R_{pq}}, \tag{10}$$

where α_{pq} is defined to remove the effect of quantization error due to the integer function.

Next, the panorama transformation of the template image is explained (Fig. 2 (b)). w_{bt} and w_{pt} denote the width of the basic template image and the width of the panorama template image, respectively. R'_{pt} and R_{pt} denote the calculational radius and the actual radius of the cylinder projecting the panorama images, respectively. For efficient matching, the calculational radius of panorama cylinder of template image is set to be equal to that of query image:

$$R'_{pt} = R'_{pq}. \tag{11}$$

Then the width of the panoramic template image and the radius of it are derived as follows:

$$w_{pt} = \mathrm{INT}\left(2\pi R'_{pt}\right), \tag{12}$$

$$R_{pt} = \frac{w_{pt}}{2\pi}. \tag{13}$$

The panoramic template image is generated from four basic template images, each with mutually exclusive eye directions, when the horizontal angle of view is $\frac{\pi}{2}$. In this case, the width of the basic template image is given as follows:

$$w_{bt} = 2\sin\left(\frac{\phi_{bt}}{2}\right) R_{pt} = \sqrt{2} R_{pt}. \tag{14}$$

The panorama projection of the template image can be computed as well as the projection of the query image,

$$P_{pt}^{col} = \frac{w_{pt}}{2\pi} \tan^{-1}\left(\frac{P_{bt}^{col}}{\alpha_{pt}\cos\left(\frac{\phi_{bt}}{2}\right)}\right), \tag{15}$$

$$P_{pt}^{row} = \frac{P_{bt}^{row}}{\sqrt{\alpha_{pt}^2 \cos^2\left(\frac{\phi_{bt}}{2}\right) + {P_{bt}^{col}}^2}}, \tag{16}$$

$$\alpha_{pt} = \frac{R'_{pt}}{R_{pt}}, \tag{17}$$

where α_{pt} is defined to remove the effects of quantization error due to the integer function.

In the matching process, the camera yaw direction (θ) is varied at regular intervals from 0 to 2π by shifting the pixels to find the direction with the maximum similarity. This operation estimates the yaw direction as well as the translation.

C. Narrowing function of matching range

Although our proposed VGM is available for global localization, restriction the search range around the approximate current location based on historical location data can reduce the computational time and present the incorrect matching caused by similar building structures. We implemented a function to narrow matching range by listing the template images within the selected area. We also prepared a communication interface that allows clients to set the area when making location inquiries with center pose: x_c, y_c and θ_c and ranges: m_d and m_θ. The template images within the following conditions are listed and used in matching process:

$$|x - x_c| \leq m_d, \quad |y - y_c| \leq m_d, \quad |\theta - \theta_c| \leq m_\theta. \tag{18}$$

III. Applying for Robot Localization

The results of VGM are integrated with the odometry data of robot by Extended-Kalman-Filter (EKF) to estimate better pose and position. There are several hundred milliseconds of time delay using VGM positioning system due to VGM process on the server and round trip communication. To compensate the time delays, Augmented State Kalman Filter[10] where the past information is explicitly considered as augmented dimension. By this technique, information including time delay, is regarded to be measured with no-delay as part of the augmented dimension. For the problems of different frequencies between measurements, the information of VGM, which is low frequency, is dispersed into augmented dimensions so as to update the pose smoothly. These two techniques are implemented referring to Autoware[11].

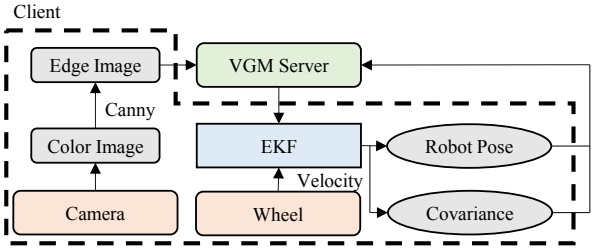


Fig. 3: Workflow of robot localization.

TABLE I: Parameters of server setting.

Parameter	Value
query image size for VGM	320×240[px]
Gradient dilation width of template image	5[px]
Gradient dilation bottom brightness	30
Canny threshold1	50.0
Canny threshold2	110.0

Fig. 3 shows the localization flow applied for a robot. Since the robot used in this study is an omnidirectional mobile robot, the state variables of Kalman filter are set as follows,

$$\boldsymbol{x} = \begin{bmatrix} x & y & \theta & v_x & v_y & \omega \end{bmatrix}^T. \tag{19}$$

The EKF predicts the position of the robot and the covariance matrix that represents its uncertainty. The covariance matrix is used to determine the range of position requests (m_d, m_θ) to the server and the traveling speed of the robot.

IV. Single Performance Evaluation

A. Experimental Condition

The single performance of our proposed panorama-VGM was evaluated in the environment shown in Fig. 4. The dataset includes fifty number of query images taken with an iPhone 15 along with corresponding coordinate positions calculated by AMCL from LiDAR data collected at the same time. The specs of the server machine are as follows: CPU: Intel Core i9-12900K, RAM: 64GB, and NVIDIA RTX 4090. The parameters of the panorama-VGA are listed in Table I. Three types of template image databases are prepared and used for the evaluation; the intervals of template images in the x and y directions are $d_x, d_y = 0.1, 0.2, 0.4$ [m]. The narrowing function is not used in the experiments here.

B. Results

Fig. 5 shows the result of the Canny edge detection, while Fig. 6 shows the result images of matching with a query image and a panorama template image. The colored pixels magenta, green, and white represent the edges of the edged-query image, the edges of the template image, and the pixels where the edges of both images overlapped respectively. In case (a), the edges of the building structures in both images match well, despite some noisy edges caused by reflections on the floor. In contrast, case (b), shows that there are too few edges in the query image to match correctly. In case (c), the matching fails due to the similar appearance caused by the repetitive structure of the building.

The results of localization errors for all query images are presented in a histogram (Fig. 7), where the cumulative proportions are plotted against log-scaled distance errors. And Table II shows statistics on the distance errors, while Table III shows the processing times for each interval setting of the template images. Although our proposed panorama-VGM is not perfect and may output wrong poses, the low rate of wrong outputs suggests that they can be compensated by fusion techniques.

V. Experiments of Robot Navigation

A. Experimental Condition

We conducted two types of experiments, the one is accuracy evaluation and the other is durability test, in the environment depicted in Fig. 4. The robot system used for this experiment is shown in Fig. 8. The robot chassis (SCOUT mini mecanum wheel, AgileX Robotics Ltd.) mounts the camera (c922n, Logitech) for panorama-VGM, the onboard computer (ThinkPad X1 5th-Gen, Lenovo) with ROS 2 Humble, and the LiDAR (UTM-30LX-EW, HOKUYO Co.,LTD.) for position reference. The parameters of server settings are the same as those used in the experiment described in Chapter IV($d_x, d_y = 0.1$ [m]). Note that the LiDAR is only used to create ground truth data for the accuracy evaluation and is not used for robot navigation in either experiments.

The robot localization method is described in Chapter III. The range of the narrowing function of panorama-VGM is set

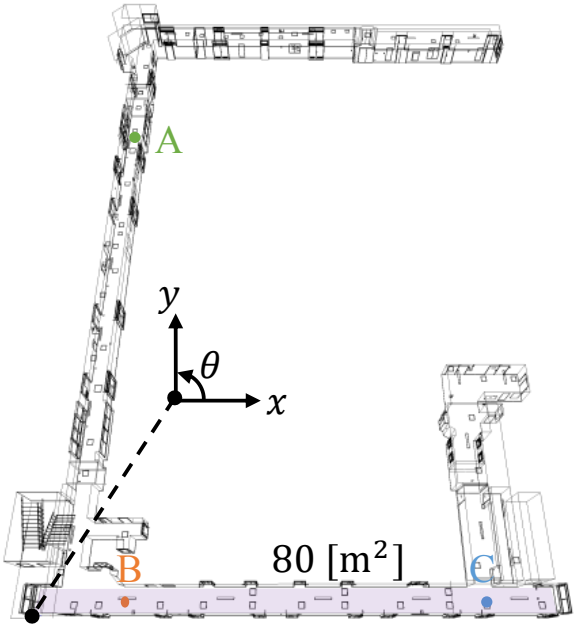


Fig. 4: Experimental environment (4th of D-building in TUT)

TABLE II: Resulted pose error.

d_x, d_y [m]	max error		average error		median error	
	x, y [m]	θ [rad]	x, y [m]	θ [rad]	x, y [m]	θ [rad]
0.1	21.8	0.523	0.553	0.019	0.109	0.018
0.2	23.4	3.14	2.30	0.190	0.134	0.019
0.4	28.3	3.06	1.67	0.159	0.196	0.032

TABLE III: Size of database and processing times per query.

d_x, d_y [m]	No. of templates	Size of database [MB]	Time [ms]
0.1	5,109	2,701	1,419
0.2	1,379	729	250
0.4	396	209	68

TABLE IV: Parameters of robot setting.

	Parameter	Value	
Update rate	VGM request	1	[Hz]
	Odometry update	50	[Hz]
	EKF update	10	[Hz]
VGM	Min percentage of edge pixel	2.50	[%]
	m_{bd}	0.30	[m]
	$m_{b\theta}$	0.15	[rad]
EKF std. dev.	VGM x, y	0.2	[m]
	VGM θ	0.2	[rad]
	Odometer velocity x, y	0.8	[m/s]
	Odometer velocity θ	0.8	[rad/s]
Robot	Max velocity	1.0	[m/s]
	Max acceleration	1.0	[m/s^2]
	Max angular velocity	1.0	[rad/s]
	Max angular acceleration	1.0	[rad/s^2]

based on the predicted covariance: C of the EKF as follows:

$$m_d = m_{bd} + \sqrt{C_{xx} + C_{yy}}, \quad m_\theta \quad = m_{b\theta} + \sqrt{C_{\theta\theta}}. \quad (20)$$

In both experiments, the A* algorithm was used for global path planning and the DWA[12] algorithm for local path planning for autonomous navigation. The parameters for the experiments are described in Table IV.

B. Accuracy Evaluation

The robot traveled through the environment three times following the sequence described in Table V. All data from the experiment, including (odometry, query images, the resulting position of panorama-VGM and LiDAR data) were recorded as rosbag and used for analysis. In this experiment, we applied EMCL[13] to the rosbag data to calculate the

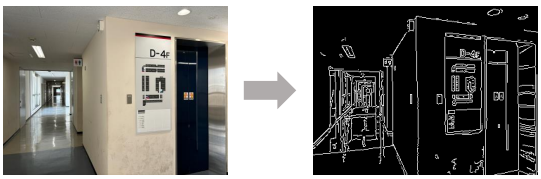
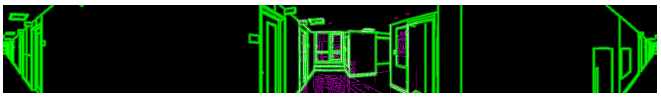
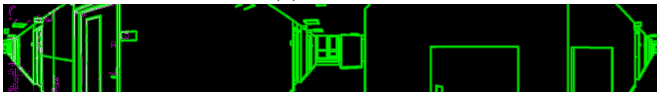


Fig. 5: Canny edge detection.



(a) Success

(b) Failure1



(c) Failure2

Fig. 6: Result images of panorama VGM (d_x, d_y = 0.2[m])

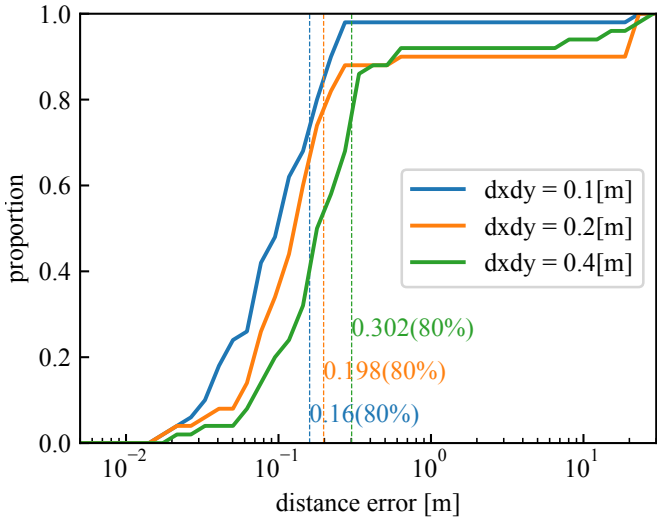


Fig. 7: Cumulative histogram vs distance error.

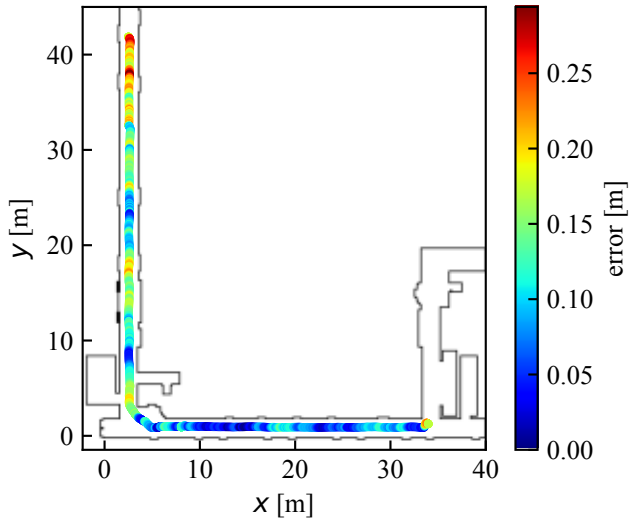


Fig. 9: The error map of trial 1.

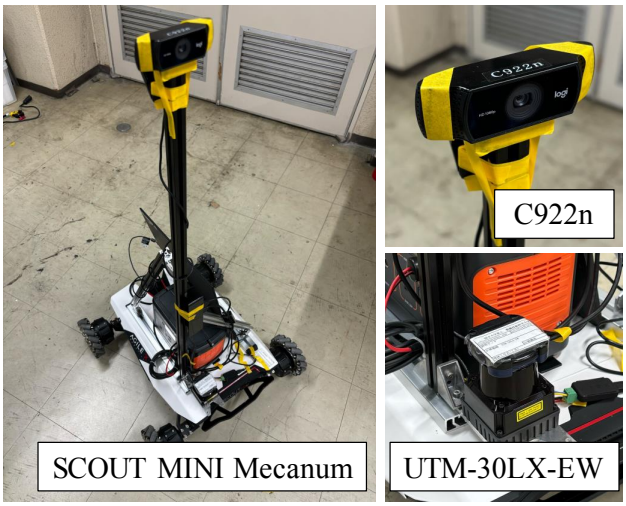


Fig. 8: The robot system for experiment.

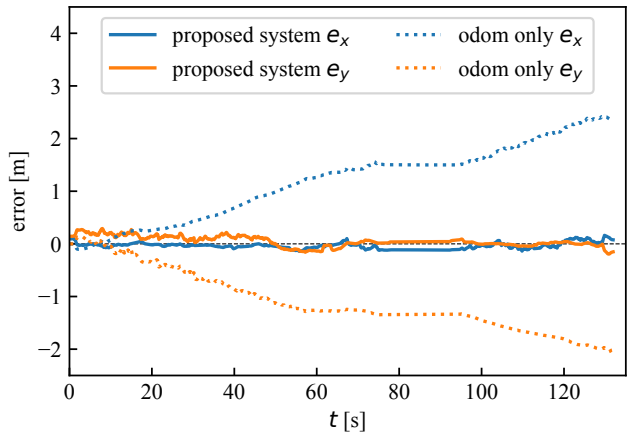


Fig. 10: The comparison of error of proposed system and odometry error in trial 1.

ground truth position because it was more precise than AMCL as discussed in Chapter IV.

Table VI shows the results of the localization error, and Fig. 9 shows the distance error map from trial 1. Fig. 10 shows comparison graphs between localization using only odometry and our proposed method fuses odometry with panorama-VGM. All results indicate that our proposed method corrects the drift error of odometry and validates our proposed system. In addition, we ran ORB-SLAM3[14] with this rosbag, but tracking failed midway through all sequences.

C. Durability Test

This experiment evaluates the durability of our proposed system through the trials where the robot is made to run until it stops for any reasons. Table VII shows the route along with the results for continuous distance and running time achieved through the experiment. The reasons why the robot stopped running were battery loss of onboard computer in trial 4 and 5, and lost the connection of camera due to vibrations from running in trial 6. Additionally, the robot has never hit the walls, nor come close to hitting them. Therefore, it has been confirmed that a small number of wrong outputs of panorama-VGM can be compensated perfectly by EKF and our proposed system is suitable for practical use.

TABLE V: Routes, trip distance and elapsed time.

Trial No.	Route	Trip distance [m]	Time [min]
trial 1	A → C	70	2.2
trial 2	A → C	70	1.8
trial 3	A → C → A	140	3.7

D. Discussion

In fact, the errors described in Table VI represents the difference between the results of LiDAR approach and the re-

TABLE VI: Resulted pose error.

Trial No.	max error		average error		median error	
	x, y [m]	θ [rad]	x, y [m]	θ [rad]	x, y [m]	θ [rad]
trial 1	0.294	0.118	0.117	0.023	0.119	0.020
trial 2	0.808	0.166	0.119	0.025	0.099	0.019
trial 3	0.928	0.140	0.141	0.021	0.107	0.018

sults of panorama-VGM approach, and the small error means that the performance of both approaches are equivalent. Moreover, our proposed approach achieved over one hour of continuous running. The panorama-VGM only requires a camera for sensing and is better in terms of volumetric cost, weight cost, energy cost, computational cost, and price cost than those of LiDAR approach.

The effect of low-price cost becomes noticeable when considering multiple robots operation. As shown in Table VIII, the processing time of panorama-VGM on the server machine is less than 10 [ms] per query, and the robot works well with 1 [Hz] VGM request. Therefore, it can be said that one server machine simultaneously handles 100 number of robots. The equipments used in this project was as follows: server PC: $4,433, camera: $65, and LiDAR: $2,353. Assuming the use of 10 robots application, our approach costs only $5,083 for robot navigation, while the LiDAR approach costs $23,530. This means our proposed panorama-VGM based localization offers cost-effective scalability.

VI. Summary

We proposed a novel localization approach named "panorama-VGM" which requires only a monocular camera on robot by implementing it into a server-client system. The query image and template image are projected onto a cylindrical panorama surface and matched there, significantly reducing memory usage compared to the basic VGM method and improving localization accuracy. The median error for both single performance evaluation and of robot localization results are around 0.1 [m]. Moreover, the robot achieved continuous running for 2,590 [m] in 77 [min] during the durability test.

The panorama-VGM can work on textureless environment, such as a building of university, hospital, so on, unlike the visual-odometry approach using image features. Few approach has existed to make a robot running in textureless environment without LiDAR sensor before. We are confident that the panorama-VGM becomes mainstream in robot localization in the near future. Future works include parameter tuning of panorama-VGM to improve the performance, and conducting experiments with multiple robots.

TABLE VII: Routes and resulted distance and elapsed time.

trial No.	Route	Cycles	Trip distance [m]	Time [min]
trial 4	B $\leftrightarrow$ C	25	1350	42
trial 5	B $\leftrightarrow$ C	48	2590	77
trial 6	A $\leftrightarrow$ C	11	1540	48

TABLE VIII: Resulted number of total query images and server processing times per query.

trial No.	Number of total query	Time [ms]
trial 4	2367	9.96
trial 5	4232	9.63
trial 6	2625	9.55

References

[1] C. Cadena *et al.*, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Transactions on Robotics*, vol. 32, no. 6, pp. 1309–1332, 2016.

[2] F. Dellaert *et al.*, "Monte carlo localization for mobile robots," in *Proc. of 1999 IEEE Int. Conf. on robotics and automation*, IEEE, vol. 2, 1999, pp. 1322–1328.

[3] A. Merzlyakov *et al.*, "A comparison of modern general-purpose visual slam approaches," in *Proc. of 2021 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2021, pp. 9190–9197.

[4] S. Yang *et al.*, "Pop-up slam: Semantic monocular plane slam for low-texture environments," in *Proc. of 2016 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, IEEE, 2016, pp. 1222–1229.

[5] S. Murali *et al.*, "Indoor scan2bim: Building information models of house interiors," in *Proc. of 2017 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, IEEE, 2017, pp. 6126–6133.

[6] R. Hendrikx *et al.*, "Connecting semantic building information models and robotics: An application to 2d lidar-based localization," in *Proc. of 2021 IEEE Int. Conf. on Robotics and Automation (ICRA)*, IEEE, 2021, pp. 11 654–11 660.

[7] D. Acharya *et al.*, "Bim-posenet: Indoor camera localisation using a 3d indoor model and deep learning from synthetic images," *ISPRS journal of photogrammetry and remote sensing*, vol. 150, pp. 245–258, 2019.

[8] J. Chen *et al.*, "Align to locate: Registering photogrammetric point clouds to bim for robust indoor localization," *Building and Environment*, vol. 209, p. 108 675, 2022.

[9] J. Takahashi *et al.*, "Image-retrieval method using gradient dilation images for cloud-based positioning system with 3d wireframe map.," *Sensors & Materials*, vol. 32, 2020.

[10] B. D. Anderson *et al.*, *Optimal filtering*. Courier Corporation, 2012.

[11] S. Kato *et al.*, "An open approach to autonomous vehicles," *IEEE Micro*, vol. 35, no. 6, pp. 60–68, 2015.

[12] D. Fox *et al.*, "The dynamic window approach to collision avoidance," *IEEE Robotics & Automation Magazine*, vol. 4, no. 1, pp. 23–33, 1997.

[13] R. Ueda *et al.*, "Expansion resetting for recovery from fatal error in monte carlo localization-comparison with sensor resetting methods," in *Proc. of 2004 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, IEEE, vol. 3, 2004, pp. 2481–2486.

[14] C. Campos *et al.*, "Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam," *IEEE Transactions on Robotics*, vol. 37, no. 6, pp. 1874–1890, 2021.