2026 IEEE/SICE International Symposium on System Integration (SII)
January 11-14, 2026. Cancún, México

Topological Mapping with Constrained Optimization based on Visual Place Recognition and Orientation Constraints

Takaya Nakao *

* Graduate School of Science and Technology, Meiji University, Kanagawa, Japan

Yoshitaka Hara †

† Future Robotics Technology Center (fuRo), Chiba Institute of Technology, Chiba, Japan

Yoji Kuroda ‡

‡ School of Science and Technology, Meiji University, Kanagawa, Japan

***Abstract*— This paper proposes a method for topological mapping that guarantees orientation accuracy. Unlike pure topological maps, the orientations of arcs connecting nodes in our topological maps match those in real environments. The proposed method uses the Visual Place Recognition (VPR) method, AnyLoc-VLAD-DINOv2, to extract global descriptors from images obtained by a 360-degree camera, and then creates nodes and detects loops based on these descriptors. Furthermore, the initial orientation of each node is calculated using angular velocity obtained by an IMU. The proposed method creates two types of orientation constraints. Heading constraints are created from initial orientations, and loop constraints are created from loop detection. Subsequently, node orientations and arc lengths are corrected through constrained optimization with two types of orientation constraints. Through indoor and outdoor experiments, the proposed method enabled topological mapping with both topological consistency and orientation accuracy. Nodes of the topological maps were created adaptively based on the appearance of each location within the environments, and the node spacing varied accordingly. Furthermore, through constrained optimization with two types of orientation constraints, node loops were closed and arc orientations matched those in real environments. Even in environments containing multiple loops, the proposed method enabled topological mapping while simultaneously satisfying the constraint of each loop.**

I. Introduction

For robot navigation, there are metric maps and topological maps. Metric maps represent the shape and spatial relationships of environments in a Cartesian coordinate system. Topological maps abstractly represent the connectivity of environments using a graph structure composed of nodes and arcs.

Our previous work [1] proposed a method for topological mapping that maintains topological consistency between real environments and topological maps. The topological maps of our previous method, without using metric information, are pure topological maps, and the connectivity of nodes topologically matches that of real environments. Our previous method extracts features using AnyLoc [2], a highly versatile Visual Place Recognition (VPR) method, from images obtained by a 360-degree camera. Based on their similarity, nodes are adaptively created and loops are detected.

This paper proposes a novel method for topological mapping that guarantees orientation accuracy between real environments and topological maps. In the topological maps of the proposed method, the orientations of arcs connecting nodes actually match those in real environments. At first, the proposed method creates nodes and detects loops based on VPR from images obtained by a 360-degree camera, and calculates the initial orientation of each node from angular velocity obtained by an IMU. Next, two types of orientation constraints are created based on loop detection and the initial orientation of each node. Then, thorough constrained optimization with two types of orientation constraints, node orientations and arc lengths are corrected.

The contributions of this paper are as follows:

- Using VPR, nodes are created and loops are detected from images obtained by a 360-degree camera, and the initial orientation of each node is calculated from angular velocity obtained by an IMU.
- Through constrained optimization with two types of orientation constraints, node orientations and arc lengths are corrected for topological mapping with both topological consistency and orientation accuracy.
- From experiments in both indoor and outdoor environments with single or multiple loops, we confirm that, using constrained optimization, node loops are closed and arc orientations match those in real environments.

II. Related Work

A. Topological Maps

Topological maps abstractly represent environments as graph structures. This type of map representation is known for its robustness to environmental changes and computational efficiency, making it effective for navigation [3] [4] [5].

Furthermore, the existence of cognitive maps based on topological structures has been suggested in human spatial cognition, and their applications in robotics are also advancing [6] [7] [8].

Thrun [5] proposed a hybrid map that has both the precision of metric maps and the computational efficiency of topological maps. This method first constructs a metric map for the entire environment, and then constructs a topological map from its geometry.

Kuipers et al. [6] introduced the concept of cognitive maps into robotic spatial representation. They proposed a hierarchical representation called the Spatial Semantic Hierarchy

978-1-6654-5784-2/26/$31.00 ©2026 IEEE

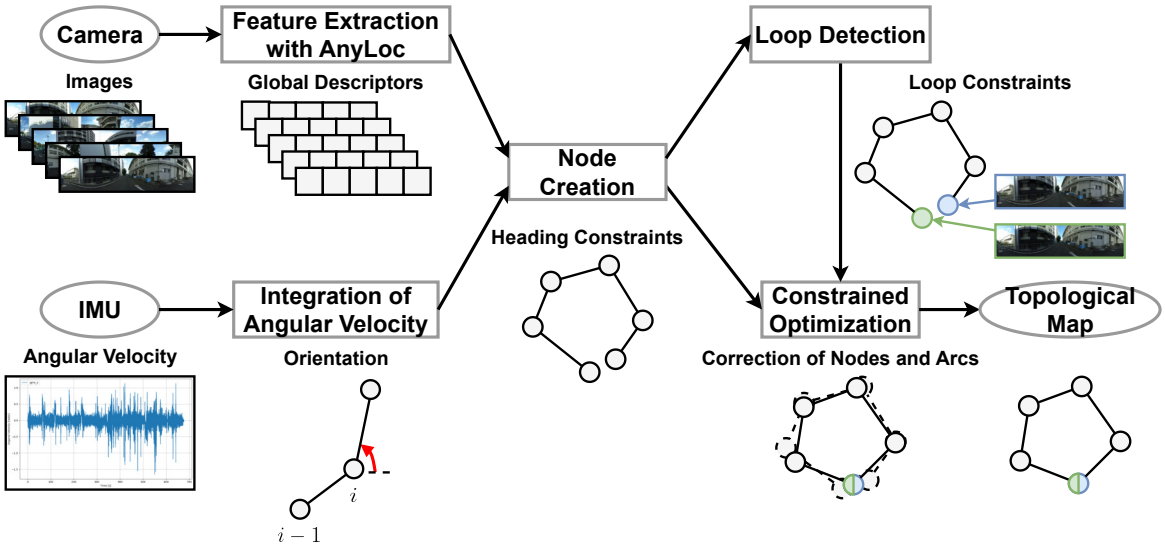


Fig. 1. Pipeline of the proposed topological mapping method using constrained optimization.

(SSH), which captures environmental structure at multiple levels of abstraction.

Choset et al. [7] proposed topological SLAM for simultaneous localization and mapping based on environmental connectivity. They achieved efficient SLAM using a Generalized Voronoi Graph (GVG) to represent the connectivity of the environment.

Konolige et al. [8] proposed a navigation method using metric-topological maps. This method constructs a metric-topological map by building metric local maps of small areas and topologically connecting these local maps. This shows that navigation is possible even without detailed geometric information of the entire map.

These conventional methods use topological maps to capture the connectivity of the environment as a graph structure of nodes and arcs, enabling localization and path planning. In contrast, the proposed method is a novel approach that constructs a topological map in which the orientations of arcs connecting nodes match those in real environments using a camera and an IMU.

B. Visual Place Recognition

Visual Place Recognition (VPR) is a method that recognizes previously visited locations using images, performing image retrieval based on similarities between images. Robust feature representations are required to correctly recognize the same location under diverse conditions, such as viewpoints, lighting, and seasonal changes [9].

Classical VPR methods extended image retrieval tasks by representing local descriptors, such as SIFT [10] and SURF [11], as feature vectors using Bag of Visual Words (BoVW) [12] or VLAD [13]. Subsequently, learning-based methods such as NetVLAD [14], GeM [15], and CosPlace [16] emerged and achieved high performance.

AnyLoc [2] is a high performance VPR method. AnyLoc leverages general purpose feature representations of self-supervised learning models, such as CLIP [17] and DINOv2 [18], achieving high VPR performance across diverse environments without retraining or fine-tuning.

Fig. 2. Feature extraction using AnyLoc-VLAD-DINOv2.

In this paper, we use AnyLoc for node creation and loop detection. Using AnyLoc solves the limitations of conventional topological mapping, such as dataset dependency and environmental conditions.

III. Topological Mapping

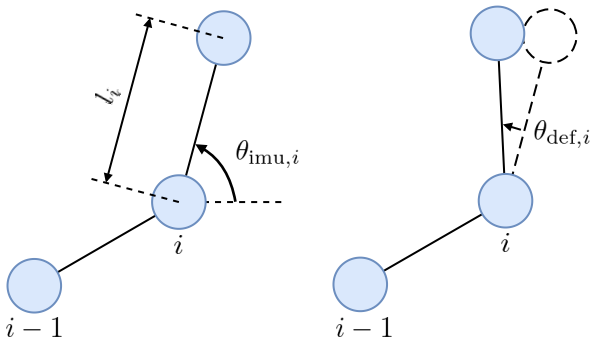
A. Overview

Fig. 1 shows the proposed method pipeline. Compared to our previous method [1], the proposed method adds the creation of orientation constraints and constrained optimization. In addition to images obtained by a 360-degree camera, the proposed method utilizes angular velocity obtained by an IMU.

The proposed method enables topological mapping that achieves both topological consistency and orientation accuracy. First, global descriptors are extracted from images to create nodes and detect loops. Furthermore, the initial orientation of each node is calculated using angular velocity. Heading constraints are created from initial orientations, and loop constraints are created from loop detection. Then, node orientations and arc lengths are corrected through constrained optimization with two types of orientation constraints.

B. Node Creation and Loop Detection using VPR

Fig. 2 shows the process of extracting features from images obtained by a 360-degree camera. The proposed



(a) Before Optimization (b) After Optimization

Fig. 3. Node configurations before and after optimization.

method uses AnyLoc-VLAD-DINOv2 [2]. First, DINOv2 [18] is used to obtain local descriptors from images. Next, the local descriptors are aggregated into global descriptors using VLAD [13].

The proposed method creates nodes and detects loops in the topological map based on global descriptors extracted from images. Images are sampled at regular intervals from collected data, and global descriptors are extracted from each image. For node creation, a threshold is set based on the cosine similarity of descriptors, and nodes are added adaptively. Loop detection is also based on cosine similarity.

C. Heading Constraints and Loop Constraints

The proposed method creates two types of orientation constraints: heading constraint and loop constraints. Heading constraints correct node orientations to align with initial orientations obtained by an IMU. Loop constraints correct entire loops so that detected nodes overlap and loops close.

First, as show in Fig. 1, heading constraints are created. Angular velocity obtained by an IMU is integrated to calculate the initial orientation of each node relative to the beginning node.

Next, loop constraints are created. Angular velocity obtained by an IMU contains errors, and arc lengths are topological and indeterminate. As a result, loops cannot be correctly closed using only heading constraints. Therefore, loop constraints correct node orientations and arc lengths within detected loops so that detected nodes overlap and loops close. These constraints maintain both topological consistency and orientation accuracy.

D. Constrained Optimization

Fig. 3 shows node configurations before and after optimization. The proposed method corrects node orientations and arc lengths through constrained optimization with heading constraints and loop constraints.

The optimized orientation of each node $\theta_{\mathrm{map},i}$ is defined as shown in Eq. (1).

$$\theta_{\mathrm{map},i} \coloneqq \theta_{\mathrm{imu},i} + \theta_{\mathrm{def},i} \tag{1}$$

It is determined by adding the correction angle $\theta_{\mathrm{def},i}$, estimated through optimization, to the initial orientation $\theta_{\mathrm{imu},i}$ obtained from an IMU.

The optimization uses the loop set $\mathcal{L}$, as shown in Eq.(2), which is selected from the topological map. Each loop $\mathcal{L}_k$, as shown in Eq.(3), is defined by the indices from the start node $i_{\mathrm{s\_k}}$ to the end node $i_{\mathrm{e\_k}}$.

$$\mathcal{L} \coloneqq \{\mathcal{L}_1, \mathcal{L}_2, \ldots, \mathcal{L}_K\} \tag{2}$$

$$\mathcal{L}_k \coloneqq \{i_{\mathrm{s\_k}}, \ldots, i_{\mathrm{e\_k}}\} \tag{3}$$

The proposed method performs constrained optimization for heading constraints and loop constraints. If multiple conflicting loop constraints exist in a loop, the resulting topological map becomes inconsistent. Therefore, loop selection for constrained optimization is performed as follows. First, based on loop detection using VPR, the graph structure of the topological map is obtained. Next, the cycle_basis function in the NetworkX library is used to extract representative loops from the graph. Then, from these extracted loops, loops containing a certain number of nodes are selected for constrained optimization.

The problem of simultaneously optimizing the correction angles $\boldsymbol{\theta}_{\mathrm{def}}$ and the arc lengths $\boldsymbol{l}$ is formulated as in Eqs. (4), (5), and (6).

$$\hat{\boldsymbol{\theta}}_{\mathrm{def}}, \hat{\boldsymbol{l}} = \underset{\boldsymbol{\theta}_{\mathrm{def}}, \boldsymbol{l}}{\operatorname{argmin}} \sum_{i=0}^{N-1} (\theta_{\mathrm{def},i})^2 \tag{4}$$

$$\text{subject to} \sum_{i \in \mathcal{L}_k} l_i \begin{bmatrix} \cos(\theta_{\mathrm{map},i}) \\ \sin(\theta_{\mathrm{map},i}) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \quad \forall\ k = 1, \ldots, K \tag{5}$$

$$l_i \geq 1, \quad \forall\ i = 0, \ldots, N-1 \tag{6}$$

The objective function in Eq. (4) minimizes the sum of squares of the correction angles $\boldsymbol{\theta}_{\mathrm{def}}$. This minimizes the amount of correction required to close the loops. This corresponds to the heading constraints.

The constraint in Eq. (5) is to close the loops. By constraining the sum of the arc vectors in each loop to be the zero vector, it guarantees that the start node and end node of the loop overlap. This corresponds to the loop constraints. Furthermore, in Eq. (6), the minimum value of the arc lengths $\boldsymbol{l}$ is set to 1 to prevent excessive contraction of the arcs.

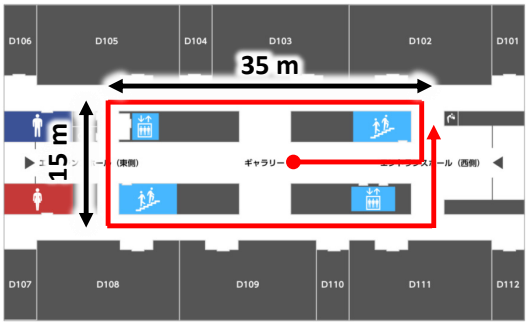
As described above, the proposed method solves the constrained optimization problem. For the optimization implementation, we use the optimize submodule of the SciPy library. Our method adopt sequential quadratic programming (SLSQP) algorithm.

IV. Experiments

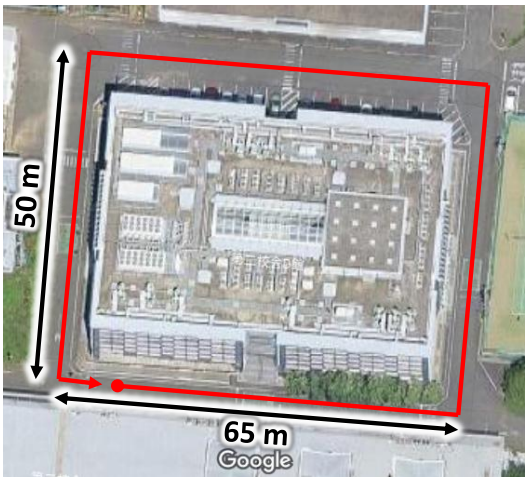
A. Setup

Fig. 4 shows the experimental environments and the paths to collect data, where the paths are shown in red. Fig. 5 shows examples of images obtained in each environment.

To verify the effectiveness of the proposed method, we qualitatively evaluated the topological maps. In the indoor environment and in the outdoor environment 1, a single loop path was used. In contrast, in the outdoor environment 2, a figure-eight loop path was used for multiple loops.



(a) Indoor Environment



(b) Outdoor Environment 1

(c) Outdoor Environment 2

Fig. 4. Experimental environments. Data was collected on both indoor and outdoor paths at Meiji University.

Data was collected using a wheeled robot based on the WHILL Model CR, at a speed of 1 m/s. The sensors were a Ricoh Theta S 360-degree camera and an Xsens MTi-30 IMU. To process the collected data, a PC equipped with CPU: Intel Core i9 14900KF, RAM: 64 GB, GPU: NVIDIA GeForce RTX 4090, and VRAM: 24 GB was used.

B. Evaluation

Fig. 6, Fig. 8, and Fig. 10 show the node positions created by the proposed method on the metric maps for each environment. The start point is shown in blue, and the end point is shown in green. For outdoor environment 2, the proposed method detected two loops for constrained optimization. Loop 1 is drawn in green, and Loop 2 is drawn in blue.

TABLE I

COMPARISON OF TRAVEL TIME, NUMBER OF IMAGES, MAPPING TIME, AND NUMBER OF NODES IN EACH ENVIRONMENT

	Indoor	**Outdoor 1**	**Outdoor 2**
Travel Time	2 min 33 s	4 min 34 s	11 min 7 s
Number of Images	153	273	667
Mapping Time	14.692 s	27.393 s	50.831 s
Number of Nodes	54	33	75

(a) Indoor Environment

(b) Outdoor Environment

Fig. 5. Example images obtained in each environment.

These figures indicate that nodes are created adaptively based on the appearance of each location within the environments, and the node spacing varies accordingly. Nodes are created when global descriptors change beyond a predefined threshold, enabling adaptive adjustment of node spacing.

In the indoor environment, as shown in Fig. 5(a), the appearance is monotonous and consists primarily of white structures. As a result, the appearance changes uniformly at every location, leading to node creation at regular intervals, as shown in Fig. 6.

In contrast, in the outdoor environment, as shown in Fig. 5(b), the appearance is complex and includes diverse elements such as buildings and trees. As a result, the appearance changes significantly depending on location, leading to node creation at varying intervals, as shown in Fig. 8 and Fig. 10.

Table I shows the comparison of travel time, number of images, mapping time, and number of nodes in each environment. This table indicates that the proposed method can process mapping in a time sufficiently shorter than the travel time. The computational cost of the proposed method is strongly influenced by node creation based on VPR and increases with the number of images. The number of nodes in the map is determined adaptively based on the appearance in each environment, so it is not directly related to the number of images.

Fig. 7, Fig. 9, and Fig. 11 show the topological maps of our previous method [1] and the proposed method. In these figures, the start and end nodes of loops are shown in green.

As shown in Fig. 7(a), Fig. 9(a), and Fig. 11(a), loops were

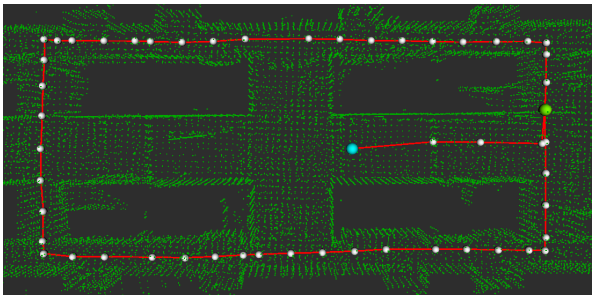
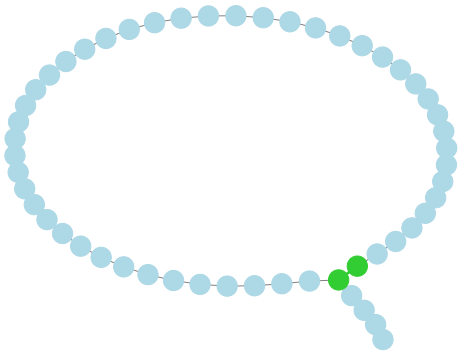
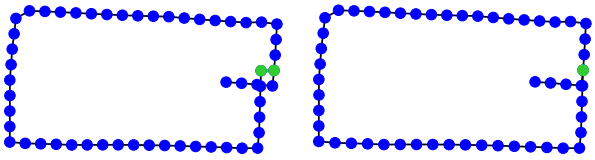


Fig. 6. Node positions on the metric map in the indoor environment.



(a) Previous Method [1]



(b) Ours (Before Optimization) (c) Ours (After Optimization)

Fig. 7. Topological map of the indoor environment.

closed correctly by loop detection of our previous method [1], enabling topological mapping that maintains topological consistency between real environments and topological maps. However, our previous method [1] does not impose orientation constraints, resulting in the complete loss of path shape information such as straight passages and corners. Therefore, all loop paths are represented as simple circular loops.

In contrast, the proposed method optimizes with two types of orientation constraints, correcting node orientations and arc lengths. Consequently, as shown in Fig. 7(b)(c), Fig. 9(b)(c), and Fig. 11(b)(c), our topological maps achieve both topological consistency and orientation accuracy. Note that, since the proposed method performs topological mapping, arc lengths are deformed and differ from those in the metric maps.

In the environment shown in Fig. 6, nodes and arcs extend inside the loop. As shown in Fig. 7(a), our previous method [1] maintains topological consistency but has nodes and arcs extending outside the loop. In contrast, as shown in Fig. 7(b)(c), the proposed method correctly positions nodes and arcs inside the loop.

In the proposed method before optimization, as shown in Fig. 7(b), Fig. 9(b), and Fig. 11(b), nodes and arcs are

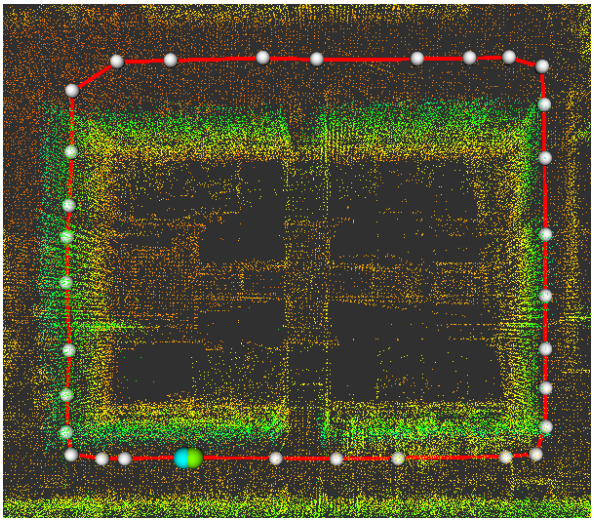
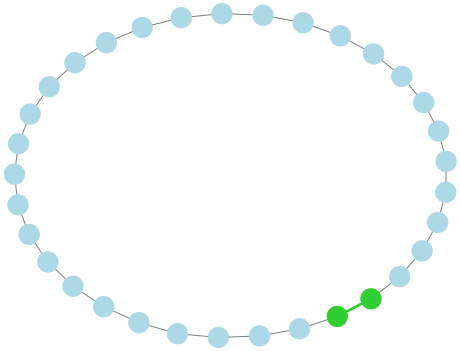
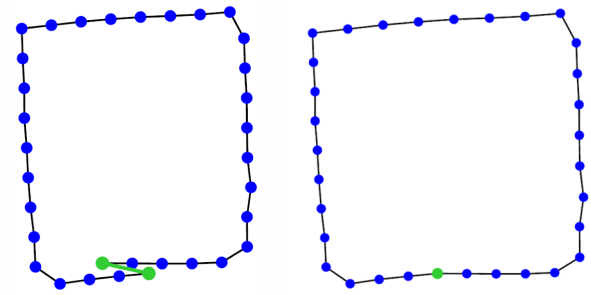


Fig. 8. Node positions on the metric map in the outdoor environment 1.



(a) Previous Method [1]



(b) Ours (Before Optimization) (c) Ours (After Optimization)

Fig. 9. Topological map of the outdoor environment 1.

positioned based on the initial orientation obtained by the IMU. Since the IMU orientation contains errors and the arc lengths are topological and indeterminate, the loops are not closed and topological consistency is not maintained.

On the other hand, in the proposed method after optimization, as shown in Fig. 7(c), Fig. 9(c), and Fig. 11(c), the node orientations and arc lengths are corrected. Constrained optimization using two types of orientation constraints closes the loops, achieving both topological consistency and orientation accuracy.

As shown in Fig. 11(c), the proposed method can close the loops even when multiple loops exist. Since our method

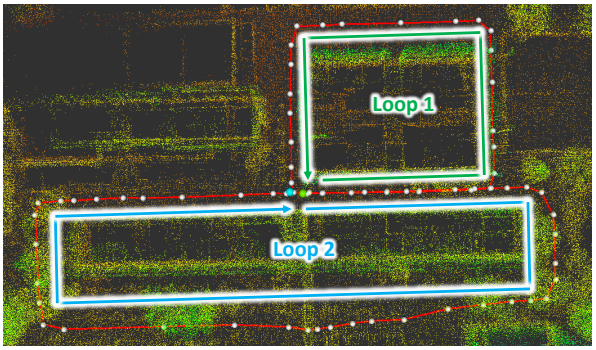
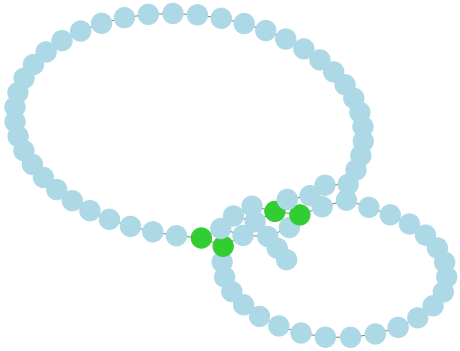
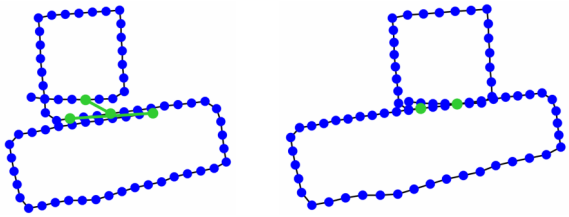


Fig. 10. Node positions on the metric map in the outdoor environment 2.



(a) Previous Method [1]



(b) Ours (Before Optimization) (c) Ours (After Optimization)

Fig. 11. Topological map of the outdoor environment 2.

employs constrained optimization, if multiple conflicting loop constraints exist, the resulting topological map becomes inconsistent. The proposed method selects appropriate loops to impose loop constraints for constrained optimization. Thus, consistent topological mapping is achieved even with multiple loops.

V. Conclusion

This paper proposed a novel method for topological mapping that guarantees orientation accuracy between real environments and topological maps. In the topological maps of the proposed method, the orientations of arcs connecting nodes actually match those in real environments.

Through indoor and outdoor experiments, the proposed method enabled topological mapping with both topological consistency and orientation accuracy. Nodes of the topological maps were created adaptively based on the appearance of each location within the environments, and the node spacing varied accordingly. Furthermore, through constrained optimization with two types of orientation constraints, node loops were closed and arc orientations matched those in real environments. Even in environments containing multiple loops, the proposed method enabled topological mapping while simultaneously satisfying the constraint of each loop.

References

[1] Daichi Ogiwara, Naoki Takahashi, Kosuke Kimizuka, Takaya Nakao, Yoshitaka Hara, and Yoji Kuroda, "Topological Mapping using Visual Place Recognition," *Proc. of Robotics Symposia (ROBOSYM)*, 2025. (in Japanese)

[2] Nikhil Keetha, Avneesh Mishra, Jay Karhade, Krishna Murthy Jatavallabhula, Sebastian Scherer, Madhava Krishna, and Sourav Garg, "AnyLoc: Towards Universal Visual Place Recognition," *IEEE Robotics and Automation Letters*, vol. 9, no. 2, pp. 1286–1293, 2023.

[3] Benjamin Kuipers, "Modeling Spatial Knowledge," *Cognitive Science*, vol. 2, no. 2, pp. 129–153, 1978.

[4] Benjamin J. Kuipers and Todd S. Levitt, "Navigation and Mapping in Large Scale Space," *AI Magazine*, vol. 9, no. 2, pp. 25–25, 1988.

[5] Sebastian Thrun, "Learning Metric-Topological Maps for Indoor Mobile Robot Navigation," *Artificial Intelligence*, vol. 99, no. 1, pp. 21–71, 1998.

[6] Benjamin Kuipers, "The Spatial Semantic Hierarchy," *Artificial Intelligence*, vol. 119, no. 1–2, pp. 191–233, 2000.

[7] Howie Choset and Keiji Nagatani, "Topological Simultaneous Localization and Mapping (SLAM): Toward Exact Localization without Explicit Localization," *IEEE Trans. on Robotics and Automation*, vol. 17, no. 2, pp. 125–137, 2001.

[8] Kurt Konolige, Eitan Marder-Eppstein, and Bhaskara Marthi, "Navigation in Hybrid Metric-Topological Maps," *Proc. of IEEE Int. Conf. on Robotics and Automation (ICRA)*, 2011.

[9] Carlo Masone and Barbara Caputo, "A Survey on Deep Visual Place Recognition," *IEEE Access*, vol. 9, pp. 19516–19547, 2021.

[10] David G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int. J. of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[11] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool, "SURF: Speeded Up Robust Features," *Proc. of European Conf. on Computer Vision (ECCV)*, 2006.

[12] James Philbin, Ondrej Chum, Michael Isard, Josef Sivic, and Andrew Zisserman, "Object Retrieval with Large Vocabularies and Fast Spatial Matching," *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[13] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez, "Aggregating Local Descriptors into a Compact Image Representation," *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[14] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic, "NetVLAD: CNN Architecture for Weakly Supervised Place Recognition," *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[15] Filip Radenović, Giorgos Tolias, and Ondřej Chum, "Fine-Tuning CNN Image Retrieval with No Human Annotation," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 41, no. 7, pp. 1655–1668, 2018.

[16] Gabriele Berton, Carlo Masone, and Barbara Caputo, "Rethinking Visual Geo-Localization for Large-Scale Applications," *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2022.

[17] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, *et al.*, "Learning Transferable Visual Models from Natural Language Supervision," *Proc. of Int. Conf. on Machine Learning (ICML)*, 2021.

[18] Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy V. Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, *et al.*, "DINOv2: Learning Robust Visual Features without Supervision," *Trans. on Machine Learning Research*, pp. 1–32, 2024.