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

Configuration Supporting System of Intelligent Space Based on Inherited Automatic Pose Estimation for Distributed RGB-D Cameras

Shunsuke Iwasaki[1] and Kazuyuki Morioka[2]

***Abstract*— This study proposes an automatic pose estimation method for distributed RGB-D cameras aimed at supporting the configuration of intelligent spaces. The proposed method is based on the independent pose estimation of each camera, that is effective for scalability of intelligent space configuration. The distributed camera poses are automatically estimated by utilizing the extrinsic parameters of already aligned cameras and the point cloud alignments among the adjacent cameras and the 3D map. This approach significantly reduces the workload required for building intelligent spaces. Furthermore, a person tracking system is constructed as the application, demonstrating the practical usability of the proposed method.**

I. Introduction

The intelligent space is a spatial robotic system including networked sensors placed in the wide space and actual robot systems[1]. These systems aim to accurately recognize the movements of people and the other objects within the space and provide several services based on the recognized information. For example, the mobile robot service for human support was achieved as one of the practical applications of the intelligent space based on networked intelligent cameras[2]. In recent years, many types of systems with similar concepts as the intelligent space have been increasingly implemented in several urban environments. Especially, one of the core technologies constituting intelligent spaces is multi-camera person tracking. Person tracking is indispensable for precisely grasping and analyzing spatial situations.

To build a person tracking system using multiple cameras, it is essential to accurately acquire the pose of each distributed camera in the world coordinate system. This enables the integration of person positions acquired by each camera into a common world coordinate system, allowing continuous tracking throughout the entire space. Traditionally, the acquisition of camera poses has relied on marker-based methods which require manual adjustments. However, iterating these processes for many distributed cameras is complicated and time-consuming. Also, parameter accuracy depends heavily on the operator's experience and skill, often resulting in variability. Furthermore, the same procedure must be performed whenever cameras are newly added or repositioned. This feature presents a significant bottleneck for large-scale system deployment.

[1]Meiji Univ, Graduation School of Advanced Mathematical Sciences, Network Design Program, Japan email: cs243002@meiji.ac.jp
[2]Meiji Univ, School of Interdisciplinary Mathematical Sciences, Japan email: morioka@meiji.ac.jp

This study aims to realize a user-friendly system for configuring intelligent spaces by automating camera pose estimation processes, as shown in Fig.1. This method can build intelligent spaces without depending on the user's skills. This paper particularly describes the person tracking application as a core element of intelligent spaces based on auto-acquisition of camera poses.

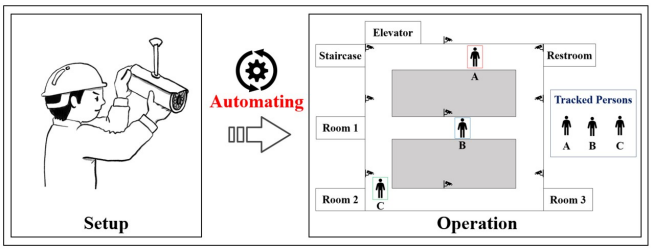

Fig. 1. Research objective

For this purpose, this study proposes an auto pose estimation method of RGB-D cameras distributed in the wide target space. Specifically, the proposed method is based on point cloud alignment between the 3D map and each camera, using inherited camera poses from the adjacent cameras. This paper presents the details of the proposed method and demonstrates its effectiveness through experiments in the actual setup of the intelligent space.

II. Related Works

In multi-camera systems, accurate pose estimation of each camera is crucial for building intelligent spaces in a consistent world coordinate system across the entire space. Traditionally, this task has been addressed mainly using marker-based methods. These approaches employ artificial markers, such as checkerboards or AprilTags, to establish geometric correspondences between cameras and estimate camera poses through global optimization [3][4]. However, such methods require precise placement and adjustment of the markers, which can be particularly burdensome when deploying a large number of cameras in wide environments.

A method proposed by Liu et al. extracts natural features from walls, floors, and other structures in RGB images and combines them with depth information to determine camera poses [5]. Yoon et al. propose a method that detects the positions and poses of specific objects in the environment and establishes correspondences across different sensors, enabling markerless calibration [6]. Additionally, methods utilizing human motion, such as trajectories and postures, have been proposed to estimate camera poses [7][8].

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

Furthermore, methods utilizing a 3D map have been proposed to estimate the extrinsic parameters of cameras with non-overlapping fields of view [9][10][11]. Particularly in recent years, camera calibration systems in such non-overlapping multi-camera configurations have been focused on. However, systems that rely on non-overlapping distributed cameras inevitably generate blind spots, which present significant challenges in practical applications. For example, in person tracking systems, it is necessary to incorporate prediction or individual identification to match the same person across cameras. These challenges strongly indicate the need for calibration methods that are easy to deploy, scalable to large camera networks, and capable of ensuring accuracy sufficient for real-world use.

Our proposed approach belongs to the field of markerless methods that utilize a 3D map, targeting distributed cameras with overlapping fields of view. Instead of performing global optimization in estimating camera poses, the proposed method assigns a suitable initial parameter for searching an optimal pose to each camera and individually aligns each point cloud to the 3D map. This approach reduces computational load while allowing flexible expansion of the camera network. Moreover, since our method relies solely on point cloud data, it is robust to external factors such as lighting, ensuring stable operation.

III. Automatic Pose Acquisition for Distributed RGB-D Cameras

A. Overview of the Proposed Method

This study assumes building an intelligent space with widely distributed RGB-D cameras. A method for automatically estimating the poses of RGB-D cameras within a space is proposed for easy building of the intelligent space. The method is based on aligning point clouds captured from the cameras with a pre-constructed 3D map of the target space. The proposed method can adopt a markerless approach, eliminating the need for physical markers such as checkerboards, by utilizing depth data of RGB-D cameras. The overview of the entire procedure is shown in Fig.2.

In this chapter, we first explain the method for constructing a 3D map of the target space to be used as a reference. Next, we discuss the point cloud alignment method and the challenges associated with it. Finally, we introduce an inherited automatic search method for estimating the poses of multiple RGB-D cameras, along with a detailed description of its implementation procedure.

B. Building a 3D Map

The proposed method needs a pre-constructed 3D map of the target space as a reference. This map is built using a 3D LiDAR sensor (Velodyne HDL-32E) and a Graph SLAM method called hdl_graph_slam [12]. The 3D map generated by Graph SLAM is represented as a graph structure, where the sensor's trajectory is composed of nodes and edges. Point clouds captured at multiple locations are integrated to reconstruct the full 3D structure of the target space. This style of the 3D map is common for mobile robot navigation. Also,

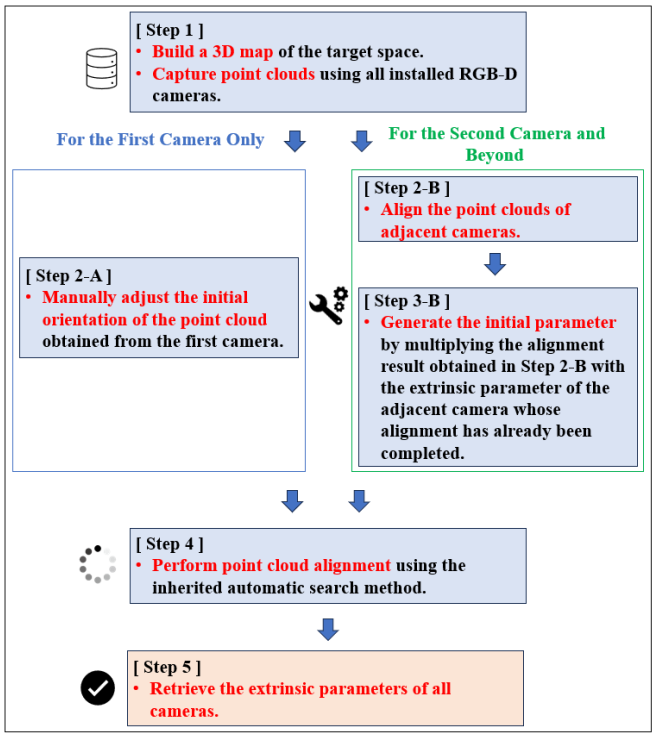

Fig. 2. Workflow of the proposed method

since the map is mainly composed of static objects such as walls and furniture in the target space, the map reconstruction is not needed as long as the space configuration doesn't change significantly. An example of the generated 3D map is shown in Fig. 3. The resulting 3D map serves as a highly accurate reference point cloud for alignment with point clouds obtained from the viewpoint of each RGB-D camera installed within the target space.

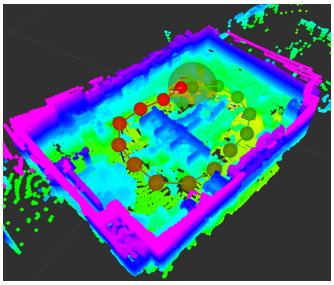

Fig. 3. Example of the 3D map

C. Point Cloud Alignment and Its Challenges

The pose of the camera can be represented as the extrinsic camera parameter in the world coordinate system as a transformation matrix. The pose estimation is equivalent with the extrinsic parameter estimation, that is performed by aligning the point cloud acquired from the RGB-D camera's viewpoint with a reference point cloud of the 3D map. This study adopts the ICP (Iterative Closest Point) algorithm, which is widely used to improve the consistency of 3D point clouds, for this alignment process.

ICP matching can achieve highly accurate alignment when the initial parameters are properly provided. However, the

inappropriate initial parameters may converge to incorrect alignment by locally optimal solution. In particular, the initial orientation setting in parameters is critical. The incorrect initial orientation often fails to achieve optimal alignment even when expanding the search range of ICP. The left image in Fig. 4 shows an example of failed alignment due to insufficient initial orientation adjustment. On the other hand, when the initial orientation is properly adjusted, as shown in the right image of Fig. 4, the point clouds of the desk and walls accurately match the corresponding point clouds of the 3D map of the target space. This indicates that initial orientation adjustment has a significant impact on the success of the alignment.

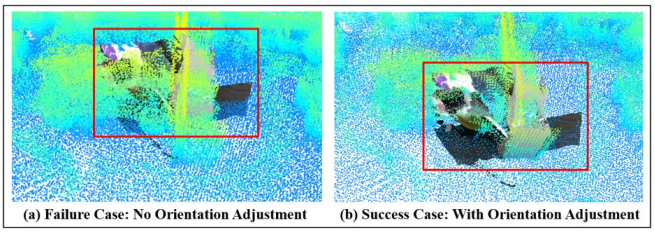

Fig. 4. Examples of point cloud alignment results

D. Approach for Automatic Parameter Estimation

In this study, to address the challenge of initial parameter dependency in point cloud alignment using ICP, we adopted a policy based on rough adjustment of the initial orientations. This is because the setting of the orientation has a significant impact on the success of the alignment as described above.

Specifically, different estimation methods are allied to the first camera and the other cameras respectively. For the first camera's point cloud, a rough orientation is manually set to ensure alignment with the 3D map. Then, multiple initial extrinsic parameters are generated within the pre-defined range around the manually set orientation. ICP is applied to each of these candidates of initial parameters. The RMSE (Root Mean Square Error) is used to evaluate each result, and the extrinsic parameter with the smallest RMSE is selected as the optimal solution. Next, the extrinsic parameter of the first camera is utilized for parameter estimation of adjacent cameras. From this step, estimation process is performed automatically. The details of this auto-estimation are described below. In this way, the proposed method minimizes manual effort by limiting it to a rough initial orientation adjustment of the first camera.

E. Inherited Automatic Search Method

The parameter estimation process of the second and subsequent cameras is also based on providing rough initial orientations. The proposed method adopts initial orientation setting with the extrinsic parameter of the adjacent camera and the relationship among adjacent cameras. Multiple RGB-D cameras were installed so that their fields of view among adjacent cameras are overlapped. This overlap is utilized to perform point cloud alignment between neighboring cameras.

For example, the parameter estimation process of the second camera is described. The second camera is assumed to be placed as overlapping with the first camera that the extrinsic parameter is obtained. At first, the relative transformation matrix between the first camera and the second camera is calculated from point cloud alignment of two cameras. Next, relative transformation matrix from this alignment is then multiplied by the extrinsic parameter matrix of the first camera. Through this operation, the transformation matrix including camera orientation around the optimal solution in the 3D map coordinate system can be obtaind. This transformation matrix serves as the initial parameter for the second camera in the ICP-based search. This is equivalent to the manual initial orientation adjustment performed for the first camera. After the third cameras, same procedures are iterated, and the all distributed cameras can obtain the suitable extrinsic parameters in the common 3D map coordinate system.

Figure 5 shows an example of alignment results between adjacent cameras, illustrating the initial states before alignment, the result after alignment, and the 3×4 transformation matrix obtained by ICP.

Fig. 5. Alignment between camera point clouds

Let the extrinsic parameter matrix of camera i in the 3D map coordinate system be $\mathbf{T}^{(i)}$, and the relative transformation matrix from camera j to camera i (obtained from alignment) be $\mathbf{T}_{i\to j}$. Then, the initial extrinsic parameter matrix of camera j in the 3D map coordinate system is computed as follows:

$$\mathbf{T}^{(j)}_{\text{init}} = \mathbf{T}_{i\to j} \cdot \mathbf{T}^{(i)}. \tag{1}$$

The initial matrix $\mathbf{T}^{(j)}_{\text{init}}$ obtained from Equation (1) is used as the initial parameter in the inherited automatic search method described in Section III-D, where the pose of camera j can be estimated. By sequentially processing based on cameras already aligned, the extrinsic parameters of all cameras from the second onward can be automatically estimated without manual adjustment.

IV. Building of the Intelligent Space with the Proposed Method

A. Overview

This section demonstrates the effectiveness of the proposed method by applying it to estimate the poses of multiple cameras. That means the intelligent space based on widely distributed RGB-D cameras is built automatically. The procedure of estimation follows the flowchart shown in Fig. 2. In this study, five RGB-D cameras (Camera A to E) were placed in the target space. The camera placement can be flexible as long as there are enough overlapping fields of view between adjacent cameras.

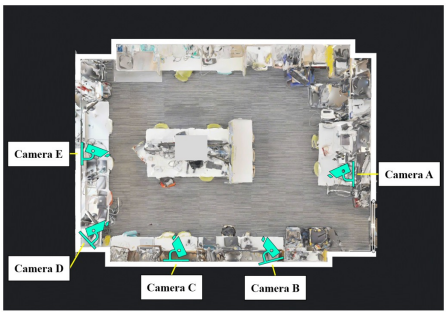

Fig. 6. Camera placement

Step 1: Point Cloud Acquisition

First, a 3D map of the target space was built in advance, and point cloud data were collected from each camera's viewpoint (Fig. 7).

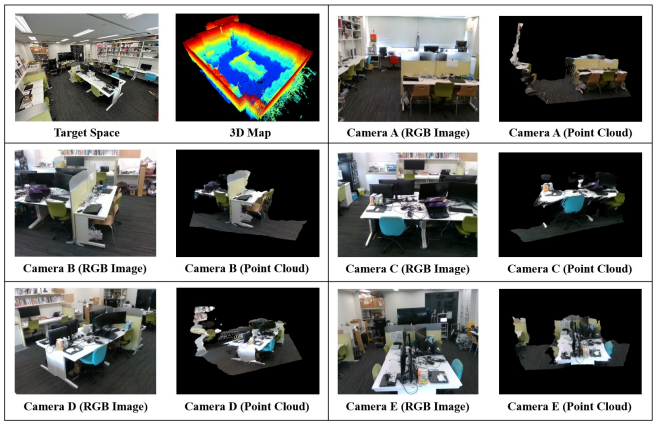

Fig. 7. Point cloud data for experiment

Step 2-A to Step 5 (Camera A)

For only the first camera (Camera A), a rough initial orientation was manually adjusted so that the camera angles around the x-, y-, and z-axes were within approximately 15 degrees of the true orientation, and the extrinsic parameter was estimated from the manual initial parameter using ICP (Top figure of Fig. 8).

Step 2-B to Step 5 (Camera B–E)

For the second and subsequent cameras (B to E), point cloud alignment was performed with adjacent cameras whose fields of view overlapped. The resulting relative transformation matrices (e.g., $\mathbf{T}_{A\to B}$) were combined with the extrinsic parameter matrices of already-aligned cameras (e.g., $\mathbf{T}^{(A)}$) to generate the initial extrinsic matrix for each camera (e.g., $\mathbf{T}^{(B)}_{\text{init}}$). Using these initial matrices, the extrinsic parameter of each camera in the common world coordinate can be estimated (Bottom four figures of Fig. 8).

In this inherited automatic search method, the range of variations around the initial parameters are set uniformly for all cameras. The positional variations are limited to ±2.0 m in the x and y directions and ±0.2 m in the z direction, while rotations are restricted to ±0° around the x and y axes and ±60° around the z axis in the world coordinate system. The target space measures approximately 6.0 × 8.5 × 2.7 [m],

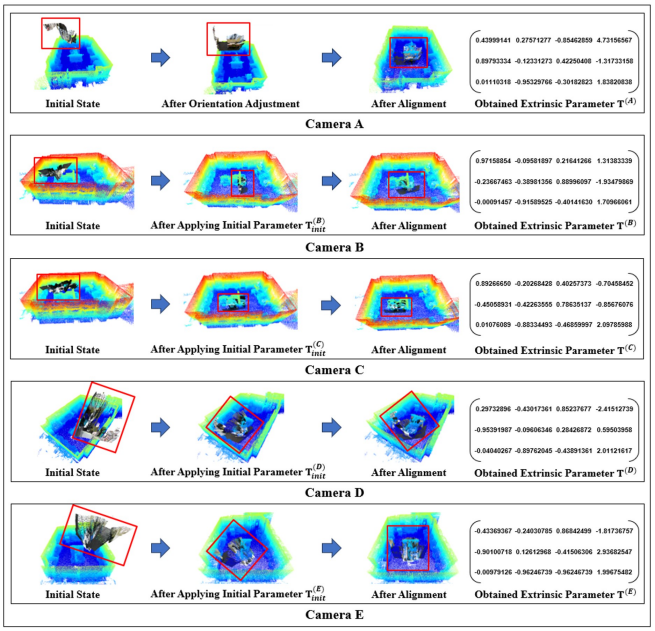

Fig. 8. Alignment results

and the search range is determined according to the scale of this space. The alignment process is repeated 100 times using different initial parameters in the range.

As demonstrated, the proposed method enables the integration of data from multiple cameras into a unified coordinate system and allows automatic pose estimation even for those placed at different angles.

B. Accuracy Verification of the Obtained Parameters

To evaluate the accuracy of the estimated camera parameters, three corresponding points in the world coordinate system were selected for each pair of adjacent cameras. Figure 9 illustrates an example of corresponding points between camera A and camera B. The local coordinates of each camera were transformed into the common world coordinate system using the obtained parameters, and the Euclidean distances between corresponding points were calculated as positional errors. The same procedure was applied to other pairs of adjacent cameras to assess accuracy. This verification is intended only as an approximate reference of accuracy and does not aim for strict accuracy evaluation because average errors might change according to the selection of points.

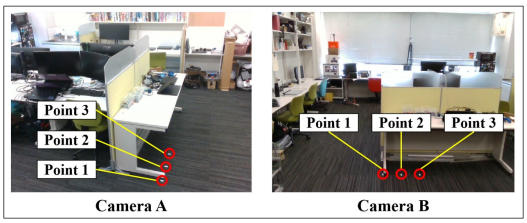

Fig. 9. Example of corresponding points observed by adjacent cameras

Table I summarizes the minimum, maximum, and average errors for each camera pair. Although the results include measurement errors due to the performance of the RGB-D cameras themselves, the positional errors remain within

TABLE I
ALIGNMENT ERRORS BETWEEN ADJACENT CAMERA PAIRS

Camera Pair	Min Error [m]	Max Error [m]	Average Error [m]
CamA - CamB	0.044	0.122	0.072
CamB - CamC	0.071	0.092	0.084
CamC - CamD	0.055	0.096	0.079
CamD - CamE	0.134	0.185	0.152

0.20 m. Therefore, the obtained parameters can be considered sufficiently accurate for constructing an intelligent space.

In the following chapter (Section V), we introduce a person tracking system built using these parameters as the practical application of the intelligent space. The tracking system is efficient for evaluating the overall performance of the intelligent space.

V. PERSON TRACKING SYSTEM

A. System Overview

This chapter introduces a wide-area person tracking system using multiple networked RGB-D cameras. Additionally, we constructed a digital twin system that represents tracked person information in real time. The performance of the tracking system can verify the accuracy of the camera parameters obtained automatically by the proposed method. Also, the tracking system demonstrates the feasibility of intelligent space applications by utilizing the proposed approach.

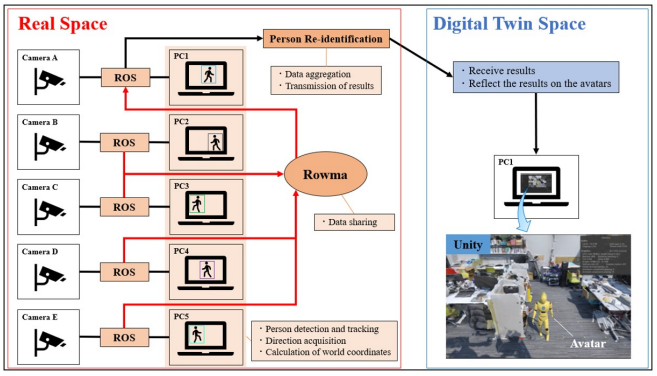

Fig. 10. Tracking system configuration

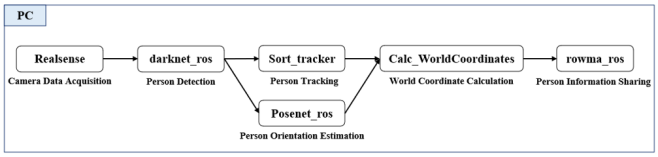

Fig. 11. Software configuration of each ROS system

Fig. 10 shows an overview of the entire tracking system, while Fig. 11 shows the ROS software configuration implemented in each camera. In the target space, five Intel RealSense D435 cameras are placed. Each camera is controlled by a separate ROS-based PC. Each camera performs person detection, tracking, and orientation estimation based on the acquired RGB and depth data, using YOLO [13] for person detection, SORT [14] for tracking, and PoseNet [15] for pose estimation. Each camera works independently of the other cameras. The position of a person is defined as the center of the detected bounding box. The orientation is estimated by calculating a vector between the 3D coordinates of both shoulders, as obtained from PoseNet, to represent the person's facing direction. The person positions obtained in each camera's local coordinate system are transformed by each extrinsic camera parameter estimated in the proposed method. Therefore, person positions in all cameras are represented in the common world coordinate system.

Next, a common ID number is assigned to the tracked person in the space, because same person identification cannot be performed by only independent tracking in each camera. Person identification is carried out by aggregating person data from all cameras asynchronously at once. The data communication among ROS-based cameras is implemented by Rowma network [16]. Specifically, if the following two conditions are satisfied, persons tracked by multiple cameras are considered the same and assigned a common ID:

- The person is tracked by multiple cameras within a maximum time difference of 0.3 seconds.
- The Euclidean distance between the person positions, measured in the world coordinate system on the (x, y) plane, is less than or equal to 0.6 m.

Finally, the identified person data (ID, position, and orientation) are then sent in real time to the digital twin system. The tracked person movements are reflected on avatars in the virtual environment. Each avatar, which is a 3D human model in the digital twin system, is displayed in a different color according to the person ID, allowing visual distinction among multiple persons. The digital twin system is constructed based on a 3D model of the target environment. In this study, the 3D model of the actual target space is 3D scanning by Matterport. By importing this 3D model into the virtual environment, the base environment of the digital twin system is represented.

In this way, the system assigns the common ID to the person tracked in multiple cameras. This enables real-time wide-area person tracking and the visualization of tracking information through reproduction in the digital twin.

B. Tracking Experiment

To evaluate the effectiveness of the proposed method described in Section IV, we conducted a person tracking experiment using the automatically estimated camera parameters shown in Fig. 8.

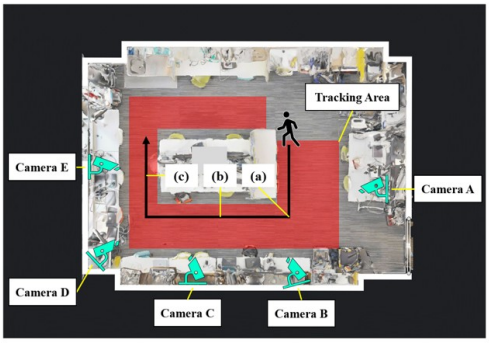

Fig. 12. Experiment setup

Fig. 12 illustrates the experimental setup. A single person walked along a predefined path spanning the fields of view of multiple cameras. In this system, each camera operates independently and asynchronously, performing person tracking within each field of view. Subsequently, person identification based on asynchronous tracking data aggregation from all cameras enables peron tracking with the common ID across multiple camera views.

Fig. 14 shows the trajectory of the person obtained by the system. The color of each data point indicates the same person positions obtained independently by the camera (A–E).

Fig. 13. Person tracking at each location

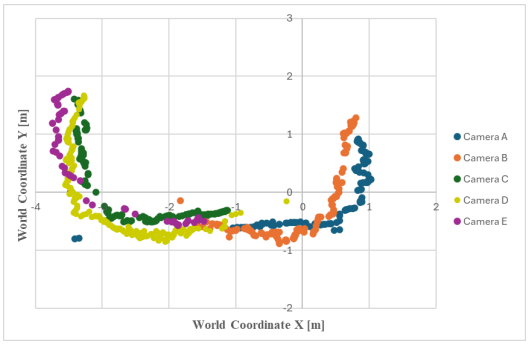

Fig. 14. Trajectory of movement

As shown in Fig. 13, the same color assignment to the person in the digital twin indicates that the person was tracked as the same person with the common ID steadily during the experiment. In addition, the trajectory shown in Fig. 14 closely aligns with the actual walking path, indicating that the camera parameters estimated by the proposed method are sufficiently accurate for use in person tracking systems.

VI. Conclusion and Future Work

This paper introduced the configuration supporting system of intelligent spaces based on distributed RGB-D cameras. Especially, we proposed the inherited automatic camera pose estimation method for easy construction of the intelligent spaces. This proposed method can flexibly adapt to various environments by increasing the number of cameras and arranging them with overlapping fields of view, demonstrating high scalability. The experimental results provided that the proposed method achieved accurate camera pose estimation in the actual intelligent space configured by RGB-D cameras. Also, the method significantly reduced the cumbersome manual setup traditionally required for configuring intelligent spaces. Person tracking experiments using multiple cameras with the automatically obtained parameters were performed. The results confirmed that continuous tracking across multiple cameras was achievable, demonstrating applications with practical accuracy in intelligent spaces built by the proposed method.

For future works, the required degree of overlap between adjacent cameras when applying the proposed method should be examined. In addition, applicability of the proposed method to intelligent space building in several types of environments such as larger spaces with complex structures should be evaluated. Furthermore, we plan to demonstrate the applications with mobile robots in the intelligent space configured with the proposed method.

References

[1] J.H. Lee and H. Hashimoto, "Intelligent space - concept and contents", Advanced Robotics, vol. 16, no. 3, pp. 265–280, 2002.

[2] K. Morioka, J.H. Lee and H. Hashimoto, "Human-following mobile robot in a distributed intelligent sensor network", IEEE Transactions on Industrial Electronics, vol. 51, no. 1, pp. 229-237, 2004.

[3] K. Koide and E. Menegatti, "Non-overlapping RGB-D Camera Network Calibration with Monocular Visual Odometry", 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 9005-9011, 2020.

[4] T. Zeng, D. He, F. Yan and M. He, "YOCO: You Only Calibrate Once for Accurate Extrinsic Parameter in LiDAR-Camera Systems", arXiv preprint arXiv:2407.18043, 2024.

[5] H. Liu, H. Li, X. Liu, J. Luo, S. Xie and Y. Sun, "A Novel Method for Extrinsic Calibration of Multiple RGB-D Cameras Using Descriptor-Based Patterns", Sensors, vol. 19, no. 2, p. 349, 2019.

[6] B.-H. Yoon, H.-W. Jeong and K.-S. Choi, "Targetless Multiple Camera-LiDAR Extrinsic Calibration using Object Pose Estimation", 2021 IEEE International Conference on Robotics and Automation (ICRA), pp. 13377-13383, 2021.

[7] S.-E. Lee, K. Shibata, S. Nonaka, S. Nobuhara and K. Nishino, "Extrinsic Camera Calibration From a Moving Person", IEEE Robotics and Automation Letters, vol. 7, no. 4, pp. 10344-10351, 2022.

[8] J. M. Naruniec, M. Munaro, A. Fossati, L. Natale and E. Menegatti, "OpenPTrack: Open source multi-camera calibration and people tracking for RGB-D camera networks", Robotics and Autonomous Systems, vol. 75, pp. 525–538, 2016.

[9] E. Ataer-Cansizoglu, Y. Taguchi, S. Ramalingam and Y. Miki, "Calibration of Non-overlapping Cameras Using an External SLAM System", 2014 2nd International Conference on 3D Vision, pp. 509-516, 2014.

[10] T. Pollok and E. Monari, "A Visual SLAM-Based Approach for Calibration of Distributed Camera Networks", 2016 13th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS), pp. 429-437, 2016.

[11] J. Xu, R. Li, L. Zhao, W. Yu, Z. Liu and B. Zhang, "CamMap: Extrinsic Calibration of Non-Overlapping Cameras Based on SLAM Map Alignment", IEEE Robotics and Automation Letters, vol. 7, no. 4, pp. 11879-11885, 2022.

[12] N. Koide, "koide3/hdl_graph_slam: 3D LIDAR-based Graph SLAM", GitHub repository, last updated Jul. 16, 2024. [Online]. Available: https://github.com/koide3/hdl_graph_slam

[13] J. Redmon, S. Divvala, R. Girshick and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection", 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 779-788, 2016.

[14] A. Bewley, Z. Ge, L. Ott, F. Ramos and B. Upcroft, "Simple Online and Realtime Tracking", 2016 IEEE International Conference on Image Processing (ICIP), pp. 3464-3468, 2016.

[15] A. Kendall, M. Grimes and R. Cipolla, "PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization", 2015 IEEE International Conference on Computer Vision (ICCV), pp. 2938-2946, 2015.

[16] R. Suenaga and K. Morioka, "Rowma: A Reconfigurable Robot Network Construction System", 2021 IEEE/SICE International Symposium on System Integration (SII), pp. 537-542, 2021.