automation

IMPACT FACTOR 2.9

CITESCORE 4.5

Article

Reliability-Aware Occupancy Map Merging in Dynamic Environments Using Temporal and Probabilistic Maps

Hanngyoo Kim and Seunghwan Lee

Special Issue

AI-Enhanced Measurement and Control for Robotic Systems

Edited by

Dr. Ameer Tamoor Khan and Prof. Dr. Shuai (Steven) Li

MDPI

https://doi.org/10.3390/automation7040119

automation

MDPI

Article

Reliability-Aware Occupancy Map Merging in Dynamic Environments Using Temporal and Probabilistic Maps

Hanngyoo Kim and Seunghwan Lee *

School of Electronic Engineering, Kumoh National Institute of Technology, Gumi 39177, Republic of Korea; kaywen1227@gmail.com
* Correspondence: leesh@kumoh.ac.kr

Abstract

Occupancy map merging in dynamic environments is challenging because moving objects introduce time-varying disturbances that degrade registration accuracy and structural consistency. This paper proposes a reliability-aware map-merging method that explicitly models temporal validity and probabilistic reliability in occupancy grid maps. The method constructs two complementary maps: a trajectory-guided temporal reliability map that reflects the recency and persistence of cell observations and a probabilistic reliability map that refines detected candidate regions based on temporal persistence and spatial reliability criteria. By suppressing transient clutter before registration, the proposed approach focuses alignment on structurally stable regions and improves merging robustness. Experiments in simulation and real-world indoor environments demonstrate clear improvements over a conventional feature-based baseline, substantially reducing both translation and rotation errors. These results show that incorporating temporal validity and probabilistic reliability can improve occupancy map merging under dynamic conditions.

Keywords: map merging; map matching; SLAM; multi-robot SLAM; multi-agent robot system; occupancy grid map; temporal reliability; probabilistic reliability; hierarchical approach; dynamic environments

1. Introduction

Map merging is a fundamental capability in multi-robot systems because it enables robots operating in the same environment to integrate local maps into a unified global representation. Through this process, robots can share spatial information, reduce localization uncertainty, and improve their overall understanding of the environment. Such capability is increasingly important in many automation applications, including logistics, manufacturing, surveillance, and service robotics, where multiple robots are required to operate cooperatively in large-scale or partially overlapping spaces [1–5].

A wide range of map-merging methods has been studied over recent decades. Early work mainly focused on aligning occupancy grid maps by searching for overlapping regions and estimating relative transformations between maps [1,6,7]. Subsequent studies improved robustness by introducing feature-based registration, resolution-adaptive matching, and more reliable initialization strategies. For example, Ma et al. [8] proposed a trimmed and scaling iterative closest point (TsICP) method for maps with different resolutions, while Topal et al. [9] and Jiang et al. [10] developed feature-based approaches for robust map matching under uncertain overlap conditions. Other studies further addressed

Check for updates

Academic Editor: Tianwen Zhang

Received: 29 May 2026
Revised: 17 July 2026
Accepted: 28 July 2026
Published: 1 August 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

limited overlap and poor initialization using directional observations, Radon-transform-based descriptors, or spectral features [11,12]. More broadly, survey and system-level studies have discussed map merging in heterogeneous multi-robot settings and emphasized the need for robust matching across different sensing conditions [2,3,13–15]. Despite these advances, most existing methods are still developed under an implicit quasi-static assumption. In practice, however, indoor and service environments often contain moving objects such as people, carts, or temporary obstacles. These dynamic entities are frequently recorded in occupancy grid maps as if they were static structures, even though their positions change over time. As a result, the maps may contain outdated traces that do not represent the persistent geometry of the environment. When such maps are later aligned and merged, these transient observations can distort feature extraction, degrade geometric consistency, and ultimately destabilize the estimated transformation.

Figure 1 illustrates this problem. Dynamic objects observed at one time instant may later move away while their occupancy traces remain in the map. This issue becomes more severe when maps generated at different times are merged because time-varying discrepancies accumulate and amplify alignment errors. Therefore, observations in occupancy maps should not be treated as equally reliable over time. Instead, their temporal validity and spatial consistency should be explicitly considered during map refinement and merging. Recent studies on dynamic environments have addressed related issues from different perspectives. Chen et al. [16] considered continuous occupancy mapping in changing environments, while Malakouti-Khah et al. [17] studied multi-robot simultaneous localization and mapping (SLAM) in dynamic settings with unknown correspondences. Ni et al. [18] improved visual SLAM by filtering unreliable dynamic map points, and Stathoulopoulos et al. [19] proposed a scalable map-merging framework based on place recognition and global pose optimization. Related advances have also been reported in robotic localization and environmental mapping. Kolla and Vundavilli [20] developed a cartographer-based SLAM approach using multi-sensor fusion for indoor trajectory estimation. Pimentel de Figueiredo et al. [21] presented a LiDAR–inertial–camera system for three-dimensional semantic–geometric mapping in industrial environments, while Salunkhe et al. [22] developed a SLAM-based framework for autonomous indoor CO_2 mapping and environmental monitoring. Collectively, these studies demonstrate continued progress in dynamic mapping, sensor-fusion SLAM, semantic mapping, and environmental mapping. However, comparatively little attention has been given to the explicit refinement of completed two-dimensional occupancy maps for the map-merging stage itself through the joint consideration of temporal decay and probabilistic reliability. To address this limitation, this paper proposes a reliability-aware occupancy map-merging method for dynamic environments. The key idea is to suppress transient and unreliable observations before map registration so that alignment is governed mainly by temporally stable structures. To this end, we construct two complementary reliability representations: a trajectory-guided temporal reliability map that reflects the recency of cell observations and a probabilistic reliability map that refines detected candidate regions based on temporal persistence and spatial reliability criteria. These complementary reliability cues are integrated to attenuate or remove unreliable dynamic artifacts prior to map alignment and fusion.

The main contributions of this work are summarized as follows:

- We propose a reliability-aware occupancy-grid map-merging framework based on two complementary information streams in which transient obstacles are suppressed before registration.
- We formulate a trajectory-guided temporal reliability map using a time-decay model so that stale occupancy evidence is progressively down-weighted when it is not re-observed.

- We develop a reliability-refinement procedure that integrates temporal persistence with spatial-preservation and boundary-context criteria to reduce the influence of dynamic artifacts during alignment.
- We validate the proposed method through both simulation and real-world experiments, demonstrating improved merging performance over a conventional feature-based baseline.

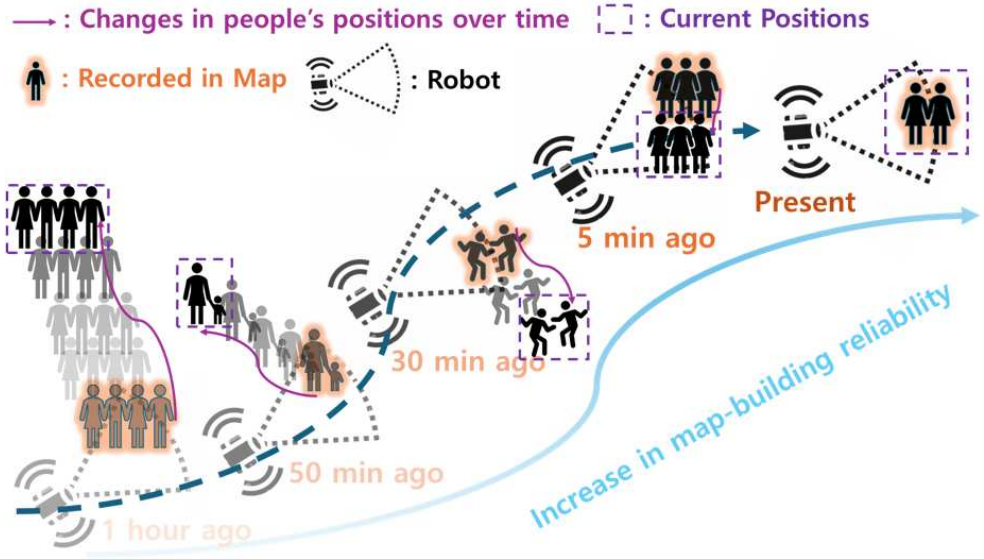


Figure 1. Problem illustration of dynamic objects during the mapping process. Traditional approaches directly record moving entities such as people in the map. However, their positions often change over time, leaving outdated traces that reduce map-merging reliability. This observation motivates the use of temporal reliability estimation and probabilistic filtering to mitigate the influence of outdated dynamic information.

The remainder of this paper is organized as follows. Section 2 presents the proposed method, including notation, problem formulation, temporal reliability modeling, probabilistic reliability filtering, and map alignment. Section 3 reports the simulation and real-world experimental results. Section 4 discusses the main findings and limitations of the proposed approach. Finally, Section 5 concludes the paper and outlines future research directions.

2. Methods

2.1. Notation and Terminology

Table 1 summarizes the main symbols used throughout this paper. The notation is grouped by workspace, map representations, robot trajectories, rigid transformations, dynamic-object variables, temporal decay, and removal logic. Scalars are denoted in italic, vectors and matrices in boldface, sets in calligraphic style, and operators in upright roman font. Unless otherwise stated, occupancy values lie in the range of $[0, 1]$, and rotation angles are expressed in degrees. The terminology used in this paper is outlined as follows. The spatial map (M_S) denotes a two-dimensional LiDAR (Light Detection and Ranging)-based occupancy grid generated by simultaneous localization and mapping (SLAM). The time-decay map (M_T) adjusts cell occupancies according to their recency so that older observations gradually relax toward the unknown state. The probabilistic reliability map (M_R) is the final refined map obtained after suppressing unreliable dynamic regions. A dynamic object cluster (D_j^i) denotes a fused region obtained from convolutional neural network (CNN)-based detections and Density-Based Spatial Clustering of Applications with Noise (DBSCAN). Its occupied persistence score (s_j^i) represents the confidence that the detected occupied evidence remains valid in the environment. Dynamic regions

are then handled using either hard removal or Gaussian-based soft attenuation. Finally, the alignment transform ($\mathcal{T}_{i\to j}$) denotes the rigid transformation used to register M_R^i to M_R^j. Other acronyms used in this paper include OGM (Occupancy Grid Map), SURF (Speeded-Up Robust Features), and IMU (Inertial Measurement Unit).

Table 1. Notation used throughout the paper.

Symbol	Meaning
Indices, sets, and workspace	
$\Omega \subset \mathbb{R}^2$	2D workspace (index set of grid cells).
$l \in \{1, \ldots, N\}$	Grid-cell index.
$\mathcal{F}, \mathcal{O}, \mathcal{U}$	Free, occupied, and unknown cell sets, respectively.
Maps and fields	
M_S^i	Spatial (SLAM) occupancy map of robot i.
$m_l^i \in [0,1]$	Occupancy value of cell l on M_S^i.
$r_l^i \in [0,1]$	Cell reliability (static confidence) on M_S^i.
M_T^i	Time-decayed occupancy map of robot i.
M_R^i	Probabilistic reliability map of robot i (refined/filtered).
Robot trajectory and time	
$\mathcal{P}^i = \{(p_k, t_k)\}_{k=1}^M$	Discrete positions and timestamps along robot i's trajectory.
$t_i(x,y)$	Last update time of cell (x,y) by robot i.
t_{final}	Final observation time at server.
$\Delta t_i(x,y)$	Temporal age: $t_{\text{final}} - t_i(x,y)$.
Transforms and objectives	
$\mathbf{R} \in SO(2),\ \mathbf{t} \in \mathbb{R}^2$	Rotation and translation for rigid alignment.
$T_{\mathbf{R},\mathbf{t}}(\cdot)$	Rigid transform acting on cell coordinates.
$(\mathbf{R}^*, \mathbf{t}^*)$	Optimal transform minimizing map discrepancy.
$\mathcal{T}_{i\to j}$	Estimated transform aligning M_R^i to M_R^j.
Dynamic objects and detections	
$\mathcal{F}_\theta$	CNN detection function with weights (θ).
$\mathcal{B}_k^i$	Bounding boxes detected on local crop at pose k.
$\mathcal{B}_{\text{global}}^i$	Globalized union of detections for robot i.
D_j^i	Dynamic object cluster region (post fusion).
B_j^i	Individual dynamic object bounding box.
∂D_j^i	Boundary cells of cluster D_j^i.
o_{ext}	Cells in D_j^i outside original boxes (structure check).
Temporal decay and reliability	
$\lambda > 0$	Temporal decay rate in (10).
α, β	Occupancy thresholds ($0 \le \alpha < 0.5 < \beta \le 1$).
$\gamma > 0$	Weight for spatio-temporal association cost.
$\bar{m}_{j,i}^T$	Mean time-decayed occupancy over D_j^i.
$s_j^i \in [0,1]$	Occupied persistence score.
Removal/Blending logic	
ρ_j^i	Unknown-cell ratio in D_j^i.
o_j^i	Mean occupancy of o_{ext} (structure-preservation check).
$\theta_u, \theta_o, \theta_b$	Thresholds for unknown, occupied, and boundary tests.
k_c, σ_c	Cluster-level Gaussian kernel size/standard deviation.
$\mathcal{R}_{\text{free}}^i$	Fully removed regions (set to free).
$\mathcal{R}_{\text{blend}}^i$	Blended (attenuated) regions.
$\text{Free}(\cdot), \text{Blend}(\cdot)$	Removal operators (hard and soft).

2.2. Background

We consider the problem of aligning two spatial maps of a planar environment, denoted by M_S^1 and M_S^2, where each map is represented as an occupancy grid or an equivalent feature map. Let $\Omega \subset \mathbb{R}^2$ denote the discretized workspace, and let $l \in \{1, 2, \ldots, N\}$ denote

a grid-cell index. For each map (M_S^i ($i = 1, 2$)), the value of $m_l^i \in [0, 1]$ represents the occupancy probability of cell l. Values close to 1 indicate occupied cells, values close to 0 indicate free cells, and intermediate values indicate uncertainty. In addition, $r_l^i \in [0, 1]$ denotes the reliability of cell l—namely, the confidence that the cell belongs to a temporally stable part of the environment.

Because the two maps may be generated at different times or from different viewpoints, they can be inconsistent due to dynamic or transient objects. To align M_S^2 with M_S^1, we estimate a rigid transformation composed of a rotation matrix ($\mathbf{R} \in SO(2)$) and a translation vector ($\mathbf{t} \in \mathbb{R}^2$). The goal is to find the optimal transformation $(\mathbf{R}^*, \mathbf{t}^*)$ that best aligns the two maps.

Conventional methods often define alignment by minimizing the discrepancy between corresponding occupancy values:

$$(\mathbf{R}^*, \mathbf{t}^*) = \arg\min_{\mathbf{R},\mathbf{t}} \sum_{l \in \Omega} w_l \left\| m_l^1 - m_{S,\, T_{\mathbf{R},\mathbf{t}}(l)}^2 \right\|^2, \tag{1}$$

where $T_{\mathbf{R},\mathbf{t}}(l)$ denotes the transformed location of cell l and w_l is a weighting factor. In the simplest case, $w_l = 1$ for all cells, so all overlapping cells contribute equally.

However, such a formulation is vulnerable in dynamic environments. Even if the estimated rigid transformation is geometrically correct, mismatches caused by moving objects can produce large residuals and degrade registration quality. A natural extension is to use reliability-aware weights, for example,

$$w_l = r_l^1 \cdot r_{T_{\mathbf{R},\mathbf{t}}(l)}^2, \tag{2}$$

so that cells that are likely to be dynamic contribute less to the objective. Although this improves robustness, it still operates on the original occupancy maps and therefore cannot completely suppress dynamic artifacts when the reliability itself is estimated from noisy observations.

To address this limitation more directly, we construct a probabilistic reliability map that refines each spatial map before alignment. For robot i, the refined map is defined as $M_R^i = \{m_{R,l}^i\}$, where

$$m_{R,l}^i = r_l^i \cdot m_l^i. \tag{3}$$

Here, r_l^i acts as a soft gate that attenuates unreliable occupancy evidence. Since M_R^i is used for registration rather than probabilistic inference, additional normalization is unnecessary. As a result, cells associated with dynamic or uncertain structures are down-weighted, while cells corresponding to stable structures are preserved.

The alignment is then performed on the refined maps:

$$(\mathbf{R}^*, \mathbf{t}^*) = \arg\min_{\mathbf{R},\mathbf{t}} \sum_{l \in \Omega} \left\| m_{R,l}^1 - m_{R,\, T_{\mathbf{R},\mathbf{t}}(l)}^2 \right\|^2, \tag{4}$$

where $m_{R,\, T_{\mathbf{R},\mathbf{t}}(l)}^2$ denotes the transformed value from map 2. In this way, alignment is driven primarily by temporally persistent and spatially reliable structures, resulting in more robust map merging under dynamic conditions.

2.3. Proposed Map-Merging Method

We propose a reliability-aware map-merging framework with the common objective of reducing the influence of transient occupancy evidence before map registration. As shown in Figure 2, the framework uses three map representations for each robot i: the spatial occupancy map (M_S^i), the time-decay map (M_T^i), and the reliability-refined map M_R^i. Each robot independently performs SLAM and estimates its trajectory. After the local SLAM

process is completed, the resulting occupancy maps and trajectories are transmitted to a central server, where temporal reliability estimation, dynamic-region refinement, map alignment, and merging are performed offline. Starting from M_S^i and the corresponding robot trajectory, dynamic-object candidates are detected and spatially fused, while the trajectory information is used to construct M_T^i, which represents the temporal validity of the occupancy evidence. The temporal information in M_T^i is subsequently used to evaluate the persistence of the detected candidate regions, while spatial-preservation and boundary-context criteria determine whether and how these regions are refined. These complementary temporal and spatial evaluations produce M_R^i. The resulting reliability-refined maps are finally used as inputs to the SURF-based registration and map-composition process. Thus, each stage provides information required by the subsequent reliability evaluation and contributes to the common objective of performing map alignment using more persistent and reliable environmental structures. The overall procedure consists of four connected stages:

1. Dynamic-object extraction and spatial fusion: detect candidate dynamic-object regions and combine spatially related detections into fused candidate regions;
2. Temporal-reliability map generation: construct the time-decay map (M_T^i) from the robot trajectory to represent the temporal validity of occupancy evidence;
3. Reliability-refined map generation: evaluate the detected candidate regions using temporal persistence, spatial-preservation, and boundary-context information to generate M_R^i;
4. Map alignment and merging: use the reliability-refined maps for SURF-based registration and subsequent map composition.

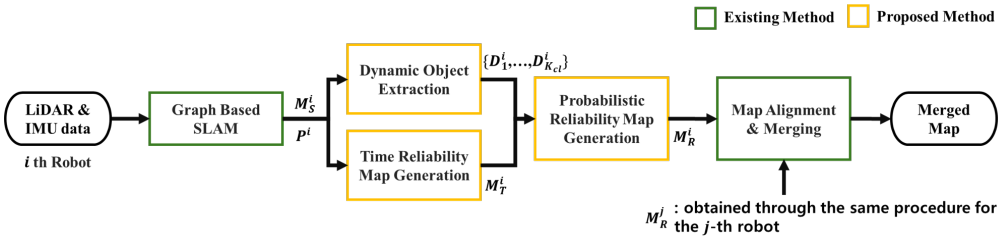


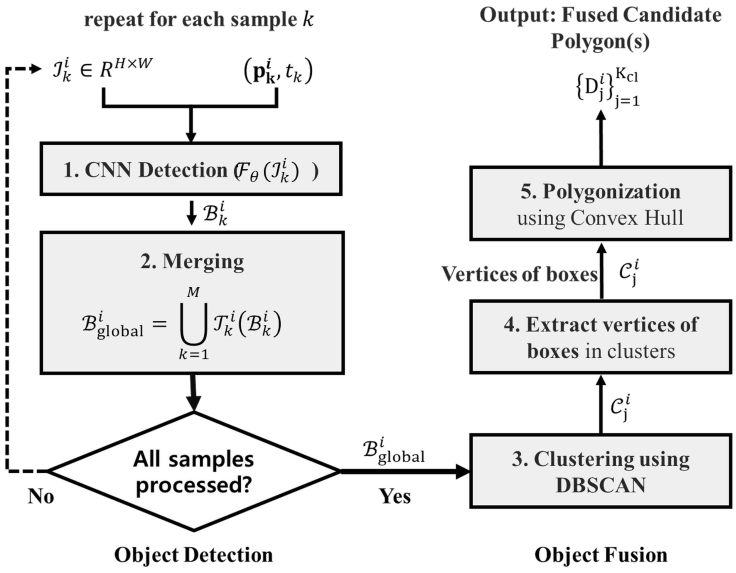
Figure 2. Overview of the proposed occupancy map-merging pipeline. Each robot first performs graph-based SLAM using LiDAR and IMU data to generate a spatial occupancy map (M_S^i) and an estimated trajectory. The resulting map and trajectory are then used to (1) extract dynamic-object regions and (2) construct a temporal reliability map. These two outputs are combined to generate a probabilistic reliability-refined map (M_R^i), which is finally used for map alignment and merging to produce the integrated map.

2.3.1. Dynamic Object Extraction

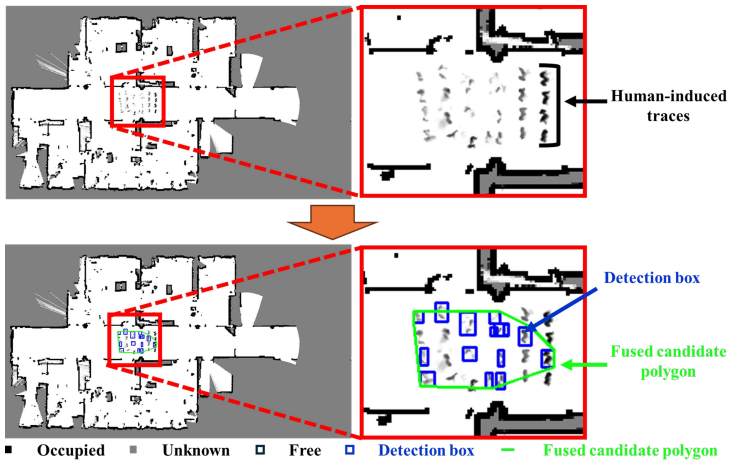
Figure 3a summarizes the sequential dynamic-object detection and spatial-fusion workflow, consisting of CNN-based candidate detection, merging of detections across map samples, DBSCAN-based spatial clustering, and convex-hull polygonization. Dynamic-object candidates are extracted using a YOLOv8x detector implemented with the Ultralytics framework [23]. The model is initialized with pretrained weights and fine-tuned on 551 manually annotated occupancy-grid images using a 9:1 training–validation split. A single class, "human on occupancy grid," is used to identify human-induced occupancy traces. Each occupancy map is converted into a three-channel image, and the confidence and NMS IoU thresholds are set to 0.60 and 0.50, respectively. The same thresholds are used for both full-map and tiled inference. The detector only proposes candidate dynamic regions, which are subsequently evaluated using temporal reliability and spatial-consistency criteria. False-positive candidates may be rejected or preserved without modification during this

validation process, while the effect of false negatives can be partially mitigated through spatial grouping and polygon fusion of neighboring detections. Directly feeding the full map to the detector typically requires resizing to a fixed input resolution, which may distort geometric details and degrade detection performance. Moreover, dynamic traces on occupancy grids are often irregular or fragmented. To mitigate these issues, we adopt a two-step pipeline consisting of object detection and object fusion. In the detection stage, we sample M poses along the trajectory of robot i, denoted by $\mathcal{P}^i = \{(\mathbf{p}_k^i, t_k)\}_{k=1}^{M}$, where $\mathbf{p}_k^i \in \mathbb{R}^2$ is the robot position at time t_k. At each sampled pose, we crop a local map patch ($\mathcal{I}_k^i \in \mathbb{R}^{H \times W}$) at the original map resolution. Zero padding is applied near the boundaries, intensities are normalized to $[0, 1]$, and unknown cells are set to 0.5. The detection function ($\mathcal{F}_\theta$) then produces a set of bounding boxes:

$$\mathcal{B}_k^i = \mathcal{F}_\theta(\mathcal{I}_k^i). \tag{5}$$



(**a**)



(**b**)

Figure 3. Overview of dynamic-object extraction and spatial fusion. (**a**) Sequential workflow consisting of CNN-based candidate detection, merging of detections across map samples, DBSCAN-based spatial clustering, and convex-hull polygonization. (**b**) Example of the resulting candidate extraction and fusion, where human-induced occupancy traces are enclosed by individual detection boxes (blue) and spatially related detections are combined into fused candidate polygons (green) for subsequent temporal and spatial reliability evaluation.

Each local detection is transformed into the global map frame using the corresponding origin transform ($\mathcal{T}_k^i$) and accumulated as

$$\mathcal{B}_{\text{global}}^i = \bigcup_{k=1}^{M} \mathcal{T}_k^i(\mathcal{B}_k^i) = \{ b_n^i \}_{n=1}^{K_{\text{box}}}. \tag{6}$$

This aggregation preserves all candidate detections, including overlapping ones, so that temporal evidence from different viewpoints is retained.

Next, we fuse the detections into spatially coherent dynamic regions. When multiple dynamic objects move close to one another, per-frame CNN detections can be incomplete or fragmented because of occlusion and motion blur, as shown in Figure 3b. To obtain robust object regions, we cluster the bounding-box centroids using DBSCAN [24]. For each cluster ($\mathcal{C}_j^i$), we construct a compact polygonal region from the vertices of the constituent bounding boxes using convex-hull-based polygonization [25]:

$$D_j^i = \text{ConvHull}\left(\bigcup_{b \in \mathcal{C}_j^i} \text{Verts}(b) \right). \tag{7}$$

Thus, the fused set of dynamic-object regions is written as

$$\{ D_1^i, D_2^i, \ldots, D_{\text{K}_{\text{cl}}}^i \} = \text{DynamicObjectFusion}\left(\mathcal{B}_{\text{global}}^i \right). \tag{8}$$

The resulting fused candidate regions are not directly removed from the occupancy map; instead, they define the spatial regions whose temporal persistence and spatial reliability are evaluated in the subsequent refinement stages.

2.3.2. Temporal-Reliability Map Generation

The time-decay map (M_T^i) forms the temporal information stream of the proposed framework and is subsequently combined with the spatial candidate regions extracted from M_S^i during reliability refinement. Figure 4 illustrates the overall process of trajectory-guided temporal-reliability map generation. The spatial occupancy-grid map (M_S^i) and the selected region of interest are shown in Figure 4a, while Figure 4b represents the temporal ordering of the final robot trajectory from older to newer observations. The resulting cell-update times are visualized in Figure 4c, where the color progression represents the estimated temporal order of the occupancy evidence. Free cells satisfy $M_S^i(x, y) \in [0, \alpha)$, occupied cells satisfy $M_S^i(x, y) \in (\beta, 1]$, and the remaining cells are treated as unknown. Here, α and β denote predefined occupancy thresholds satisfying $0 \leq \alpha < 0.5 < \beta \leq 1$.

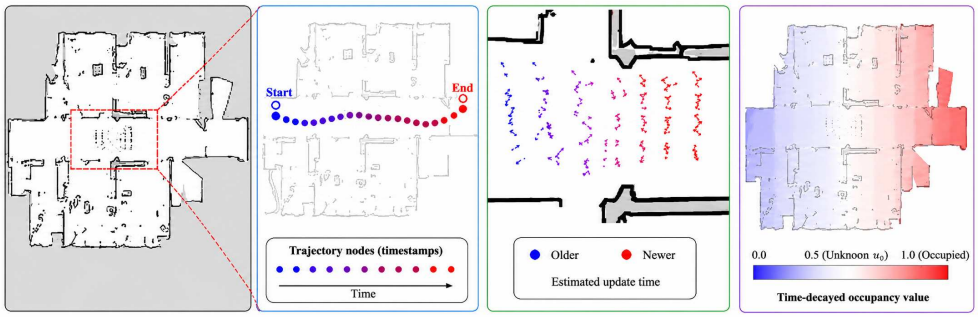
To model temporal reliability, the temporal-decay function is designed to satisfy the following behavior. Recently observed cells should retain their current occupancy evidence, whereas the confidence of stale observations should decrease monotonically with observation age. Moreover, stale free or occupied evidence should gradually approach the neutral unknown-state value of $u_0 = 0.5$ rather than being converted into the opposite occupancy state. These requirements motivate an exponential relaxation model toward u_0. Let $t_i(x, y)$ denote the last update time of cell (x, y) by robot i, and let t_{final} be the final observation time. The temporal age of the cell is

$$\Delta t_i(x, y) = t_{\text{final}} - t_i(x, y). \tag{9}$$

Accordingly, the time-decayed occupancy map is defined as

$$M_T^i(x, y) = u_0 + \left(M_S^i(x, y) - u_0 \right) \exp(-\lambda \, \Delta t_i(x, y)), \tag{10}$$

where $\lambda > 0$ is the decay rate and $u_0 = 0.5$ is the unknown-state baseline. This formulation preserves unknown cells and gradually relaxes free and occupied estimates toward uncertainty as they become older.



(**a**) (**b**) (**c**) (**d**)

Figure 4. Trajectory-guided temporal-reliability map generation. (**a**) Occupancy-grid map with the region of interest highlighted. (**b**) Final robot trajectory with temporal order indicated from older observations (blue) to newer observations (red). (**c**) Estimated update times of cells in the highlighted region obtained through trajectory-based spatiotemporal association, where the color progression from blue to red represents older to more recent estimated cell-update times. (**d**) Resulting time-decay map in which temporally older occupancy evidence is relaxed toward the unknown-state value, whereas more recent evidence retains a stronger occupancy value.

In offline SLAM settings, the update time of each cell is estimated using the completed occupancy map and the corresponding final SLAM trajectory represented in the same coordinate frame. To reduce ambiguity when the robot revisits the same location, the associated trajectory node is selected using the following spatiotemporal cost:

$$C_k(x, y) = \|(x, y) - \mathbf{p}_k\| + \gamma(t_{\text{final}} - t_k), \tag{11}$$

where the first term measures spatial proximity and thereby maintains local geometric relevance, whereas the second term favors more recent observations. The γ coefficient controls the contribution of temporal recency relative to spatial proximity; increasing γ gives greater preference to recent trajectory nodes, whereas decreasing it makes the association closer to a purely spatial rule. The most relevant trajectory-node index and the corresponding estimated cell-update time are given by

$$k^* = \arg\min_k C_k(x, y), \qquad t_i(x, y) = t_{k^*}. \tag{12}$$

The estimated cell-update time is subsequently used to compute the temporal age ($\Delta t_i(x, y)$) for the time-decay model in Equation (10). Applying these temporal ages to Equation (10) yields the final time-decay map (M_T^i) shown in Figure 4d. In this map, temporally older occupancy evidence is relaxed toward the unknown-state value, whereas more recent evidence retains a stronger occupancy value. The resulting temporal information is subsequently used to evaluate the persistence of the detected candidate regions.

2.3.3. Probabilistic-Reliability Map Generation

This stage integrates the two complementary information streams generated in the preceding stages: the fused candidate regions extracted from M_S^i and the temporal persistence information derived from M_T^i. Spatial-preservation and boundary-context criteria then determine whether and how the candidate regions are refined to produce M_R^i. When cluster-level suppression is skipped by the preservation checks, relevant individual detec-

tion boxes within the preserved regions may still undergo subsequent localized evaluation. Using the time-decay map (M_T^i), we estimate the reliability of each detected dynamic-object region with respect to the final observation time (t_{final}) and suppress regions that are unlikely to correspond to persistent structures. For each dynamic-object region (D_j^i), the average time-decayed occupancy is defined as

$$\bar{m}_{j,i}^T = \frac{1}{|D_j^i|} \sum_{(x,y)\in D_j^i} M_T^i(x,y), \tag{13}$$

where $|D_j^i|$ denotes the number of cells contained in D_j^i. The quantity $\bar{m}_T^{j,i}$ represents the mean time-decayed occupancy evidence within the detected candidate region. Because 0.5 denotes the neutral unknown state, a value of $\bar{m}_T^{j,i}$ close to 0.5 indicates that the previously observed occupancy evidence has become temporally ambiguous. In contrast, a value farther from 0.5 indicates that the evidence remains more clearly distinguishable from the unknown state. We therefore use the normalized absolute deviation from 0.5 to quantify the temporal persistence of the region, defining the occupied persistence score as

$$s_j^i = 2\left|\bar{m}_T^{j,i} - 0.5\right| \tag{14}$$

This formulation yields $s_j^i = 1$ when $\bar{m}_{j,i}^T$ is close to 1 or 0 and $s_j^i = 0$ when $\bar{m}_{j,i}^T = 0.5$. In other words, the occupied persistence score increases as the confidence moves away from the unknown-state baseline and decreases as the region becomes temporally ambiguous. The occupied persistence score is then used to determine whether a detected dynamic region should be retained, attenuated, or removed. Regions with high occupied persistence scores are regarded as reliable, whereas regions with low occupied persistence scores are treated as transient or outdated. This strategy is motivated by the observation that recently re-observed and spatially coherent objects are more likely to persist, while older or weakly supported traces are more likely to be spurious.

A practical advantage of operating on clustered dynamic-object regions rather than individual detections is that residual traces missed by the detector can be evaluated together within a fused candidate region. However, such regions may also include unknown cells or nearby static structures such as walls. Therefore, before applying cluster-level suppression, each candidate region is evaluated using a hierarchical set of spatial-preservation and refinement criteria. First, the unknown-cell ratio within each region is computed as

$$\rho_j^i = \frac{1}{|D_j^i|} \sum_{(x,y)\in D_j^i} \mathbf{1}\{M_S^i(x,y) \in \mathcal{U}\}. \tag{15}$$

If $\rho_j^i > \theta_u$, where $\theta_u \in [0,1]$ is a predefined threshold, cluster-level suppression is skipped because the available occupancy information is considered insufficient to safely modify the entire fused region. This does not classify the region as static or terminate the refinement process; relevant individual detection boxes within the preserved region may still undergo subsequent localized evaluation. Second, we examine the occupancy of cells that lie outside the original bounding boxes but remain inside the fused candidate region. Let $o_{\text{ext}} \subset D_j^i$ denote this set of external cells. Their mean occupancy is defined as

$$o_j^i = \frac{1}{|o_{\text{ext}}|} \sum_{(x,y)\in o_{\text{ext}}} M_S^i(x,y). \tag{16}$$

If $o_j^i \geq \theta_o$, where $\theta_o \in [0,1]$ is an occupancy threshold, cluster-level suppression is also skipped because the external occupancy may indicate the inclusion of persistent static structures.

For candidates that pass these preservation checks, the local boundary context determines the refinement mode.

- **Boundary isolation test (hard removal).** Let ∂D_j^i denote the boundary cells of region D_j^i. The mean boundary occupancy is computed as

$$\mu_{\partial D_j^i} = \frac{1}{|\partial D_j^i|} \sum_{(x,y)\in \partial D_j^i} M_S^i(x,y). \tag{17}$$

  If $\mu_{\partial D_j^i} \leq \theta_b$, where $\theta_b \in [0,1]$ is a predefined boundary threshold, the candidate is regarded as sufficiently isolated from surrounding occupied structures. In this case, all cells in D_j^i are set to occupancy 0, corresponding to hard removal.
- **Gaussian blending (soft removal).** If $\mu_{\partial D_j^i} > \theta_b$, the candidate may be adjacent to persistent occupied structures, and abrupt cluster-level removal may unnecessarily modify nearby map geometry. Therefore, localized Gaussian blending is applied to attenuate the candidate region more conservatively. The support of the Gaussian kernel is determined by

$$\ell_c = \frac{\ell\, n_b}{\Delta}, \qquad k_c = 2\lceil \ell_c \rceil + 1, \qquad \sigma_c = \eta\, \ell_c, \tag{18}$$

  where n_b is the number of constituent bounding boxes, Δ is the map resolution in m/cell, ℓ is a spatial prior in meters, and $\eta \in (0,1]$ controls the spread of the Gaussian kernel. In our experiments, $\Delta = 0.025$ m/cell, $\ell = 0.01$ m, and $\eta = 0.50$. The blending operation gradually attenuates occupancy values within the candidate region, reducing abrupt modification near occupied structures.

Accordingly, Gaussian blending should be regarded as a conservative attenuation mechanism within the overall hierarchical refinement process rather than as an independent guarantee of static-structure preservation.

Figure 5 illustrates the two context-dependent refinement modes applied after the preceding preservation checks. In Figure 5a, a spatially isolated fused candidate region with low boundary occupancy is processed through hard removal. In Figure 5b, a candidate region adjacent to occupied static structures is refined more conservatively through Gaussian blending. Thus, the local boundary context determines whether the candidate region is removed directly or attenuated gradually.

The cluster-level decision rule is summarized as follows:

$$\text{Removal}(D_j^i) = \begin{cases} \text{None}, & \text{if } \rho_j^i > \theta_u \text{ or } o_j^i \geq \theta_o, \\ \text{Free}(D_j^i), & \text{else if } \mu_{\partial D_j^i} \leq \theta_b, \\ \text{Blend}(D_j^i), & \text{otherwise.} \end{cases} \tag{19}$$

Once cluster-level processing is completed, we further refine the map at the level of individual bounding boxes using $\mathcal{B}_{\text{global}}^i$. This additional step complements the cluster-level decision by removing residual dynamic traces that may remain even when a cluster is retained as a whole. It also handles isolated detections that are not assigned to any cluster, as well as local detections inside preserved clusters that still require separate judgment. The same hard-removal or Gaussian-blending logic is applied but with box-specific criteria and a more localized support.

After all removal and blending operations are completed, we define two subsets of the workspace (Ω): the fully removed region ($\mathcal{R}^i_{\text{free}} \subset \Omega$) and the blended region ($\mathcal{R}^i_{\text{blend}} \subset \Omega$). For cells in $\mathcal{R}^i_{\text{blend}}$, the occupancy is attenuated according to the corresponding Gaussian blending mask. The final probabilistic reliability map is then defined as

$$M^i_R(x,y) = \begin{cases} \mathcal{F}, & \text{if } (x,y) \in \mathcal{R}^i_{\text{free}}, \\ \text{Blend}^i(x,y), & \text{else if } (x,y) \in \mathcal{R}^i_{\text{blend}}, \\ M^i_S(x,y), & \text{otherwise.} \end{cases} \tag{20}$$

As a result, M^i_R preserves stable environmental structures while suppressing dynamic artifacts before map alignment.

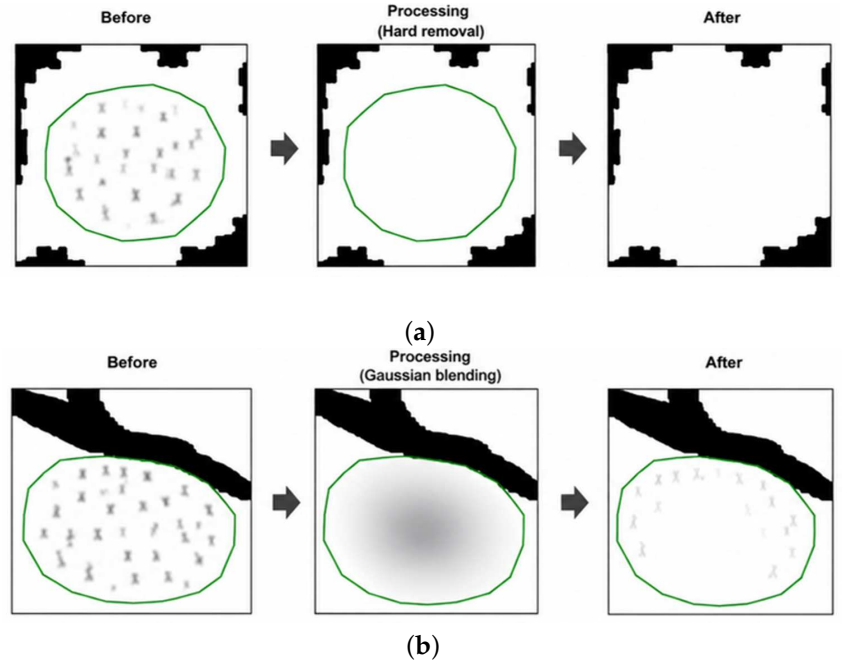


Figure 5. Examples of the two context-dependent refinement modes applied after the preceding preservation checks. (**a**) Hard removal is applied when the fused candidate region is sufficiently isolated from surrounding occupied structures, i.e., when the boundary mean occupancy satisfies $\mu_{\partial D^i_j} \leq \theta_b$. In this case, transient occupancy traces inside the fused candidate polygon are removed. (**b**) Gaussian blending (soft removal) is applied when the fused candidate region is adjacent to occupied static structures, i.e., when $\mu_{\partial D^i_j} > \theta_b$. In this case, occupancy values within the candidate region are gradually attenuated to suppress transient traces while reducing abrupt modification near persistent map structures. The green contours indicate the fused candidate polygons.

2.3.4. Map Alignment and Merging

The reliability-refined maps (M^i_R) generated in the preceding stage are used as inputs to the subsequent SURF-based registration and map-composition process, thereby completing the processing flow from the original local occupancy maps to the merged global map. After generating the probabilistic reliability maps for each robot, we perform map alignment and merging to construct a unified global map. The goal is to estimate the relative spatial transformation between robots and to fuse their occupancy information in a consistent and robust manner. To estimate the spatial transformation between maps, we adopt a conventional feature-based occupancy map-merging pipeline [13] and implement its keypoint-based registration step using the SURF algorithm [26]. Given two reliability-refined maps (M^i_R and M^j_R), SURF descriptors are extracted from both maps, and tentative correspondences are established by nearest-neighbor matching in the descriptor space.

Spurious correspondences are then removed through geometric consistency checks, such as the ratio test and random sample consensus (RANSAC)-based outlier rejection, resulting in a set of inlier matches. Using these inlier correspondences, we estimate the rigid-body transformation composed of rotation and translation that aligns M_R^i to M_R^j. Let $\mathcal{T}_{i\to j}$ denote the resulting transformation. Applying $\mathcal{T}_{i\to j}$ to M_R^i maps it into the coordinate frame of M_R^j. Because the registration is performed on reliability-refined maps, dynamic artifacts have already been attenuated or removed, and the estimated transformation is primarily determined by temporally stable structures. Once the relative transformation is obtained, the maps are merged in the reference frame of one robot or, equivalently, in a common global frame by fusing overlapping occupancy values. In this work, we employ a simple yet effective fusion rule based on probabilistic averaging:

$$\tilde{M}(x,y) = \frac{\sum_k w_k(x,y)\, M_R^k(x,y)}{\sum_k w_k(x,y)}, \tag{21}$$

where $\tilde{M}$ denotes the merged map, M_R^k is the reliability-refined map of robot k transformed into the common frame, and $w_k(x,y)$ is a weight reflecting the local reliability of cell (x,y) in M_R^k, for example, derived from the reliability field or visit frequency. Non-overlapping regions are directly copied from the corresponding local map. This fusion process produces a globally consistent map in which the influence of dynamic objects is substantially reduced.

3. Results

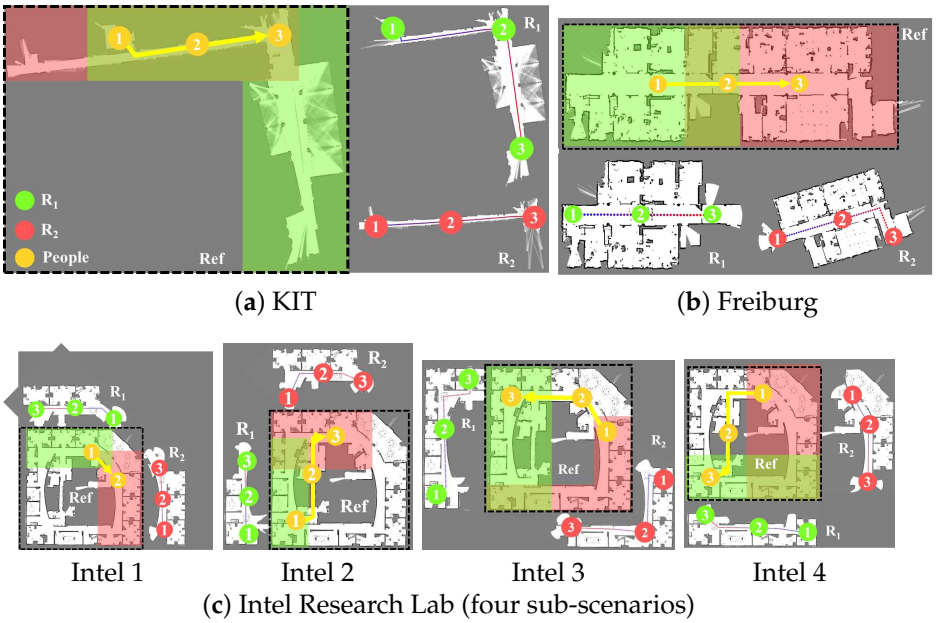
All simulation and real-world experiments were conducted using 2D LiDAR-based occupancy-grid maps generated by an offline SLAM pipeline [27]. The grid resolution was fixed at $\Delta = 0.025$ m/cell. The parameters were empirically selected through preliminary simulations, then kept unchanged across all evaluated scenarios. They were set to $\gamma = 0.03$, $\lambda = 1/30\ \mathrm{s}^{-1}$, $\alpha = 0.45$, $\beta = 0.55$, $\theta_u = 0.50$, $\theta_o = 0.50$, and $\theta_b = 0.55$, with Gaussian blending parameters of $\ell = 0.01$ m and $\eta = 0.50$. The decay rate of $\lambda = 1/30\ \mathrm{s}^{-1}$ corresponds to a characteristic time of $\tau = 30$ s. In the temporal association, γ controls the relative contribution of temporal recency, while λ determines the rate at which stale occupancy evidence relaxes toward the unknown-state value. The α and β thresholds define a symmetric uncertainty interval around 0.5. The θ_u and θ_o thresholds act as cluster-level preservation gates for candidate regions with insufficient spatial evidence or possible static structures, whereas θ_b determines whether the remaining candidates are processed through hard removal or Gaussian blending. The ℓ and η parameters determine the support and spread of the Gaussian blending operation, respectively. The same parameter configuration was used throughout all experiments, without scenario-specific tuning. To directly evaluate the effect of the proposed refinement, the conventional and proposed pipelines used the same SURF-based registration backend, including feature extraction, descriptor matching, RANSAC-based outlier rejection and transformation estimation, and map composition. The conventional pipeline used the original occupancy maps, whereas the proposed pipeline used the reliability-refined maps. Thus, the comparison isolates the contribution of the proposed reliability-refinement stage while keeping the registration backend unchanged.

Each scenario involved two robots, denoted by R_1 and R_2, together with moving humans treated as dynamic objects. The spatial maps generated by R_1 and R_2 are denoted by $M_S^{(R_1)}$ and $M_S^{(R_2)}$, respectively. For quantitative evaluation, the reference alignment used for error computation is denoted by *Ref*. This reference transform was obtained by manually aligning the maps based on static structural landmarks, thereby providing a consistent reference transformation for error analysis. To evaluate the map-merging stage

in isolation, inter-robot communication was disabled during mapping, and no information was exchanged between the robots. All local maps were merged offline on the server after data collection.

3.1. Simulation Scenarios

We evaluate the proposed method in six simulation scenarios constructed from one in-house environment and two widely used SLAM datasets: the KIT Digital Building, Freiburg Building 079, and four layouts derived from the Intel Research Lab map, as shown in Figure 6.



(**a**) KIT (**b**) Freiburg

(**c**) Intel Research Lab (four sub-scenarios)

Figure 6. Overview of the simulation scenarios and dynamic-observation configurations. (**a**) KIT Digital Building scenario. (**b**) Freiburg Building 079 scenario. (**c**) Four sub-scenarios (Intel 1–4) constructed from the same Intel Research Lab occupancy map. The Intel sub-scenarios differ in robot coverage regions, trajectories, human-motion paths, observation times, and partial-overlap configurations, thereby creating different spatial relationships between persistent structural features and transient human-induced occupancy traces. Green and red circles indicate the positions of robots R_1 and R_2, respectively, while yellow circles indicate human positions over time. Robot trajectories are shown as dotted lines, with older positions in blue and newer positions in red. The numbers 1, 2, and 3 inside the circles correspond to the representative timestamps t_1, t_2, and t_3, respectively; the corresponding time values for each scenario are listed in Table 2. The coverage regions of R_1 and R_2 are shown in green and red, respectively, and the reference region (Ref) is indicated by a black dotted rectangle.

Figure 6a shows the simulation scenario based on the KIT Digital Building at the Kumoh National Institute of Technology. This environment contains a substantial overlap between the maps generated by R_1 and R_2. Robot R_1 starts in the central laboratory and moves downward along the east corridor, whereas R_2 moves from left to right along the north corridor. The human moves from the starting point of R_1 to the endpoint of R_2. As a result, strong dynamic interference is introduced into the overlapping region, making conventional feature-based registration highly sensitive to transient footprints. Figure 6b shows the scenario based on Freiburg Building 079. In this case, the human trajectory lies between the paths of R_1 and R_2 rather than near either robot trajectory. Robot R_1 moves from the west side of the building toward the center, while R_2 moves from the center toward the east side. The human travels from the midpoint of the trajectory of R_1 to the midpoint of the trajectory of R_2, thereby generating dynamic clutter in the shared spatial

region. The Intel Research Lab scenarios shown in Figure 6c consist of four sub-cases with different corridor geometries, robot trajectories, and human-motion patterns. In Intel 1, R_1 moves from right to left along the north corridor, while R_2 moves upward along the east corridor, and the human moves from the start point of R_1 to the endpoint of R_2. In Intel 2, R_1 moves upward along the west corridor, while R_2 moves from left to right along the north corridor, with the human again moving from the start of R_1 to the end of R_2. In Intel 3, R_1 moves upward along the west hallway, whereas R_2 moves from the upper part of the east hallway toward the left side of the south hallway, and the human moves from the start of R_2 to the end of R_1. In Intel 4, R_1 moves from right to left along the south corridor, while R_2 moves downward in the east corridor, and the human moves from the start point of R_2 to the goal point of R_1.

Table 2 summarizes the three representative timestamps (t_1–t_3) used to visualize the temporal evolution of the trajectories of R_1, R_2, and the moving humans in each simulation scenario.

Table 2. Simulation timestamps (s). For each dataset, three representative instants (t_1–t_3) are listed for R_1, R_2, and the moving humans. "–" indicates that the corresponding agent is not present or not moving at that time.

	R_1			R_2			People		
Dataset	t_1	t_2	t_3	t_1	t_2	t_3	t_1	t_2	t_3
KIT	0	40	80	20	40	60	20	35	50
Freiburg	0	10	20	15	25	35	10	15	20
Intel 1	0	20	40	40	60	80	30	50	–
Intel 2	0	20	40	35	55	75	30	40	50
Intel 3	25	50	65	0	20	40	10	20	35
Intel 4	10	30	50	0	10	25	5	20	35

3.2. Simulation Results

Figure 7 compares the map-merging results of the proposed method with those of a conventional baseline across the six simulation scenarios. Figure 7a shows the KIT scenario, Figure 7b shows the Freiburg scenario, and Figure 7c presents the four Intel sub-scenarios. The conventional method often produces noticeable misalignments because the moving human acts as a dynamic object and introduces transient artifacts into the occupancy maps. In the Intel cases, these effects appear as different registration failures: Intel 1 exhibits large translational and rotational offsets; Intel 2 shows an approximately 90° orientation error; Intel 3 produces an almost 180° flipped alignment; and Intel 4 exhibits severe translation, together with an approximately 90° rotation error. In contrast, the proposed reliability refinement reduces the influence of these transient artifacts and yields substantially more consistent structural alignment. Similar improvements are also observed in the KIT and Freiburg environments.

Table 3 reports the translation error (E_M), which measures the discrepancy between the translation vector estimated by each method and the reference translation vector. The reference transform $(\mathbf{R}^\star, \mathbf{t}^\star)$ is obtained independently, as described above for *Ref*, and is used only for evaluation. The translation error is defined as

$$E_M = \left\|\mathbf{t}^\star - \hat{\mathbf{t}}\right\|_2 = \sqrt{(\mathbf{t}^\star - \hat{\mathbf{t}})^\mathsf{T}(\mathbf{t}^\star - \hat{\mathbf{t}})}, \tag{22}$$

where $\mathbf{t}^\star$ denotes the translation component of the reference transform and $\hat{\mathbf{t}}$ denotes the translation estimated by the corresponding map-merging method.

Table 4 reports the rotation error (E_θ), which quantifies the difference between the rotation angle estimated by each method and the reference rotation angle ($\theta^\star$) associated

with $\mathbf{R}^\star$ in $(\mathbf{R}^\star, \mathbf{t}^\star)$. Since the rotation is represented as a scalar in this setting, the error is defined as

$$E_\theta = \left|\theta^\star - \hat{\theta}\right|, \tag{23}$$

where $\theta^\star$ denotes the reference rotation angle and $\hat{\theta}$ denotes the rotation angle estimated by the corresponding method.

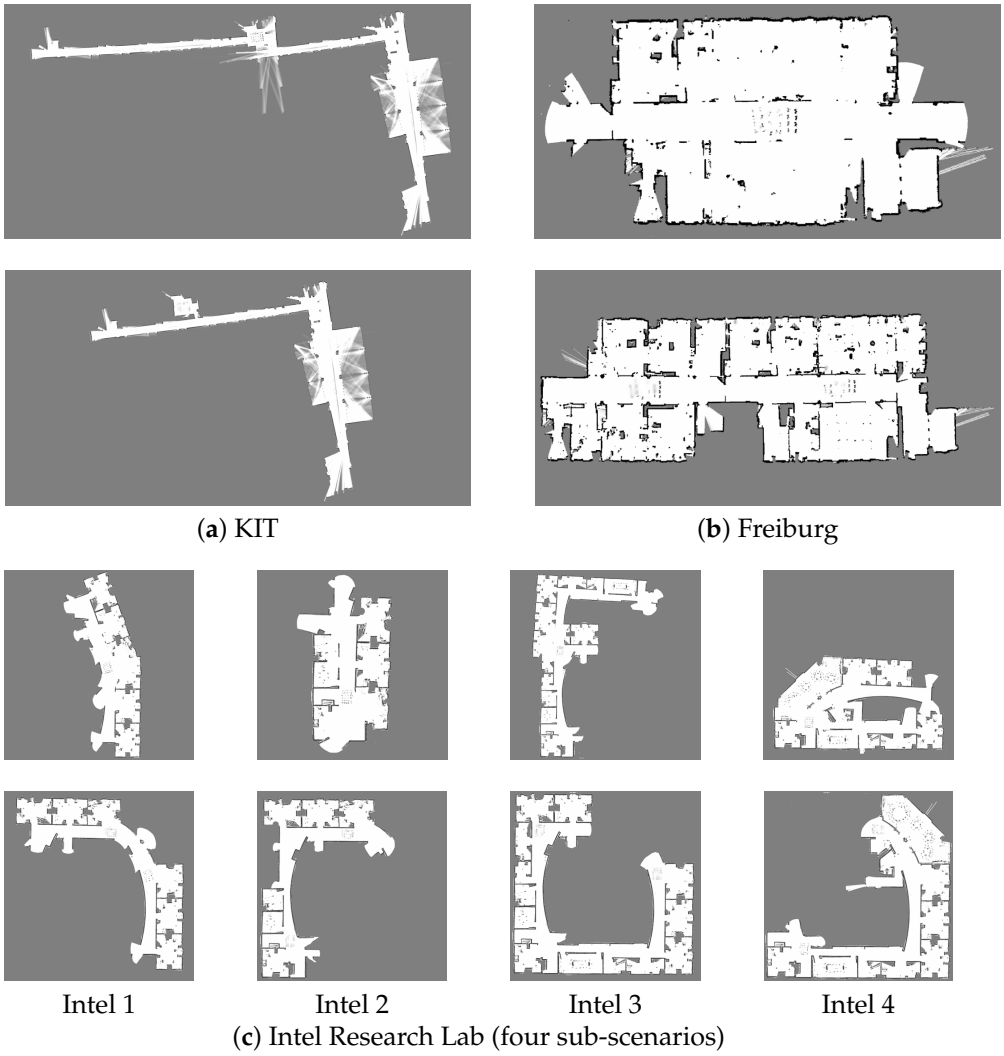


Figure 7. Qualitative comparison of conventional and reliability-refined SURF-based map registration. (**a**) KIT Digital Building scenario. (**b**) Freiburg Building 079 scenario. (**c**) Intel Research Lab sub-scenarios (Intel 1–4). For each scenario, the upper image shows the registration result obtained from the original occupancy maps, whereas the lower image shows the result obtained after reliability refinement using the same SURF-based registration backend. The conventional results for the Intel sub-scenarios exhibit distinct failure modes: Intel 1 shows large translational and rotational offsets, Intel 2 exhibits an approximately 90° orientation error, Intel 3 produces an almost 180° flipped alignment, and Intel 4 exhibits severe translation, together with an approximately 90° rotation error. After reliability refinement, the influence of transient human-induced traces is reduced, allowing persistent corridor and room-boundary structures to be aligned more consistently. Quantitative translation and rotation errors are reported in Tables 3 and 4.

The proposed method consistently outperforms the conventional baseline across all datasets, as shown in Tables 3 and 4. In particular, the translation error is reduced from 1826.90 to 8.09 pixels in Intel 1 and from 3332.65 to 1.05 pixels in Intel 4. Similarly, the rotation error decreases from 69.11° to 0.28° in Intel 1 and from 176.97° to 0.48° in Intel 3. The qualitative comparisons in Figure 7 further support these results: despite

the differences in failure modes observed in Intel 1–4, the proposed reliability refinement reduces the influence of transient human-induced traces and preserves substantially more consistent alignment of persistent structural features.

Table 3. Translation errors (E_M) for the conventional and proposed methods (unit: pixels). Pixel errors can be converted to meters using $E_M[\mathrm{m}] = E_M[\mathrm{px}] \cdot \Delta$, where $\Delta = 0.025$ m/cell.

Dataset	Conventional (E_M)	Proposed (E_M)
KIT	1501.09	13.67
Freiburg	885.80	15.99
Intel 1	1826.90	8.09
Intel 2	1147.44	8.08
Intel 3	2667.47	9.24
Intel 4	3332.65	1.05

Table 4. Rotation errors (E_θ) for the conventional and proposed methods in different environments (unit: degrees).

Dataset	Conventional (E_θ)	Proposed (E_θ)
KIT	6.41	1.82
Freiburg	2.52	1.78
Intel 1	69.11	0.28
Intel 2	92.10	0.14
Intel 3	176.97	0.48
Intel 4	88.13	0.21

3.3. Experimental Settings

Figure 8 shows the hardware configuration used in the real-world experiments. Each platform is a TurtleBot2 equipped with an RPLiDAR A3 for 2D range sensing and an Intellithings iAHRS for inertial measurement, with the IMU mounted adjacent to the LiDAR sensor, as shown in the figure. The two robots navigate independently without communication or information exchange, consistent with the assumptions adopted in the simulation studies. All data, including LiDAR scans, inertial measurements, and odometry, are recorded onboard for subsequent offline processing and map-merging evaluation.

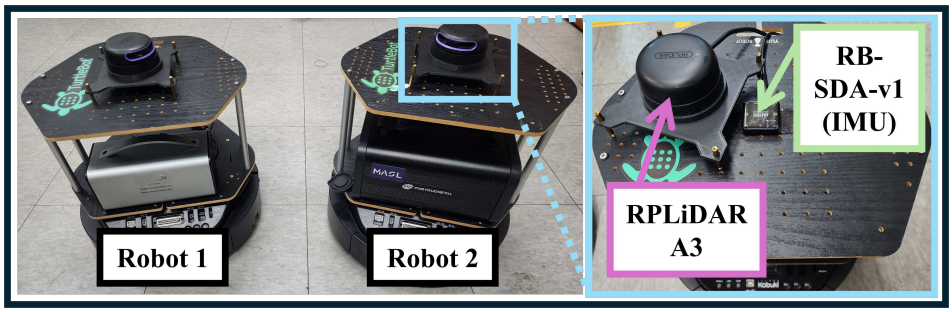


Figure 8. Experimental setup of the TurtleBot2 platform equipped with an RPLiDAR A3 and an Intellithings iAHRS. Two robots are used in the experiments, with the IMU mounted beside the LiDAR sensor.

All map-refinement and merging experiments were processed offline using an Intel NUC 11 equipped with an Intel Core i7-1165G7 processor and 16 GB DDR4 RAM (Intel Corporation, Santa Clara, CA, USA). The additional processing stages introduced by the proposed method consist of CNN-based candidate detection, DBSCAN clustering of detected bounding-box centroids, convex-hull polygonization, and reliability refinement within

candidate regions. CNN inference constitutes the primary additional processing component, while DBSCAN clustering and convex-hull construction operate only on detected candidates and reliability refinement is restricted to the corresponding candidate regions.

The system ran Ubuntu 20.04 and ROS Noetic. Occupancy-grid processing, DBSCAN clustering, convex-hull polygonization, reliability refinement, and SURF-based registration were implemented in C++ using OpenCV, whereas dynamic-region detection was implemented in Python 3.10 using the Ultralytics framework and PyTorch 2.5. The YOLOv8x-based detector contains approximately 68.2 million trainable parameters.

3.4. Experimental Scenario 1

Figure 9 illustrates the scenario and key scenes for Experiment 1 conducted in the Digital Building at Kumoh National Institute of Technology. In Figure 9a, R_1 starts at the entrance of Room A and moves along the corridor toward Room C, whereas R_2 moves from Room A to Room B. Humans initially stand near the entrances of Rooms B and C, then move through the corridor shared by both robots, thereby introducing dynamic clutter into the overlapping region.

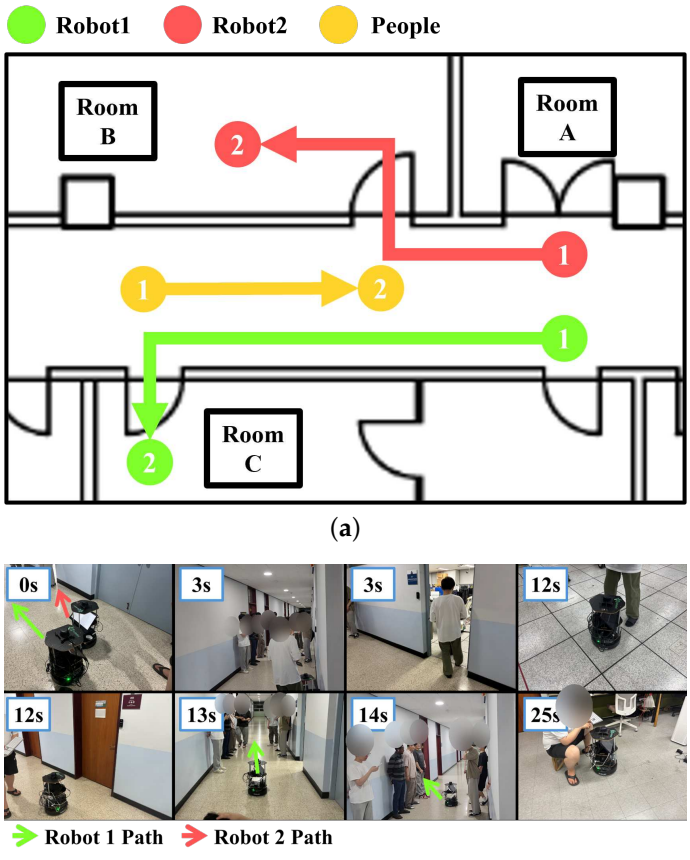


Figure 9. Experiment 1 in the KIT Digital Building. (**a**) Room layout and planned trajectories of R_1, R_2, and humans: R_1 moves from Room A to Room C (green line), R_2 moves from Room A to Room B (red line), and humans traverse the shared corridor between Rooms B and C (yellow line). The numbers 1 and 2 inside the colored circles indicate the start and end positions, respectively, along each planned trajectory. (**b**) Key scenes at representative timestamps (see Table 5), illustrating the sequential motion of R_2 followed by R_1 under changing human positions in the overlapping field of view.

Figure 9b, together with Table 5, summarizes the key moments of Experiment 1. Robot R_2 starts at $t = 0$ s, enters Room B at $t = 3$ s, and stops at $t = 12$ s. At the same time, R_1 begins to move, passes Room B around $t = 13$–14 s, and reaches Room C at

$t = 25$ s. During this period, the humans change their standing positions in the corridor and repeatedly enter and leave the overlapping field of view of the two robots.

Table 5. Key scenes of Experiment 1.

Timestamp	Description
0 s	Robot 2 starts moving
3 s	Robot 2 enters Room B
12 s	Robot 2 stops moving
12 s	Robot 1 starts moving
13 s	Robot 1 passes near Room B
14 s	Robot 1 moves past Room B
25 s	Robot 1 stops moving

3.5. Experimental Scenario 2

Experiment 2 uses the same building layout as Experiment 1 but with a different trajectory for R_2 and a distinct human-motion pattern, as shown in Figure 10. In Figure 10a, R_1 again moves from Room A to Room C. This time, R_2 starts near the left side of Room B, traverses the corridor in front of the room, and stops near the right side of Room B. Meanwhile, humans move from the entrance of Room C toward Room B, repeatedly crossing the corridor segment shared by both robots.

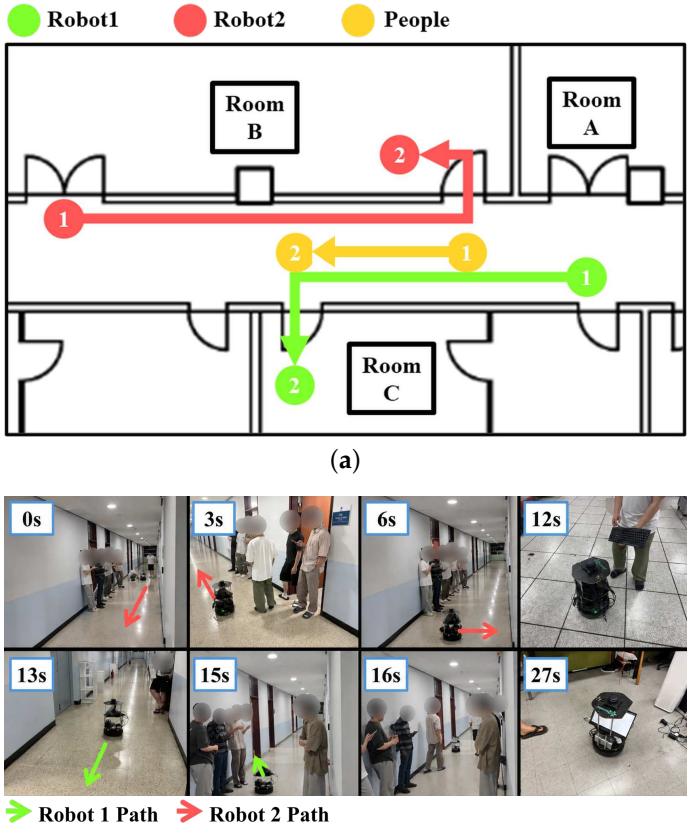


Figure 10. Experiment 2 in the KIT Digital Building. (**a**) Room layout and planned trajectories of R_1, R_2, and humans: R_1 moves from Room A to Room C (green line), R_2 traverses in front of Room B from left to right (red line), and humans move from the entrance of Room C toward Room B, repeatedly crossing the shared corridor (yellow line). The numbers 1 and 2 inside the colored circles indicate the start and end positions, respectively, along each planned trajectory. (**b**) Key scenes at representative timestamps (see Table 6), illustrating the motion of R_2 followed by R_1 under strong human traffic in the overlapping region of their paths.

Figure 10b, together with Table 6, summarizes the main timeline. Robot R_2 starts at $t = 0$ s, passes near Room C and along Room B during $t = 3$–6 s, and stops at $t = 12$ s. Robot R_1 then starts at $t = 13$ s, passes Room B at approximately $t = 15$ s, reaches the front of Room C at $t = 16$ s, and stops at $t = 27$ s. As in Experiment 1, human motion is concentrated in the overlapping region of the two robot trajectories, thereby creating strong dynamic interference.

Table 6. Key scenes of Experiment 2.

Timestamp	Description
0 s	Robot 2 starts moving
3 s	Robot 2 passes near Room C
6 s	Robot 2 is near the right side of Room B
12 s	Robot 2 stops moving
13 s	Robot 1 starts moving
15 s	Robot 1 passes Room B
16 s	Robot 1 is in front of Room C
27 s	Robot 1 stops moving

3.6. Experimental Results

Figure 11 presents the occupancy grids and map-merging results for Experiment 1. The upper row shows the individual occupancy grids and trajectories of R_1 and R_2, whereas the lower row compares the conventional feature-based baseline with the proposed reliability-aware method. Because of the frequency of human motion in the shared corridor, the baseline produces a misaligned merged map: dynamic footprints in the overlapping region generate spurious correspondences and lead to incorrect relative pose estimation. In contrast, the proposed method exploits the probabilistic reliability maps to down-weight regions affected by dynamic objects, resulting in a visually consistent merged map.

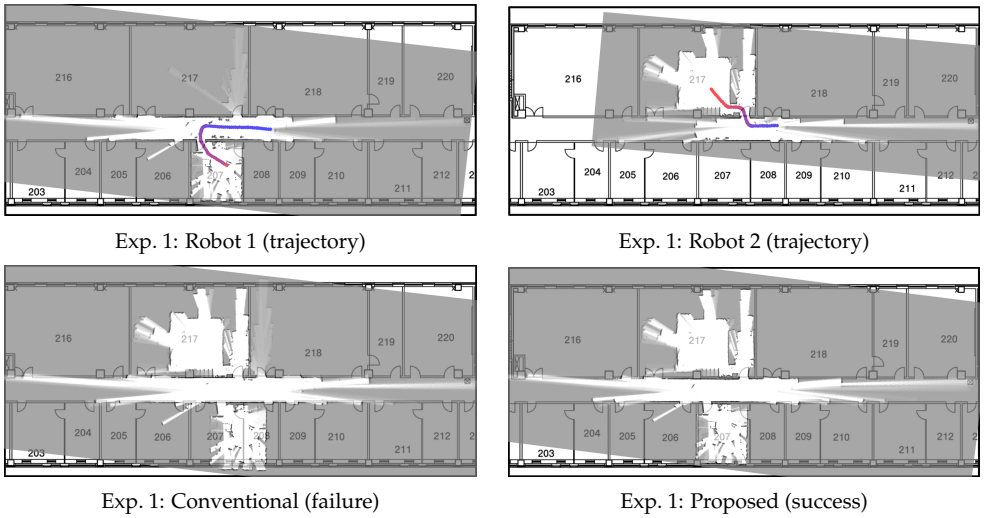


Figure 11. Experiment 1 in the KIT Digital Building. Top: individual occupancy grids with trajectories for R_1 and R_2, with the robot trajectories shown using a blue-to-red color progression, where blue indicates earlier positions and red indicates more recent positions. Bottom: map-merging results of the conventional method (**left**) and the proposed reliability-aware method (**right**). Frequent human motion in the shared corridor leads to severe misalignment for the baseline, whereas the reliability-refined alignment preserves a consistent global structure.

Figure 12 shows the occupancy grids, trajectories, and merging results for Experiment 2. As in Experiment 1, the conventional method is strongly affected by moving humans in the shared corridor and produces a severely misaligned merged map. In con-

trast, the proposed method maintains a consistent alignment between the two local maps despite the presence of dynamic clutter.

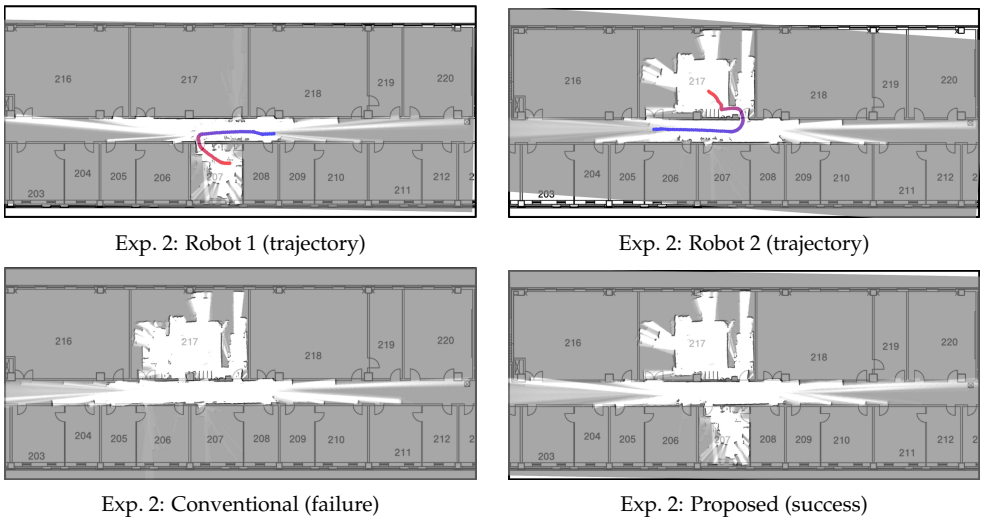


Figure 12. Experiment 2 in the KIT Digital Building under robot and human-motion patterns different from those in Experiment 1. Top: individual occupancy grids for R_1 and R_2, with the robot trajectories shown using a blue-to-red color progression, where blue indicates earlier positions and red indicates more recent positions. Bottom: map-merging results of the conventional method (**left**) and the proposed method (**right**). Despite strong dynamic clutter in the overlapping area, the proposed reliability-aware alignment maintains a coherent merged map, whereas the baseline again fails to align the two grids correctly.

Table 7 reports a quantitative comparison between the conventional and proposed methods in Experiments 1 and 2. In Experiment 1, the translation error (E_M) decreases from 2.42 m to 0.06 m, and the rotation error (E_θ) decreases from 0.23° to 0.15°. In Experiment 2, E_M is reduced from 3.864 m to 0.04 m, while E_θ decreases from 179.17° to 0.37°. These consistent improvements indicate that incorporating temporal decay and probabilistic reliability filtering substantially enhances map-merging robustness. The results further demonstrate that the proposed method remains effective in real indoor environments containing strong dynamic interference.

Table 7. Translation errors (E_M) and rotation errors (E_θ) for the conventional and proposed methods (units: m for E_M and deg for E_θ).

	E_M [m]		E_θ [deg]	
Scenario	**Conventional**	**Proposed**	**Conventional**	**Proposed**
Experiment 1	2.42	0.06	0.23	0.15
Experiment 2	3.864	0.04	179.17	0.37

4. Discussion

The results demonstrate that the proposed reliability-aware refinement improves occupancy-map merging by reducing the influence of temporally unstable human-induced traces before registration. The temporal reliability map provides information on the persistence of detected candidate regions, while the subsequent spatial-preservation checks determine whether and how these regions should be refined. Regions with insufficient spatial evidence or possible static structures are conservatively preserved at the cluster level, while relevant individual detections may still undergo localized refinement. The remaining cluster-level candidates are processed according to their boundary context using hard removal or Gaussian blending. This hierarchical refinement reduces the influence of

transient occupancy evidence while limiting abrupt modification of persistent structures such as walls and room boundaries. Consequently, the same SURF-based registration backend can rely more strongly on reliable map features. The consistent reductions in translation and rotation errors across the simulation and real-world experiments, including the Intel sub-scenarios with different trajectories, human-motion patterns, and overlap configurations, support this interpretation. The benefit of the proposed refinement becomes particularly evident when dynamic clutter strongly affects the overlapping region between maps. In Experiment 2, for example, the conventional pipeline produced a translation error of 3.864 m and a near-180° rotation error, whereas the proposed method reduced these errors to 0.040 m and 0.37°, respectively. This result indicates that suppressing transient occupancy patterns can reduce not only ordinary registration errors but also severe alignment failures caused by unstable feature correspondences. Nevertheless, the effectiveness of feature-based registration still depends on the availability of sufficiently stable and informative structural features in the overlapping region. In highly cluttered or crowd-dominated environments where persistent structural features are limited relative to dynamic clutter, registration performance may still degrade because insufficient reliable correspondences remain.

The present evaluation was intentionally conducted using the same SURF-based registration backend for both the conventional and proposed pipelines. Therefore, the comparison isolates the contribution of the proposed reliability refinement while keeping the registration method unchanged. Because the refinement operates on completed occupancy maps, it can be applied without modifying the original SLAM front end. At the same time, final merging performance remains influenced by the capability of the registration method. Combining the proposed refinement with more advanced registration backends may provide additional robustness in such challenging cases.

The proposed method also has several practical considerations. The temporal association assumes that the completed occupancy map and the corresponding final SLAM trajectory remain locally consistent. Although a common global pose offset does not directly affect this association, substantial local trajectory distortion or an incorrect loop closure may affect timestamp assignments in the corresponding regions. The quality of dynamic-region refinement also depends on the accuracy of the CNN detector, spatial grouping, and the selected parameter configuration. In the present study, the same fixed parameters were used across all evaluated scenarios and produced consistent improvements; however, systematic sensitivity analysis and validation under substantially different map resolutions, sensing rates, and dynamic time scales remain for future study. The current implementation focuses on offline centralized refinement and pairwise merging of two-dimensional indoor occupancy maps, with CNN inference constituting the main additional computational component. Future work will therefore investigate adaptive parameter selection, improved temporal association using per-cell timestamps or pose-uncertainty information, computational optimization for periodic or online operation, and extension to larger multi-robot teams and three-dimensional environments. Comparisons with compatible SLAM methods designed for dynamic environments and robust map-merging approaches will also be considered, along with the integration of complementary structural or semantic information.

5. Conclusions

This paper presented a reliability-aware method for occupancy map merging in dynamic indoor environments. The proposed approach estimates reliability from temporal evidence in sequential observations and integrates two key components: (i) a time-decay occupancy model guided by robot trajectories and (ii) a probabilistic reliability map that

attenuates or removes dynamic, low-confidence regions before alignment. By aligning reliability-refined maps rather than raw occupancy grids, the proposed method emphasizes temporally stable structures and reduces the influence of transient human footprints and other dynamic clutter. Across six simulation scenarios—namely, Intel 1–4, Freiburg, and KIT—the proposed method consistently outperformed the conventional feature-based baseline (Tables 3 and 4, Figure 7). The translation error (E_M) was reduced to below 20 pixels in all cases, which corresponds to less than 0.5 m for $\Delta = 0.025$ m/cell. For example, the translation error decreased from 1826.90 to 8.09 pixels in Intel 1 and from 3332.65 to 1.05 pixels in Intel 4. The rotation error (E_θ) also showed substantial improvement, decreasing from 69.11° to 0.28° in Intel 1 and from 176.97° to 0.48° in Intel 3. The real-world experiments conducted in the KIT Digital Building further confirmed the effectiveness of the proposed method (Figures 11 and 12). In Experiment 1, the translation error decreased from 2.42 m to 0.06 m, while the rotation error decreased from 0.23° to 0.15°. In Experiment 2, the translation error was reduced from 3.864 m to 0.04 m, and the rotation error was reduced from 179.17° to 0.37° (Table 7). In particular, Experiment 2 demonstrated that the conventional pipeline can suffer from a near-180° flip failure under strong dynamic clutter, whereas the proposed reliability-aware filtering successfully preserved the correct alignment. Overall, these results demonstrate that explicit modeling of temporal recency and probabilistic reliability is an effective strategy for improving occupancy map merging in environments containing moving people and other dynamic objects. By suppressing temporally inconsistent occupancy evidence before registration, the proposed method improves both alignment accuracy and robustness. In future work, we plan to extend the proposed framework to online multi-robot map fusion with inter-robot communication and three-dimensional LiDAR maps. We also aim to integrate reliability estimation more tightly with pose-graph optimization and active simultaneous localization and mapping (SLAM) behaviors in larger-scale dynamic environments.

Author Contributions: Conceptualization, H.K. and S.L.; methodology, H.K. and S.L.; software, H.K.; validation, H.K. and S.L.; formal analysis, H.K.; investigation, H.K.; resources, S.L.; data curation, H.K.; writing—original draft preparation, H.K.; writing—review and editing, S.L.; visualization, H.K.; supervision, S.L.; project administration, S.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by Kumoh National Institute of Technology (2024–2026).

Institutional Review Board Statement: Ethical review and approval were waived for this study because the experiments involved non-invasive robot navigation in indoor environments, no sensitive personal data were collected, and no identifiable human information is reported in this paper.

Informed Consent Statement: Written informed consent for publication of the images in this paper was obtained from the individuals appearing in the experimental photographs.

Data Availability Statement: The data presented in this study are available from the corresponding author upon reasonable request. The data are not publicly available because they form part of ongoing research.

Acknowledgments: During the preparation of this manuscript, the authors used ChatGPT (OpenAI, version 5.5) for language refinement and editorial assistance. The authors reviewed and edited the output and take full responsibility for the content of this publication.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

2D	Two-Dimensional
CNN	Convolutional Neural Network
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
IMU	Inertial Measurement Unit
KIT	Kumoh National Institute of Technology
LiDAR	Light Detection and Ranging
OGM	Occupancy Grid Map
RANSAC	Random Sample Consensus
SLAM	Simultaneous Localization and Mapping
SURF	Speeded-Up Robust Features

References

1. Carpin, S.; Birk, A.; Juncikas, V. On map merging. *Robot. Auton. Syst.* **2005**, *53*, 1–14. [CrossRef]
2. Yu, S.; Fu, C.; Gostar, A.K.; Hu, M. A review on map-merging methods for typical map types in multiple-ground-robot SLAM solutions. *Sensors* **2020**, *20*, 6988. [CrossRef] [PubMed]
3. Andersone, I. Heterogeneous map merging: State of the art. *Robotics* **2019**, *8*, 74. [CrossRef]
4. Lee, S.; Kim, H.; Lee, B. An efficient rescue system with online multi-agent SLAM framework. *Sensors* **2020**, *20*, 235. [PubMed]
5. Vaščák, J.; Herich, D. Map merging for multi-robotic applications. In Proceedings of the 2023 IEEE 21st World Symposium on Applied Machine Intelligence and Informatics (SAMI), Herľany, Slovakia, 19–21 January 2023; 21–26.
6. Birk, A.; Carpin, S. Merging occupancy grid maps from multiple robots. *Proc. IEEE* **2006**, *94*, 1384–1397. [CrossRef]
7. Carpin, S. Fast and accurate map merging for multi-robot systems. *Auton. Robot.* **2008**, *25*, 305–316. [CrossRef]
8. Ma, L.; Zhu, J.; Zhu, L.; Du, S.; Cui, J. Merging grid maps of different resolutions by scaling registration. *Robotica* **2016**, *34*, 2516–2531.
9. Topal, S.; Erkmen, I.; Erkmen, A.M. A novel map merging methodology for multi-robot systems. In Proceedings of the World Congress on Engineering and Computer Science (WCECS), San Francisco, CA, USA, 20–22 October 2010; Volume I, pp. 383–387.
10. Jiang, Z.; Zhu, J.; Jin, C.; Xu, S.; Zhou, Y.; Pang, S. Simultaneously merging multi-robot grid maps at different resolutions. *Multimed. Tools Appl.* **2020**, *79*, 14553–14572.
11. Lee, H.; Lee, S. Extended spectra-based grid map merging with unilateral observations for multi-robot SLAM. *IEEE Access* **2021**, *9*, 79651–79662. [CrossRef]
12. Lee, H.; Lee, S. Grid map merging with insufficient overlapping areas for efficient multi-robot systems with unknown initial correspondences. In Proceedings of the 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Toronto, ON, Canada, 11–14 October 2020; pp. 1427–1432.
13. Sunil, S.; Mozaffari, S.; Singh, R.; Shahrrava, B.; Alirezaee, S. Feature-based occupancy map-merging for collaborative SLAM. *Sensors* **2023**, *23*, 3114. [CrossRef] [PubMed]
14. Konolige, K. Large-scale map-making. In Proceedings of the AAAI, San Jose, CA, USA, 25–29 July 2004; pp. 457–463.
15. Sun, Q.; Liao, T.; Du, H.; Zhao, Y.; Chen, C.-C. A method of merging maps for MUAVs based on an improved genetic algorithm. *Sensors* **2023**, *23*, 447. [CrossRef] [PubMed]
16. Chen, G.; Dong, W.; Peng, P.; Alonso-Mora, J.; Zhu, X. Continuous occupancy mapping in dynamic environments using particles. *IEEE Trans. Robot.* **2024**, *40*, 64–84. [CrossRef]
17. Malakouti-Khah, H.; Sadeghzadeh-Nokhodberiz, N.; Montazeri, A. Simultaneous localization and mapping in a multi-robot system in a dynamic environment with unknown initial correspondence. *Front. Robot. AI* **2024**, *10*, 1291672. [CrossRef] [PubMed]
18. Ni, J.; Wang, L.; Wang, X.; Tang, G. An improved visual SLAM based on map point reliability under dynamic environments. *Appl. Sci.* **2023**, *13*, 2712. [CrossRef]
19. Stathoulopoulos, N.; Lindqvist, B.; Koval, A.; Agha-Mohammadi, A.-A.; Nikolakopoulos, G. FRAME: A modular framework for autonomous map merging: Advancements in the field. *IEEE Trans. Field Robot.* **2024**, *1*, 1–26. [CrossRef]
20. Kolla, N.K.; Vundavilli, P.R. A novel SLAM approach for trajectory generation of a dual-arm mobile robot (DAMR) using sensor fusion. *Automation* **2026**, *7*, 42. [CrossRef]
21. de Figueiredo, R.P.; Eriksen, S.N.; Rodriguez, I.; Bøgh, S. A complete system for automated semantic–geometric mapping of corrosion in industrial environments. *Automation* **2025**, *6*, 23. [CrossRef]
22. Salunkhe, P.; Shirole, M.; Mehendale, N. SLAM-based autonomous CO_2 mapping for indoor environmental monitoring: A proof-of-concept framework for multi-parameter hazard assessment. *Automation* **2026**, *7*, 94. [CrossRef]

23. Varghese, R.; Sambath, M. YOLOv8: A novel object detection algorithm with enhanced performance and robustness. In Proceedings of the 2024 International Conference on Advances in Data Engineering and Intelligent Computing Systems (ADICS), Chennai, India, 18–19 April 2024; pp. 1–6.
24. Ester, M.; Kriegel, H.-P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD), Portland, OR, USA, 2–4 August 1996; pp. 226–231.
25. Lei, T.; Luo, C.; Jan, G.; Bi, Z. Deep learning-based complete coverage path planning with re-joint and obstacle fusion paradigm. *Front. Robot. AI* **2022**, *9*, 843816. [CrossRef] [PubMed]
26. Bay, H.; Ess, A.; Tuytelaars, T.; Van Gool, L. Speeded-up robust features (SURF). *Comput. Vis. Image Underst.* **2008**, *110*, 346–359. [CrossRef]
27. Hess, W.; Kohler, D.; Rapp, H.; Andor, D. Real-time loop closure in 2D LiDAR SLAM. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 1271–1278.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.