OCCUPANCY GRID MAPPING BASED ON ICP FOR INDOOR AUTONOMOUS VEHICLES

XÂY DỰNG BẢN ĐỒ LƯỚI CHIẾM DỤNG DỰA TRÊN THUẬT TOÁN ICP CHO PHƯƠNG TIỆN TỰ HÀNH HOẠT ĐỘNG TRONG NHÀ

Cai Viet Anh DUNG, Dang Hoang LAM, Le Thanh TUNG*

Ho Chi Minh City University of Technology and Engineering, Vietnam

*Corresponding author: tunglt@hcmute.edu.vn

(Received: April 04, 2026; Revised: May 07, 2026; Accepted: May 21, 2026)

DOI: 10.31130/ud-jst.2026.24(5A).210E

Abstract - For indoor autonomous vehicle systems, environment mapping is a fundamental step for localization and navigation tasks. This paper presents an occupancy grid mapping method using a 2D LiDAR sensor and the Iterative Closest Point (ICP) algorithm for scan alignment. The ICP algorithm estimates the transformation between consecutive scans to register data into a global coordinate frame. The aligned LiDAR data is then used to update the occupancy grid map based on the log-odds model. The system architecture is modularized into sensor data acquisition, scan alignment, and map updating blocks, improving deployability on embedded platforms with limited resources. Simulation and indoor experimental results with static obstacles demonstrate that the proposed system can generate consistent maps when the robot operates at low to medium speeds.

Key words - Autonomous vehicle; map building; occupancy grid map; LiDAR; Iterative Closest Point

Tóm tắt - Đối với hệ thống xe tự hành trong nhà, việc xây dựng bản đồ môi trường là bước nền tảng cho các bài toán định vị và điều hướng. Bài báo trình bày phương pháp xây dựng bản đồ lưới chiếm chỗ (occupancy grid map) sử dụng cảm biến LiDAR 2D và thuật toán Iterative Closest Point (ICP) để căn chỉnh các lần quét không gian. Thuật toán ICP được sử dụng để ước lượng biến đổi giữa các lần quét liên tiếp nhằm đưa dữ liệu vào hệ tọa độ toàn cục. Dữ liệu LiDAR sau căn chỉnh được sử dụng để cập nhật bản đồ lưới chiếm chỗ theo mô hình log-odds. Kiến trúc hệ thống được mô-đun hóa thành các khối thu thập dữ liệu cảm biến, căn chỉnh dữ liệu quét và cập nhật bản đồ, giúp tăng khả năng triển khai trên các nền tảng nhúng có tài nguyên hạn chế. Kết quả mô phỏng và thử nghiệm trong môi trường trong nhà với các chướng ngại vật cho thấy hệ thống có khả năng xây dựng bản đồ với độ nhất quán cao khi robot di chuyển ở tốc độ thấp đến trung bình.

Từ khóa - Xe tự hành; xây dựng bản đồ; bản đồ lưới chiếm chỗ; LiDAR; Iterative Closest Point

1. Introduction

In indoor autonomous vehicle systems, constructing a map of the working environment is a fundamental step towards robust positioning and navigation. In many indoor scenarios, GPS is unavailable and landmarks are often sparse, requiring the vehicle to rely primarily on on-board range sensors to perceive its surroundings.

Occupancy grid mapping is a standard choice for such environments. The workspace is discretized into cells, where each cell stores the probability of being occupied or free, typically updated using a Bayesian or log-odds model [1]–[3]. Combined with 2D laser range finders, occupancy grids provide a compact representation that supports path planning, collision checking, and localization in structured indoor spaces [2], [4]. Modern SLAM systems extend this idea with particle filters or graph-based optimization, but the underlying cell-wise probabilistic representation remains the core component [3]-[6].

A significant challenge in occupancy grid mapping is efficiently fusing range measurements acquired over time when the vehicle pose is not perfectly known. In practice, many existing approaches either rely on external localization systems or assume sufficiently accurate odometry, which may not hold in low-cost embedded platforms. The Iterative Closest Point (ICP) and its variants are widely used for scan alignment: given two point clouds, ICP iteratively estimates the rigid-body transformation that minimizes a point-to-point or point-to-line distance metric [7], [8]. ICP has been incorporated into many 2D and 3D SLAM frameworks to align laser scans, maintain local maps, and support loop closure [4], [9]–[11]. In 2D indoor mapping, sequential ICP between scans or between scans and a reference map is often sufficient to track vehicle motion when the sensing range is limited and the inter-scan displacement remains moderate [10], [12], [13].

Several works have explored ICP-based occupancy grid mapping for indoor robots. Some approaches couple ICP with line extraction or feature-based matching to improve robustness in corridor-like environments [12]. Others augment ICP with semantic information or robust cost functions to handle dynamic objects and outliers in cluttered scenes [13]. At a higher level, ICP has been used as the front-end of graph-based SLAM pipelines that maintain globally consistent maps through pose graph optimization [4], [6], [10], [14]. Despite these advances, existing solutions often involve computationally intensive optimization or rely on additional sensing modalities, limiting their applicability to resource-constrained systems.

Closer to the context of this work, Vietnamese researchers have begun to apply grid-based mapping and LiDAR-based navigation to assistive and service robots. For example, Ngô et al. propose a safe 2D grid map for indoor wheelchair navigation using a LiDAR-based perception system, with an emphasis on human safety and obstacle avoidance in constrained spaces [15]. Such works demonstrate the feasibility of occupancy grid mapping for assistive mobility devices, but they typically assume an existing localization framework or treat scan alignment as a secondary component relative to higher-level planning.

In this paper, we focus on indoor mapping for a small-scale autonomous vehicle operating in a confined lab environment using only a 2D LiDAR as the primary range sensor. We implement a scan-to-map ICP and log-odds occupancy grid pipeline that runs in real time on embedded hardware, without relying on external localization infrastructure such as motion capture system or UWB beacons. The architecture modularizes LiDAR acquisition, scan filtering, ICP-based pose estimation, and occupancy updating into discrete modules with clearly defined interfaces. Experiments on a mobile robot show that the system can produce consistent indoor maps at low speeds. The main contributions of this paper are summarized as follows: (1) an ICP formulation tuned for short-range 2D indoor mapping; (2) a lightweight log-odds update scheme suitable for embedded processors; and (3) an experimental study analyzing the interaction between grid resolution, ICP parameters, and vehicle speed.

2. Embedded Platform and Sensor Configuration

We implement the proposed mapping pipeline on a small indoor autonomous vehicle based on a Quanser QCar-like platform [16]. The core computing unit is an NVIDIA Jetson TX2 module, which integrates a multi-core ARM CPU and an embedded GPU within a low-power system-on-chip. In this setup, the Jetson TX2 is responsible for high-level tasks, including the LiDAR driver, ICP-based scan registration, and log-odds occupancy grid updates. Conversely, low-level motor control and safety interlocks are handled by dedicated microcontroller boards. Furthermore, the embedded system manages auxiliary functions such as logging data and network communication with a remote display station.

Figure 1. *Experimental platform:*
(1) RPLIDAR A2, (2) NVIDIA Jetson TX2

The platform is equipped with an RPLIDAR A2M8 sensor operating at a scan frequency of 5–15 Hz, producing approximately 533 to 1600 range measurements per scan depending on the rotational speed. The measurement noise is modeled as zero-mean Gaussian with a standard deviation of 4.83 cm, consistent with typical indoor LiDAR performance. To ensure stable real-time processing, only points within a limited local radius are considered in subsequent mapping and registration steps.

A 2D LiDAR sensor is mounted rigidly on the vehicle, positioned slightly above the ground plane. The LiDAR provides range measurements r_i at bearing angles ϕ_i in its local frame, typically at a fixed scan rate. We define the vehicle body frame with the x-axis pointing to the right and the y-axis pointing forward. The LiDAR frame is linked to the body frame by a known rigid transform (R_l, t_l) obtained from mechanical design or simple calibration, so that each LiDAR point can be expressed in the body frame as

$$P_i^{body} = R_l \begin{bmatrix} r_i \cos \phi_i \\ r_i \sin \phi_i \end{bmatrix} + t_l \tag{1}$$

Wheel encoders and an inertial measurement unit (IMU) are used to provide an initial estimate of the vehicle motion between consecutive scans. These estimates are not explicitly integrated into the probabilistic mapping equations; instead, they are used as initial guesses for ICP to accelerate convergence and reduce the risk of convergence to local minima [7], [8]. This loosely coupled design decouples motion prediction from scan matching, enabling ICP to refine the pose estimate and effectively compensate for accumulated drift while maintaining computational simplicity.

All mapping operations are carried out in the global frame, with LiDAR points first transformed to the body frame and then to the world frame using the current estimated pose, ensuring spatial consistency across successive scans and facilitating incremental map construction.

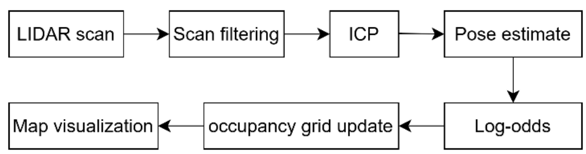

Figure 2*. The overall mapping pipeline*

3. ICP-Based Scan Registration

In the proposed framework, the Iterative Closest Point (ICP) algorithm is responsible for aligning each new LiDAR scan to a reference representation of the environment, thereby providing a refined pose estimate prior to occupancy grid updates. We adopt a point-to-point ICP formulation optimized for 2D scans. The algorithm takes as input a measured 2D scan $P = \{p_i\}_{i=1}^{N}$, a reference point set $Q = \{q_j\}_{j=1}^{M}$, a convergence threshold ε, and a maximum number of iterations I_{max}. In our implementation, the convergence threshold is set to $\varepsilon = 10^{-4}$, and the maximum number of iterations is limited to $I_{max} = 40$ to balance accuracy and computational efficiency. Its output is the rigid-body transformation $T = (R, t)$ that minimizes the following point-to-point least-squares objective:

$$E(R,t) = \sum_{i=1}^{N'} \left\| Rp_i + t - q_{c(i)} \right\|^2 \tag{2}$$

where $q_{c(i)} \in Q$ is the corresponding point for $p_i \in P$. After nearest-neighbor association and correspondence filtering, a subset of N' valid point pairs is retained for optimization, where $N' \leq N$.

At each iteration, ICP performs nearest-neighbor association, assigning every scan point p_i to its closest reference point $q_{c(i)}$. To improve robustness against outliers and ambiguous matches, a one-to-one correspondence constraint is enforced, ensuring that if multiple scan points map to the same reference point, only the pair with the smallest distance is retained, yielding matched subsets of equal size. The centroids $\bar{p} = \frac{1}{N'}\sum_{i=1}^{N'} p_i$ and $\bar{q} = \frac{1}{N'}\sum_{i=1}^{N'} q_i$ are then computed, and the cross-covariance matrix is formed as:

$$K = \sum_{i=1}^{N'} (p_i - \bar{p})(q_i - \bar{q})^T \tag{3}$$

The optimal rigid alignment is obtained in closed form via Singular Value Decomposition (SVD) $K = U \sum V^T$. The rotation R and translation t are derived as:

$$R = VU^T, t = \bar{q} - R\bar{p} \tag{4}$$

The scan is then updated by applying the estimated transform to all its points, $p_i \leftarrow Rp_i + t$, producing an increasingly aligned set. The iteration error is evaluated as:

$$error = \frac{1}{N'}\sum_{i=1}^{N'} \left\| p_i - q_{c(i)} \right\|^2 \tag{5}$$

The procedure repeats until the change in error between successive iterations falls below ε or the iteration limit I_{max} is reached.

After convergence, the estimated transformation (R_k, t_k) is used to express the scan in the world frame and to update the vehicle pose. If ICP returns a transformation that maps body-frame points to the world frame as:

$$p_i^w = R_k p_i^b + t_k \tag{6}$$

then the planar position estimate is simply $\hat{x}_k = t_{k,x}$ and $\hat{y}_k = t_{k,y}$, while the heading is recovered from the rotation matrix:

$$R_k = \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix} \tag{7}$$

as: $\hat{\theta}_k = \text{atan2}(r_{21}, r_{11})$.

For compact representation, the pose can be expressed as a homogeneous transformation:

$$T_k = \begin{bmatrix} R_k & & t_k \\ 0 & 0 & 1 \end{bmatrix} \tag{8}$$

which is used to transform the entire scan into the world frame and serves as the pose output of the localization module.

In our implementation, ICP operates as a LiDAR-based registration module that estimates pose corrections through scan matching, optionally initialized by odometry or IMU measurements. This initialization significantly improves convergence speed and reduces the likelihood of convergence to incorrect local minima. The pose is updated by left-composition:

$$T_k^W = \Delta T . T_{k-1}^W \tag{9}$$

The final transformation T_k^W allows the system to recover the 2D state (x, y, θ) and express the entire scan in the world frame. To further enhance robustness and mitigate drift accumulation, the pipeline employs a two-stage registration strategy: (1) scan-to-scan alignment for local motion tracking, and (2) scan-to-map alignment against a reference submap to maintain global consistency and minimize drift.

For clarity, the ICP algorithm and its integration into the scan-matching-based pose estimation pipeline are illustrated in Figures 3 and 4, respectively.

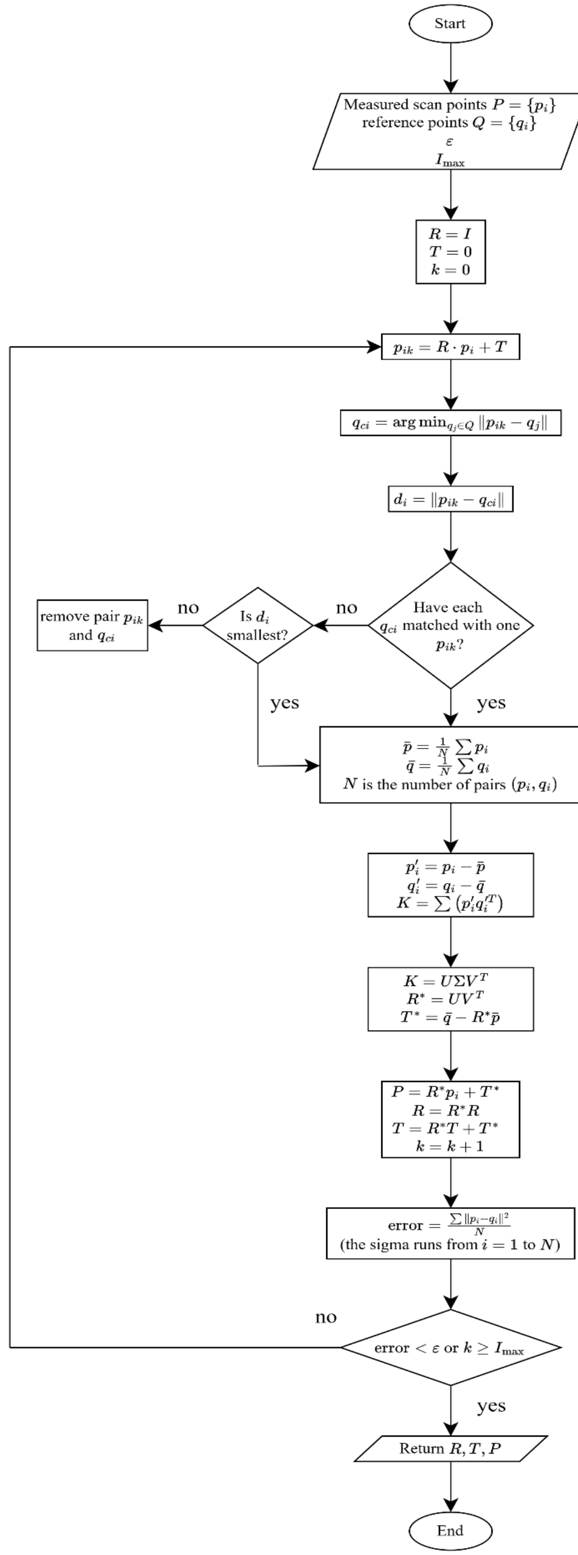

Figure 3. *Flowchart of the ICP-based scan registration process*

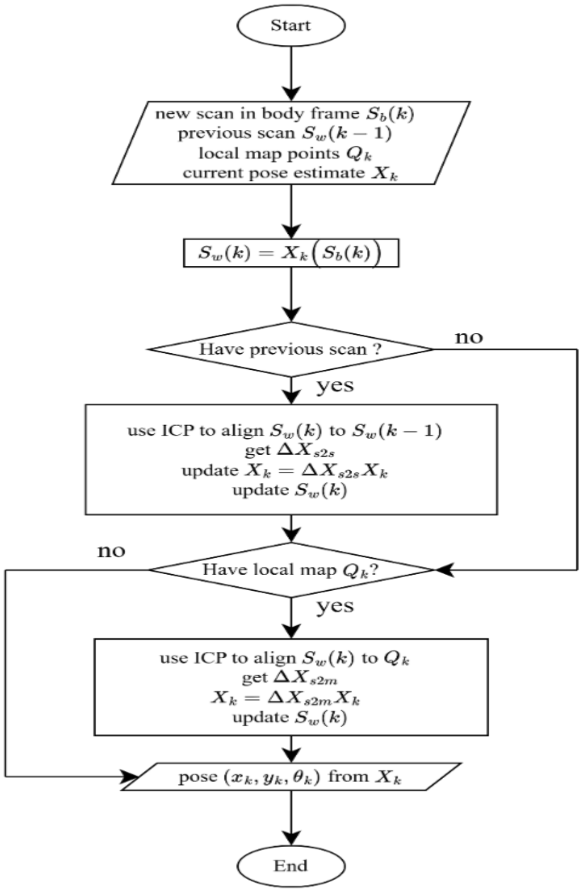

Figure 4. *Flowchart of LiDAR-based localization using scan matching*

4. Occupancy Grid Mapping with Log-Odds

The environment is represented using a 2D occupancy grid map, where the workspace is discretized into square cells m_i, each representing a fixed spatial resolution. In this work, a grid resolution of 5 × 5 cm is employed to achieve a balance between spatial accuracy and computational cost. Each cell stores the probability that it is occupied, $P(m_i)$. To improve computational efficiency and avoid numerical instability, we use the log-odds parameterization:

$$L(m_i) = log\frac{P(m_i)}{1-P(m_i)} \quad (10)$$

This transformation simplifies the recursive Bayesian update into a linear addition of log-odds values [3], [8]. Given a new LiDAR scan transformed into the world frame, an inverse sensor model is utilized to determine which cells along each measurement ray should be updated as "free" (traversed by the beam) and which should be updated as "occupied" (at the beam endpoint where a return is detected) based on ray-casting from the sensor origin to the measured endpoint.

For clarity, the occupancy grid update process using the log-odds formulation is summarized in Figure 5.

Let $L_k(m_i)$ denote the log-odds of cell m after processing scan k, and let l_{occ} and l_{free} be the log-odds increments corresponding to an occupied and free observation, respectively. In our implementation, the log-odds parameters are set to:

$$l_{occ} = \ln\left(\frac{0.7}{0.3}\right), l_{free} = \ln\left(\frac{0.3}{0.7}\right)$$

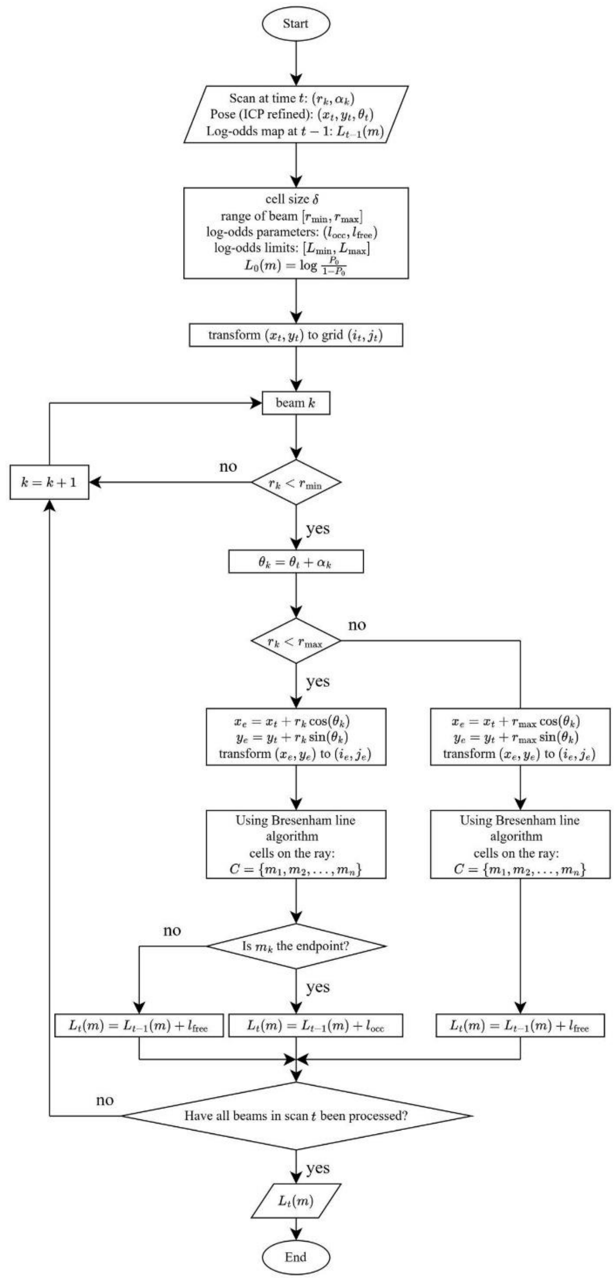

Figure 5. *Flowchart of the log-odds occupancy grid update process*

which reflect a probabilistic preference toward occupancy at beam endpoints and free space along measurement rays. For each cell affected by the scan, the update is formulated as:

$L_k(m_i) = L_{k-1}(m_i) + l_{occ}$: if m_i is at a beam endpoint.

$L_k(m_i) = L_{k-1}(m_i) + l_{free}$: if m_i lies along a beam before the endpoint

Cells that are not affected by the current scan retain their previous values. The log-odds are clipped to a bounded interval $[l_{min}, l_{max}]$ to prevent numerical saturation and to ensure the map remains adaptable to environmental changes.

The parameters l_{occ} and l_{free} are determined based on a probabilistic range-sensor model: a beam endpoint strongly suggests that the corresponding cell is occupied, while cells traversed by the beam are likely free [3]. In practice, they are tuned empirically to balance the rate at which occupancy beliefs change with the measurement noise of the LiDAR in indoor environments.

Once the log-odds map is updated, occupancy probabilities can be recovered using the logistic function:

$$P(m_i) = \frac{1}{1+e^{-L(m_i)}} \tag{11}$$

for visualization or for use by downstream planning modules. Importantly, the mapping stage relies only on the ICP-refined pose estimates and does not modify them; localization and mapping are loosely coupled, with the occupancy grid updated based on the estimated trajectory.

5. Experiment Setup and Results

To evaluate the proposed ICP-based occupancy grid mapping pipeline, we utilize two complementary testbeds. The first is a lightweight Python simulator designed to validate the ICP algorithm's performance. The second is an embedded implementation on a Quanser QCar-like platform operating in a real indoor laboratory environment.

5.1. Simulation-Based ICP Validation

Before integrating ICP into the mapping pipeline, we first tested it in a synthetic scenario. The reference point cloud ("original map") consists of two shapes: a rectangular room and a circular obstacle, sampled along their boundaries (blue points in Figure 8(a)). A second point cloud is obtained by applying a known rigid transformation (rotation and translation) to this map and additive Gaussian noise to every point (red points in Figure 8(a)). This cloud represents a LiDAR scan taken from a different sensor pose. The synthetic environment spans approximately 14 × 14 m before alignment, reflecting the spatial dispersion introduced by the transformation and noise. After ICP convergence, the effective aligned map is reduced to approximately 10 × 10 m due to improved spatial consistency.

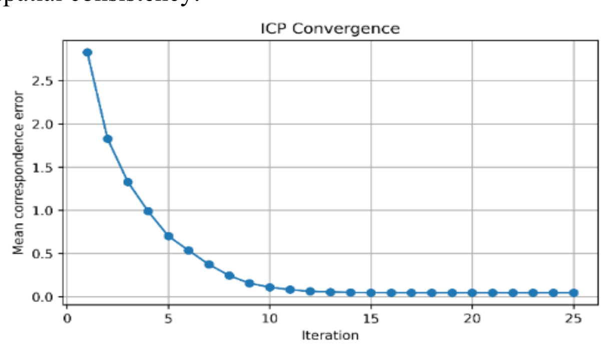

Figure 6. *ICP convergence profile*

ICP is then used to estimate the rigid transformation that aligns the noisy cloud back to the reference. The algorithm starts from a coarse initial estimate and iteratively performs nearest-neighbor matching and least-squares pose estimation until convergence. The convergence behavior of the ICP algorithm is illustrated in Figure 6.

And the distribution of residual distances after convergence is shown in Figure 7.

Figure 8 shows the result: the ICP-aligned cloud (green) almost coincides with the original map (blue) for both the rectangle and the circle. The estimated homogeneous transformation matrix is numerically close to the inverse of the synthetic transform, and the remaining error is consistent with the injected noise.

The convergence behavior illustrated in Figure 6 indicates a rapid reduction in alignment error within the first iterations, followed by gradual refinement toward a stable minimum. Similarly, Figure 7 shows that the residual distances are tightly distributed around zero after convergence, confirming accurate point-to-point alignment.

This experiment confirms that the implemented ICP can reliably register two partially overlapping 2D point sets and is suitable for integration into the scan-alignment module of the embedded system described in Section 5.2.

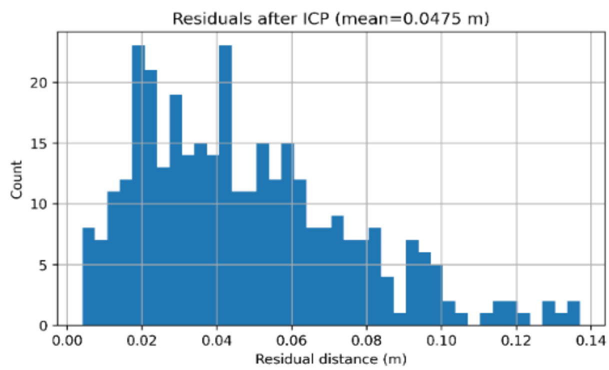

Figure 7. *Residual distance analysis*

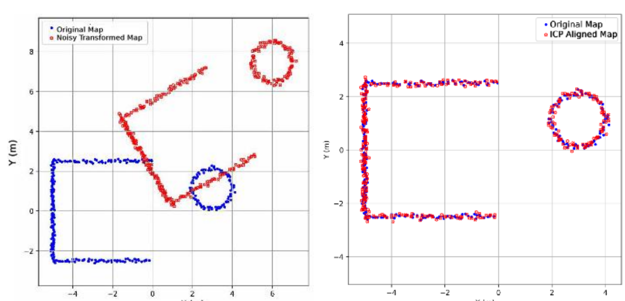

Figure 8. *Simulation results.*
(a) Before ICP; (b) After ICP

To quantitatively evaluate the ICP registration performance, the key metrics are summarized in Table 1.

Table 1. *ICP Registration Performance*

Metric	*Symbol*	Value
Initial Error	e_{first}	2.82559
Final Residual Error	e_{last}	0.047363
Total ICP Iterations	k	25
Homogeneous Transformation Matrix	T	$\begin{bmatrix} 0.867 & -0.498 & 4.01 \\ 0.498 & 0.867 & 5.02 \\ 0 & 0 & 1 \end{bmatrix}$

As shown in Table 1, the ICP algorithm significantly reduces the alignment error, converging from an initial error of 2.82559 to a final residual of 0.047363 after 25 iterations, indicating effective and stable convergence. The estimated transformation matrix closely matches the inverse of the applied synthetic transformation, confirming the accuracy of the recovered pose. The small remaining error is mainly due to the injected Gaussian noise.

Overall, the results demonstrate that the implemented ICP method achieves reliable convergence and sufficient accuracy for real-time scan registration in the proposed mapping framework.

5.2. Embedded Implementation on a Quanser QCar-Like Platform

The same mapping pipeline was then deployed on a small four-wheel platform similar to the Quanser QCar, equipped with a 2D LiDAR and an embedded computer. The onboard software handles LiDAR acquisition, ICP alignment, and log-odds occupancy grid updates. During real-time operation, the LiDAR scan frequency ranges from 5–15 Hz, with 533–1600 points processed per scan to ensure real-time performance on the embedded platform. A separate laptop is used only for visualization and teleoperation, ensuring that all core processing runs onboard.

Experiments took place in an indoor lab with walls, desks, and several static obstacles such as boxes and chairs. The vehicle was driven manually along trajectories that combine straight segments, gentle curves, and turns, at low to medium speeds consistent with safe operation. Based on odometry and IMU measurements, the observed linear velocity is consistently bounded below 0.3 m/s, with typical cruising speed around 0.2 m/s. Wheel encoder and IMU data provide initial estimates of linear and angular velocity; these are used as initial seeds for ICP between consecutive scans. Only measurements within a limited local radius are used to maintain point cloud density and computational efficiency.

The occupancy grids produced onboard capture the main structural features of the environment: walls appear as continuous bands of occupied cells, and large objects form compact clusters. When the estimated trajectory is overlaid, drift remains limited for loops of moderate length, in line with the simulation results. Situations with sharper motion between scans (for example, quick turns or abrupt steering impulses) occasionally lead to poorer ICP convergence, visible as ghosting (double edges) or smearing artifacts in the map. This behavior is consistent with the observed transient increase in scan-to-map error (S2M), particularly during motion transitions, where reduced scan overlap and increased angular velocity degrade registration quality. These artifacts are reduced by filtering outliers in the scan and by restricting ICP to a local neighborhood around the current pose.

Quantitatively, within the operational range $v < 0.3$ m/s, scan-to-scan error (S2S) remains stable in the range 0.02–0.03, indicating consistent local motion estimation. Scan-to-map error (S2M) remains low during steady motion and only exhibits short-lived spikes during turning phases, after which it quickly returns to a stable level, indicating successful re-convergence of ICP and absence of long-term drift accumulation.

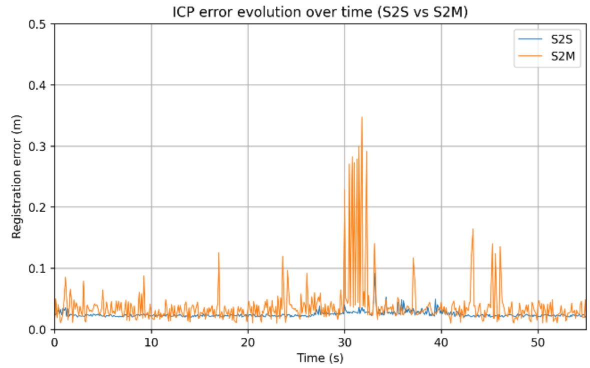

Figure 9. *ICP registration error over time (S2S & S2M)*

Overall, the results demonstrate that the proposed system achieves reliable real-time mapping on embedded hardware, given appropriate parameter selection. Although loop closure and dynamic obstacle handling are not yet included, the generated maps are sufficiently consistent for downstream navigation tasks.

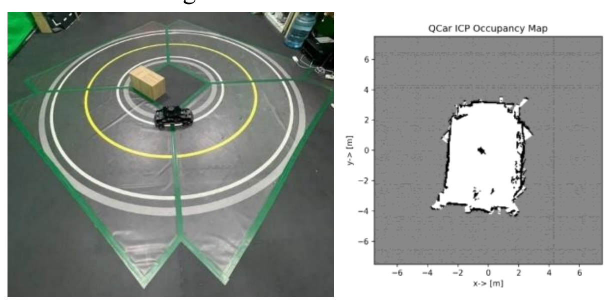

Figure 10. *The correlation between reality and the data collected by the system*

6. Conclusions

This paper presented a lightweight 2-D mapping approach that combines LiDAR-based occupancy grids with ICP scan alignment for indoor autonomous vehicles. The primary strength of the proposed system lies in its high computational efficiency, enabling it to achieve robust pose estimation and consistent map construction while running entirely on low-power embedded hardware. The method improves pose estimation through scan matching and constructs consistent occupancy maps with low computational cost. Experimental results on a small electric vehicle platform demonstrated that the approach reliably handles typical indoor trajectories with minimal drift, achieving a final residual error as low as 0.047. This confirms that the system is a viable solution for real-time indoor navigation without the need for high-end processing units.

Despite these advantages, the current system has several limitations. In particular, the reliance on relative odometry without loop closure may lead to accumulated drift over long trajectories, while the ICP-based scan matching can degrade under rapid motion or in dynamic environments with moving objects. Future work will focus on integrating the mapping module with local motion planning (e.g., DWA) and extending the framework with loop-closure mechanisms or additional sensing modalities to improve scalability in larger environments.

REFERENCES

[1] H. P. Moravec and A. Elfes, "High resolution maps from wide angle sonar", in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, St. Louis, MO, USA, 1985, pp. 116–121.

[2] A. Elfes, "Using occupancy grids for mobile robot perception and navigation", *Computer*, vol. 22, no. 6, pp. 46–57, 1989.

[3] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge, MA, USA: MIT Press, 2005.

[4] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with Rao-Blackwellized particle filters", *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, 2007.

[5] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem", in *Proc. Eighteenth national conference on Artificial intelligent*, CA, United States, 2002, pp. 593–598.

[6] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "g²o: A general framework for graph optimization", in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, 2011, pp. 3607–3613.

[7] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes", *IEEE Transactions on Pattern Analysis and Machine Intelligent*, vol. 14, no. 2, pp. 239–256, 1992.

[8] G. Censi, "An ICP variant using a point-to-line metric for registration of laser scans", in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, Rome, Italy, 2007, pp. 19–25.

[9] D. Kohlbrecher, J. Meyer, T. Graber, and U. Klingauf, "A flexible and scalable SLAM system with full 3D motion estimation", in *Proc. IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, Kyoto, Japan, 2011, pp. 155–160.

[10] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2D LIDAR SLAM", in *Proc. IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden, 2016, pp. 1271–1278.

[11] A. Nüchter, K. Lingemann, J. Hertzberg, and H. Surmann, "6D SLAM—3D mapping with an autonomous mobile robot", *Journal of Field Robotics*, vol. 23, no. 11–12, pp. 777–789, 2006.

[12] S. Lee and S. Yun, "Sequential building of 2D grid map using ICP algorithm and line extraction", in *Proc. International Conference on Control, Automation and Systems (ICCAS)*, Gwangju, South Korea, 2013, pp. 995–998.

[13] D. Parkison and R. M. Eustice, "Semantic iterative closest point through expectation-maximization", in *Proc. British Machine Vision Conference (BMVC)*, Newcastle, U.K., 2018, pp. 1–13.

[14] J. Zhang and S. Singh, "LOAM: Lidar odometry and mapping in real-time", in *Proc. Robotics: Science and Systems Conference (RSS)*, Berkeley, CA, USA, 2014.

[15] B. V. Ngo, T. H. Nguyen, T. N. Nguyen, and M. C. Cao, "A safe 2D grid map for indoor wheelchair navigation using the LiDAR camera system", *Engineering, Technology & Applied Science Research*, vol. 15, no. 5, 2025, pp. 26935-26942.

[16] Quanser Inc., "*QCar Self-Driving Car Studio – Data Sheet*", Quanser Inc., Markham, ON, Canada, Technical Report, 2023.