

Grid-based Submap Joining: An Efficient Algorithm for Simultaneously Optimizing Global Occupancy Map and Local Submap Frames

Yingyu Wang¹, Liang Zhao², and Shoudong Huang¹

Abstract—Optimizing robot poses and the map simultaneously has been shown to provide more accurate SLAM results. However, for non-feature based SLAM approaches, directly optimizing all the robot poses and the whole map will greatly increase the computational cost, making SLAM problems difficult to solve in large-scale environments. To solve the 2D non-feature based SLAM problem in large-scale environments more accurately and efficiently, we propose the grid-based submap joining method. Specifically, we first formulate the 2D grid-based submap joining problem as a non-linear least squares (NLLS) form to optimize the global occupancy map and local submap frames simultaneously. We then prove that in solving the NLLS problem using Gauss-Newton (GN) method, the increments of the poses in each iteration are independent of the occupancy values of the global occupancy map. Based on this property, we propose a pose-only GN algorithm equivalent to full GN method to solve the NLLS problem. The proposed submap joining algorithm is very efficient due to the independent property and the pose-only solution. Evaluations using simulations and publicly available practical 2D laser datasets confirm the outperformance of our proposed method compared to the state-of-the-art methods in terms of efficiency and accuracy, as well as the ability to solve the grid-based SLAM problem in very large-scale environments.

I. INTRODUCTION

Occupancy grid map (OGM) is a widely used map representation in 2D environments because it categorizes the cells in the map into occupied, free, and unknown according to the presence or absence of obstacles in the corresponding environment [1], [2], [3], [4], [5], which is valuable for robot navigation and path planning.

Early works using OGM in 2D laser simultaneous localization and mapping (SLAM) techniques are particle filter based. Such techniques, including FastSLAM [6] and GMapping [7], [8], have the advantage that the occupancy of each cell could be integrated into the state representation, it is made possible as each particle includes both a robot trajectory and an associated OGM. However, due to particle filters maintaining the representation of the full system state in each particle, it is inevitable that particle filter-based methods will take up a lot of memory consumption and computation cost when the robot's trajectory becomes longer.

Optimization-based 2D laser SLAM methods perform better in terms of accuracy and resource consumption. One

challenge in such approaches is to avoid cumulative errors as the robot trajectory grows. Hector SLAM [9] avoids the accumulation of errors by taking the scan-to-map matching method at the front-end of SLAM framework. Karto-SLAM [10] further introduces loop closure detection and implements global optimization using sparse pose adjustment to reduce the long-term error accumulation. Cartographer [11] additionally introduces branch and bound strategy to speed up the loop closure detection. SLAM Toolbox [12] integrated in ROS 2 is developed based on Karto-SLAM, which uses Ceres [13] as the pose graph solver instead of sparse pose adjustment to provide faster and more flexible optimization settings. Moreover, SLAM Toolbox refactors scan matching method and introduces K-D tree search to speed up the computation.

Typically, as in Cartographer, optimization-based methods solve the SLAM problem in two steps. First, the robot poses are optimized using a pose graph, and then the optimized poses are assumed to be the correct poses and used to build up the OGM. However, in these two-step approaches, the uncertainties of the robot poses obtained in the first step are not taken into account when building the map, resulting in suboptimal solutions as pointed in [14], [15], [16], [17].

Our recent work, Occupancy-SLAM [14], demonstrates a substantial improvement in accuracy compared to two-step approaches by simultaneously optimizing the robot poses and the occupancy map. Specifically, it performs a sampling strategy on all laser beams to collect sample points as observations and then constructs error terms between the observations and the global occupancy map. In this approach, both the robot poses and the global occupancy map are considered as optimization variables and the proposed non-linear least squares (NLLS) problem is solved by a variant of the Gauss-Newton (GN) method.

Although Occupancy-SLAM has a great advantage in terms of accuracy, its computational complexity is related to the length of the robot trajectory and the map size because it optimizes all robot poses and the occupancy values of all the cells in the 2D grid based map together. It inevitably falls into the computational bottleneck and potential local minimum as the number of poses and size of the occupancy map grows. This limits the ability of Occupancy-SLAM to solve SLAM problems in large-scale environments.

Local submap joining is a commonly used scheme for SLAM in large-scale environments, because of its efficiency and less chance of being trapped in a local minimum as compared with full optimization based SLAM. Feature-based submap joining approaches [18], [19] optimize local map

¹Yingyu Wang and Shoudong Huang are with the Robotics Institute, Faculty of Engineering and Information Technology, University of Technology Sydney, Australia (e-mail: Yingyu.Wang-1@student.uts.edu.au; Shoudong.Huang@uts.edu.au).

²Liang Zhao is with the School of Informatics, University of Edinburgh, Edinburgh, UK (e-mail: Liang.Zhao@uts.edu.au).

coordinate frames and the feature map simultaneously and can achieve high-level of accuracy. However, existing OGM based submap joining approaches, such as Karto-SLAM [10], SLAM Toolbox [12] and Cartographer [11], only optimize poses but not the grid map, thus the accuracy is sacrificed to some extent. One challenge with OGM based submap joining methods is that optimizing poses and the global map simultaneously greatly affects computational efficiency due to the huge global occupancy map in large-scale environments.

Contribution. This paper considers the grid-based submap joining problem. We propose an efficient pose-only GN method which is equivalent to full GN to solve a NLLS problem where both the global map and the submap frames are optimized together. This is achieved by first proving an independent property of the full GN method for solving the NLLS. Specifically, the contributions of the paper are

- 1) We formulate the grid-based submap joining problem as a NLLS problem where both local submap coordinate frames and the global occupancy map are considered as state variables to be optimized simultaneously.
- 2) We prove that, when solving the formulated NLLS for both poses and the global map using GN method, the increment of poses in each iteration is independent of the occupancy values of the global occupancy map.
- 3) Based on the independent property, we propose a pose-only GN method which is equivalent to full GN method for solving the proposed formulation. The optimal global occupancy map can be obtained using a closed-form formula after the optimal poses are obtained. Experiments demonstrate that the pose-only GN algorithm is very efficient and much faster than the full GN method.
- 4) Experimental results using simulated and practical datasets confirm the superior performance of the proposed method in terms of efficiency and accuracy compared to existing state-of-the-art approaches.

II. PROBLEM FORMULATION: LOCAL GRID-BASED SUBMAP JOINING

In this section, different from [11], [12], we formulate the grid-based submap joining problem as a NLLS problem in which the variables are not the robot poses, but the global grid map and the local submap frames. A description of the local grid-based submap joining problem is shown in Fig. 1.

A. Inputs and Outputs of Submap Joining Problem

For the local submap joining problem, the input is a sequence of grid-based local submaps. First, let us denote $n+1$ submaps as $\mathbf{L} = \{\mathbf{L}_0, \dots, \mathbf{L}_n\}$, where \mathbf{L}_i represents the i -th local occupancy map built by any evidence grid mapping technique such as [3], [11], [14]. Thus, occupancy values of cells in these submaps are the logarithm of odds (the ratio between the probability of being occupied and the probability of being free). Different from [11], we do not maintain all the robot poses in each submap. The only poses we care about are the submap frames. Assume that the coordinate frames of these local submaps in the global map coordinate frame

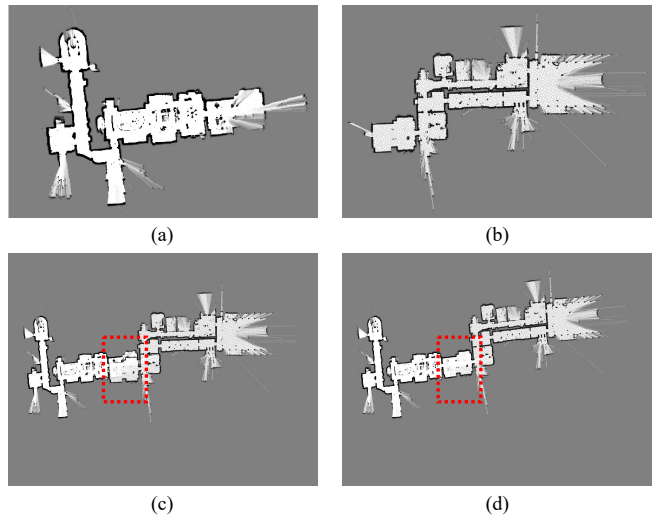


Fig. 1: *The description of the grid-based submap joining problem.* (a) and (b) show the grid-based submaps. (c) shows the global grid map generated by fusing the two submaps using the pose information only. (d) shows the consistent global grid map generated by our proposed grid-based submap joining method.

are $\{\mathbf{X}_0^r, \dots, \mathbf{X}_n^r\} \in SE(2)^{n+1}$, where $\mathbf{X}_i^r = [t_i^T, \theta_i]^T$ represents the i -th local submap coordinate frame. t_i is the x - y position and θ_i is the orientation with the corresponding rotation matrix $\mathbf{R}_i = \begin{bmatrix} \cos(\theta_i) & \sin(\theta_i) \\ -\sin(\theta_i) & \cos(\theta_i) \end{bmatrix}$.

The task of our submap joining problem is to find the optimal poses of local submap coordinate frames and the optimal global grid map. We denote the global grid map as $\{\mathbf{M}(\mathbf{m}_1), \dots, \mathbf{M}(\mathbf{m}_{l_w \times l_h})\}$, where \mathbf{m}_j ($1 \leq j \leq l_w \times l_h$) represents the coordinate of a discrete cell in the global map and $\mathbf{M}(\mathbf{m}_j)$ represents its corresponding occupancy value [20]. Therefore, the outputs of submap joining problem are the optimal solution of local submap coordinate frames and the optimal global map.

B. Global to Local Coordinate Projection

The grid cell \mathbf{m}_j in the global map \mathbf{M} can be projected to i -th local submap coordinate by pose \mathbf{X}_i^r , i.e.,

$$\mathbf{p}_{im_j} = \mathbf{R}_i(\mathbf{m}_j \cdot s - \mathbf{t}_i), \quad (1)$$

where \mathbf{p}_{im_j} is the projected position in the local submap coordinate, and s is the resolution of cells in the global occupancy map \mathbf{M} (the distance between two adjacent cells represents s meters in the real world).

C. The NLLS Formulation

The state vector of the proposed map joining problem is

$$\mathbf{X} = [(\mathbf{X}^r)^T, \mathbf{M}^T]^T, \quad (2)$$

where

$$\begin{aligned} \mathbf{X}^r &= [(\mathbf{X}_1^r)^T, \dots, (\mathbf{X}_n^r)^T]^T \\ \mathbf{M} &= [\mathbf{M}(\mathbf{m}_1), \dots, \mathbf{M}(\mathbf{m}_{l_w \times l_h})]^T. \end{aligned} \quad (3)$$

As with most submap joining problem formulations, we fix the first local map coordinate frame as the global coordinate. Therefore, \mathbf{X}^r consists of n local map coordinate frames and \mathbf{M} includes $l_w \times l_h$ discrete cells of global occupancy map.

By the global-to-local projection relationship, all cells of global occupancy map \mathbf{M} can be projected to corresponding submaps to compute the difference in occupancy values to formulate the NLLS problem, i.e., minimize

$$f(\mathbf{X}) = \sum_{i=0}^n \sum_{j \in \{obs_i\}} \left\| \omega_{im_j} \mathbf{M}(\mathbf{m}_j) - \mathbf{L}_i(\mathbf{p}_{im_j}) \right\|^2, \quad (4)$$

where $\{obs_i\}$ is the set including the indices of cells in the global occupancy map \mathbf{M} projected onto local submap \mathbf{L}_i . $\mathbf{M}(\mathbf{m}_j)$ refers to the occupancy value of the cell \mathbf{m}_j in the global occupancy map \mathbf{M} . $\mathbf{L}_i(\mathbf{p}_{im_j})$ means the occupancy value of the continuous coordinate \mathbf{p}_{im_j} in the local submap \mathbf{L}_i . Here we use bilinear interpolation to obtain the occupancy value $\mathbf{L}_i(\mathbf{p}_{im_j})$ on the discrete grid-based submap \mathbf{L}_i , similar to [9], [14].

In (4), ω_{im_j} is the weight to establish an accurate relationship between the global map and local submaps w.r.t. occupancy values of corresponding coordinates because both local submaps and the global map are represented by the Bayesian approach [21]. It can be calculated by

$$\omega_{im_j} = \frac{\mathbf{N}^{L_i}(\mathbf{p}_{im_j})}{\mathbf{N}^M(\mathbf{m}_j)}. \quad (5)$$

Here, \mathbf{N}^{L_i} is the local hit map associated with submap \mathbf{L}_i . Each cell in \mathbf{N}^{L_i} describes the observation count of the corresponding cell in \mathbf{L}_i , which is similar to [14]. $\mathbf{N}^{L_i}(\mathbf{p}_{im_j})$ approximates the observation count of the continuous coordinate \mathbf{p}_{im_j} using bilinear interpolation based on the observation count of surrounding discrete cells. \mathbf{N}^M denotes the global hit map associated with \mathbf{M} which is built by projecting all the local hit maps $\{\mathbf{N}^{L_i}\}$ into the global map coordinate frame through $\{\mathbf{X}_i^r\}$. Thus, ω_{im_j} is a function of $\{\mathbf{X}_i^r\}$.

III. OUR APPROACH: EFFICIENT POSE-ONLY GN ALGORITHM

In this section, we prove the existence of a special independent property when solving our NLLS formulation (4) using GN method. Based on this special property, we propose a pose-only GN algorithm equivalent to full GN method to solve our NLLS problem.

A. Iterative Solver to the NLLS Formulation

Our submap joining problem is to seek \mathbf{X} to minimize $f(\mathbf{X})$ in (4). Typical iterative algorithms, such as GN method, can be used to solve (4) iteratively, i.e., by starting with an initial guess $\mathbf{X}(0)$ and updating with $\mathbf{X}(k+1) = \mathbf{X}(k) + \Delta(k)$. The update vector $\Delta(k) = \left[\Delta^r(k)^T, \Delta^M(k) \right]^T$ is the solution to

$$\mathbf{J}^T \mathbf{J} \Delta(k) = -\mathbf{J}^T F(\mathbf{X}(k)) \quad (6)$$

where $F(\mathbf{X})^T F(\mathbf{X}) = f(\mathbf{X})$ and \mathbf{J} is the Jacobian matrix $\partial F / \partial \mathbf{X}$ evaluated at $\mathbf{X}(k)$. However, for grid-based submap joining problem, the number of cells $l_w \times l_h$ in the global map \mathbf{M} is very large which leads to a high dimension of the state vector, so common methods to solve (6) are time-consuming.

B. A Special Independent Property of Our Formulation

Our NLLS formulation (4) can be shown to have a special property, when optimizing poses and the global occupancy map together using GN method, the optimization of poses is independent of the global occupancy map.

Proposition. *For each step of GN iteration for minimizing the NLLS problem in (4), the increment of poses $\Delta^r(k)$ is independent of the occupancy values of the global occupancy map \mathbf{M} .*

Proof. The Jacobian matrix \mathbf{J} consists of two parts, i.e. the Jacobian of $F(\mathbf{X})$ w.r.t. the poses \mathbf{J}_r , and the Jacobian of $F(\mathbf{X})$ w.r.t. the global occupancy map \mathbf{J}_M , i.e. $\mathbf{J} = [\mathbf{J}_r \quad \mathbf{J}_M]$. All non-zero elements of \mathbf{J}_M are ω_{im_j} which only depend on poses (hit maps \mathbf{N}^M , \mathbf{N}^{L_i} are independent of occupancy values), and thus \mathbf{J}_M is independent of \mathbf{M} . In addition, it can be deduced from (1) and (4) that \mathbf{J}_r is also independent of \mathbf{M} .

Let us rewrite (6) as

$$\begin{bmatrix} \mathbf{U} & \mathbf{W} \\ \mathbf{W}^T & \mathbf{V} \end{bmatrix} \begin{bmatrix} \Delta^r \\ \Delta^M \end{bmatrix} = \begin{bmatrix} \mathbf{b}^r \\ \mathbf{b}^M \end{bmatrix} \quad (7)$$

where $\mathbf{U} = \mathbf{J}_r^T \mathbf{J}_r$, $\mathbf{V} = \mathbf{J}_M^T \mathbf{J}_M$, $\mathbf{W} = \mathbf{J}_r^T \mathbf{J}_M$, $\mathbf{b}^r = -\mathbf{J}_r^T F(\mathbf{X})$, and $\mathbf{b}^M = -\mathbf{J}_M^T F(\mathbf{X})$. According to the theory of the Schur complement [22], the solution of (7) can be calculated by

$$\left(\mathbf{U} - \mathbf{W} \mathbf{V}^{-1} \mathbf{W}^T \right) \Delta^r = \mathbf{b}^r - \mathbf{W} \mathbf{V}^{-1} \mathbf{b}^M \quad (8)$$

and

$$\mathbf{V} \Delta^M = \mathbf{b}^M - \mathbf{W}^T \Delta^r. \quad (9)$$

If we rewrite (8) as

$$\begin{aligned} & \left(\mathbf{J}_r^T \mathbf{J}_r - \mathbf{J}_r^T \mathbf{J}_M \left(\mathbf{J}_M^T \mathbf{J}_M \right)^{-1} \mathbf{J}_M^T \mathbf{J}_r \right) \Delta^r \\ & = -\mathbf{J}_r^T F(\mathbf{X}) + \mathbf{J}_r^T \mathbf{J}_M \left(\mathbf{J}_M^T \mathbf{J}_M \right)^{-1} \mathbf{J}_M^T F(\mathbf{X}), \end{aligned} \quad (10)$$

it can be seen that the left-hand side of (10) is not related to \mathbf{M} because both \mathbf{J}_M and \mathbf{J}_r are independent of \mathbf{M} .

Furthermore, we can rewrite (4) as

$$f(\mathbf{X}) = \|F(\mathbf{X})\|^2 = \|\mathbf{J}_M \mathbf{M} - H(\mathbf{X}^r)\|^2, \quad (11)$$

where $H(\mathbf{X}^r) = [\dots, \mathbf{L}_i(\mathbf{p}_{im_j}), \dots]$ is not related to \mathbf{M} . Using $F(\mathbf{X})$ in (11), the right-hand side of (10) becomes

$$\begin{aligned} & -\mathbf{J}_r^T (\mathbf{J}_M \mathbf{M} - \mathbf{J}_M (\mathbf{J}_M^T \mathbf{J}_M)^{-1} \mathbf{J}_M^T \mathbf{J}_M \mathbf{M}) \\ & -\mathbf{J}_r^T (\mathbf{J}_M (\mathbf{J}_M^T \mathbf{J}_M)^{-1} \mathbf{J}_M^T H(\mathbf{X}^r) - H(\mathbf{X}^r)). \end{aligned} \quad (12)$$

It can be deduced that the first item of (12) is equal to 0, and the second item is not related to \mathbf{M} because \mathbf{J}_M , \mathbf{J}_r and $H(\mathbf{X}^r)$ are all independent of \mathbf{M} . Thus, the right-hand side of (10) is not related to \mathbf{M} . Therefore, the increment

Algorithm 1: Our Pose-only GN Algorithm

Input: Submaps $\{\mathbf{L}_i\}$, local hit maps $\{\mathbf{N}^{L_i}\}$, and initial poses $\mathbf{X}^r(0)$

Output: Optimized poses $\hat{\mathbf{X}}^r$ and the optimized global occupancy map $\hat{\mathbf{M}}$

- 1: Initialize global occupancy map $\mathbf{M}(0)$, global hit map $\mathbf{N}^M(0)$ using $\mathbf{X}^r(0)$, $\{\mathbf{L}_i\}$ and $\{\mathbf{N}^{L_i}\}$
 - 2: **for** $k = 0$; $k \leq \tau_k$ & $\|\Delta^r(k)\|^2 \geq \tau_\Delta^r$; $k++$ **do**
 - 3: Evaluate $H(\mathbf{X}^r)$ and Jacobian $\mathbf{J}_r, \mathbf{J}_M$ at $\mathbf{X}^r(k)$
 - 4: Solve linear system (13) to get $\Delta^r(k)$
 - 5: Update poses $\mathbf{X}^r(k+1) = \mathbf{X}^r(k) + \Delta^r(k)$
 - 6: Recalculate the global hit map $\mathbf{N}^M(k+1)$ using $\mathbf{X}^r(k+1)$ and $\{\mathbf{N}^{L_i}\}$
 - 7: **end for**
 - 8: $\hat{\mathbf{X}}^r = \mathbf{X}^r(k+1)$
 - 9: Solve closed-form formula (15) to get the optimized global occupancy map $\hat{\mathbf{M}}$
-

of poses Δ^r is independent of \mathbf{M} and it can be calculated by

$$\begin{aligned} & \left(\mathbf{J}_r^T \mathbf{J}_r - \mathbf{J}_r^T \mathbf{J}_M (\mathbf{J}_M^T \mathbf{J}_M)^{-1} \mathbf{J}_M^T \mathbf{J}_r \right) \Delta^r \\ & = \mathbf{J}_r^T H(\mathbf{X}^r) - \mathbf{J}_r^T \mathbf{J}_M (\mathbf{J}_M^T \mathbf{J}_M)^{-1} \mathbf{J}_M^T H(\mathbf{X}^r). \end{aligned} \quad (13)$$

For the point feature based SLAM problem with known data association and under the assumption that the observation noises covariance matrices are isotropic, a similar property was proved in [23]. In this paper, we prove that this independent property also holds in the proposed grid-based submap joining problem without the assumption of data association. \square

C. Equivalent Pose-only GN Iteration Algorithm

According to the Proposition proved in Section III-B, we propose a pose-only GN algorithm equivalent to full GN method to solve our NLLS problem. The equivalent pose-only GN algorithm is given in Algorithm 1. In Algorithm 1, we iteratively solve (13) using Jacobian \mathbf{J}_r and \mathbf{J}_M to obtain the increments of the poses Δ^r and update poses, and then recalculate the global hit map \mathbf{N}^M based on the updated poses since \mathbf{J}_M depends on the global hit map. After the optimal solution of poses $\hat{\mathbf{X}}^r$ is obtained, our NLLS problem (4) becomes a linear least squares problem which minimize

$$g(\mathbf{M}) = (\mathbf{J}_M \mathbf{M} - H(\hat{\mathbf{X}}^r))^T (\mathbf{J}_M \mathbf{M} - H(\hat{\mathbf{X}}^r)). \quad (14)$$

Therefore, the optimal solution of the global occupancy map $\hat{\mathbf{M}}$ can be calculated by the closed-form formula

$$\hat{\mathbf{M}} = \mathbf{V}^{-1} \mathbf{J}_M H(\hat{\mathbf{X}}^r). \quad (15)$$

For Algorithm 1, the dimension of the sparse linear system to be solved at each iteration only depends on the number of poses, whereas in the full GN algorithm, it depends on the number of poses and the number of cells in the global map. For the grid-based submap joining problem in which the number of poses is much less than that of cells of the global

map, our pose-only GN algorithm can easily and quickly obtain the solution of Δ^r for each iteration. After optimal poses $\hat{\mathbf{X}}^r$ are obtained, the optimized global map $\hat{\mathbf{M}}$ can be easily obtained by the closed-form formula (15).

It should be noted that, although our pose-only GN does not need to update the map during iterations, the constraint information associated with the map is still implicitly taken into account when optimizing the poses, thus our method is quite different from pose-graph optimization.

IV. EXPERIMENTAL RESULTS

In this section, we demonstrate the accuracy and efficiency of our algorithm. In the experiments, we compare our results with Cartographer [11], SLAM Toolbox [12] and Occupancy-SLAM [14], which are the current state-of-the-art algorithms and perform significantly better than other methods such as Hector-SLAM [9], Karto-SLAM [10], etc.

First, due to the lack of ground truth in the practical datasets, we evaluate our algorithm qualitatively and quantitatively using datasets generated from two simulated environments. Second, we validate and evaluate the performance of our method through six publicly available practical datasets, including three datasets in large-scale environments. Parameters of all used datasets are listed in Table I. All OGMs are generated by poses from different approaches and corresponding observations using the same evidence grid mapping technique. In addition, we use Occupancy-SLAM to build our local submaps for all the experiments, in which the map resolution is the default setting in [14] for simulation experiments and low-resolution for practical experiments.

TABLE I: Parameters of Datasets.

Dataset	No. Scans	Duration (s)	Map Size (m)	Odometry
Simulation 1	3640	117	50 × 50	yes
Simulation 2	2680	83	50 × 50	yes
Car Park [24]	1642	164	50 × 40	yes
C5 [14]	3870	136	50 × 40	yes
Museum b0 1G [11]	5522	152	85 × 95	no
Museum b2 [11]	51833	1390	250 × 200	no
Museum b0 EG [11]	22650	615	225 × 150	no
C3	24402	610	150 × 125	no

A. Simulation Experiments

We use varying levels of nonlinearity, non-convex obstacles, and long corridors to design two different simulation experiments. For all simulated datasets, each scan consists of 1081 laser beams with angles ranging from -135 degrees to 135 degrees which simulates a Hokuyo UTM-30LX laser scanner. To simulate real-world data acquisition, we add random Gaussian noises with zero-mean and standard deviation of 0.02 m to each beam of the scan data generated from the ground truth. Similarly, we add zero-mean Gaussian noises to the odometry inputs generated from the ground truth poses (standard deviation of 0.04 m for x-y and 0.003 rad for orientation). For each simulation environment, we generate 5 datasets with different sets of random noises.

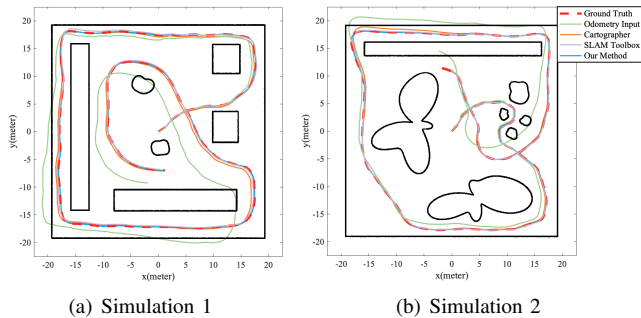


Fig. 2: *Simulation environments and robot trajectory results.* (a) and (b) show the simulation environments (the black lines indicate the obstacles in the scene) and the trajectories of ground truth, odometry inputs, Cartographer [11], SLAM Toolbox [12], and our method for one dataset in each of the two simulation experiments. Results from Occupancy-SLAM [14] are visually identical to our method and not drawn in the figure.

The robot trajectory estimation results of our method, SLAM Toolbox and Cartographer of one dataset in each simulation are compared with the ground truth and odometry in Fig. 2. Since no difference can be visualized between our trajectory and the trajectory from Occupancy-SLAM, the trajectory from Occupancy-SLAM is not drawn. It is obvious that our trajectories are closer to the ground truth trajectories.

The quantitative comparison of errors in the estimated poses from Cartographer, SLAM Toolbox, Occupancy-SLAM and our method is given in Table II, in which we use mean absolute error (MAE) and root mean squared error (RMSE) to evaluate the translation errors (in meters) and rotation errors (in radians) for the 5 runs in each simulation. It should be noted that for SLAM Toolbox, since only the poses of submap frames are involved in the pose graph optimization, here we only use those poses of the submap frames to evaluate the pose errors. Compared to Cartographer and SLAM Toolbox, our approach performs better on all metrics, with errors about 3 to 12 times smaller. The significant reduction in error validates the effectiveness of our proposed method. Occupancy-SLAM performs best in terms of accuracy due to the use of the full least squares strategy, which is the best one can achieve but is costly.

Fig. 3 shows the OGMs and point cloud maps generated by poses from ground truth, Cartographer, SLAM Toolbox, Occupancy-SLAM and our approach, where the first two rows are the results of a dataset in Simulation 1, and the last two rows are the results of a dataset in Simulation 2. The areas highlighted by red dots show that the results of our method are better than the results of Cartographer and SLAM Toolbox. It is clear that the object boundaries of both OGMs and point cloud maps obtained by our method are much clearer than Cartographer and SLAM Toolbox. Also, it can be seen that the maps of our method are very close to Occupancy-SLAM. This suggests that our submaps joining method, as an approximation scheme of full optimization based SLAM, can obtain similar results as the

TABLE II: Comparison of Robot Pose Errors in Simulations.

	Odom	Carto	Toolbox	Occupancy-SLAM*	Ours
Simulation 1					
MAE (Trans/m)	0.7827	0.2534	0.1465	0.0238	0.0453
MAE (Rot/rad)	0.0491	0.0139	0.0134	0.0008	0.0012
RMSE (Trans/m)	0.9840	0.2992	0.1788	0.0263	0.0609
RMSE (Rot/rad)	0.0551	0.0156	0.0152	0.0009	0.0015
Simulation 2					
MAE (Trans/m)	0.7535	0.1426	0.0493	0.0070	0.0151
MAE (Rot/rad)	0.0518	0.0068	0.0085	0.0006	0.0011
RMSE (Trans/m)	0.9687	0.1878	0.0626	0.0106	0.0205
RMSE (Rot/rad)	0.0593	0.0091	0.0127	0.0009	0.0018

Red and the **blue** indicate the best and second best results respectively.

* Occupancy-SLAM uses the full least squares strategy including all the robot poses, which can theoretically achieve the highest accuracy. However, it is very costly and cannot handle large-scale environments. For Simulation 1 and Simulation 2, Occupancy-SLAM takes about 20 times longer than our submap joining approach. The time consumptions on some practical datasets are given in Table. IV.

full optimization based approach Occupancy-SLAM.

For the quantitative comparison of occupancy maps, we use the results from all the ten datasets in Simulation 1 and Simulation 2 to compare the accuracy of the OGMs. We use AUC (Area under the ROC curve) [25] and precision to evaluate our method, Cartographer, SLAM Toolbox and Occupancy-SLAM, where ground truth labels are given by OGMs generated using ground truth poses. When evaluating using AUC, we remove all the unknown cells because AUC is a metric for binary classification as in [25]. The results are given in Table III. Compared with Cartographer and SLAM Toolbox, it can be seen that our method achieves better performance in both metrics, which proves our approach can obtain more accurate maps.

TABLE III: Accuracy of the Occupancy Grid Maps.

		AUC	Precision
Simulation 1	Cartographer	0.9088	0.9565
	SLAM Toolbox	0.9550	0.9749
	Occupancy-SLAM	0.9997	0.9973
	Ours	0.9751	0.9867
Simulation 2	Cartographer	0.9270	0.9660
	SLAM Toolbox	0.9726	0.9871
	Occupancy-SLAM	0.9966	0.9991
	Ours	0.9949	0.9979

B. Comparisons Using Practical Datasets

First, we use three practical datasets used in [14], namely *Deutsches Museum b0 IG* [11], *Car Park* [24] and *C5* [14], to compare our method with Cartographer and Occupancy-SLAM. The results are shown in Fig. 4 and Fig. 5. It is clear that our results are very close to those of Occupancy-SLAM. Compared with Cartographer's results, both the OGMs and point cloud maps of our results have significantly clearer object boundaries. These results show the accuracy of our submap joining method using practical datasets. Since SLAM Toolbox does not support the *MultiEchoLaserScan* sensor message format of Cartographer's companion *Deutsches Museum* dataset [11] and performs poorly on *Car Park* and *C5* datasets because no loop closures are detected

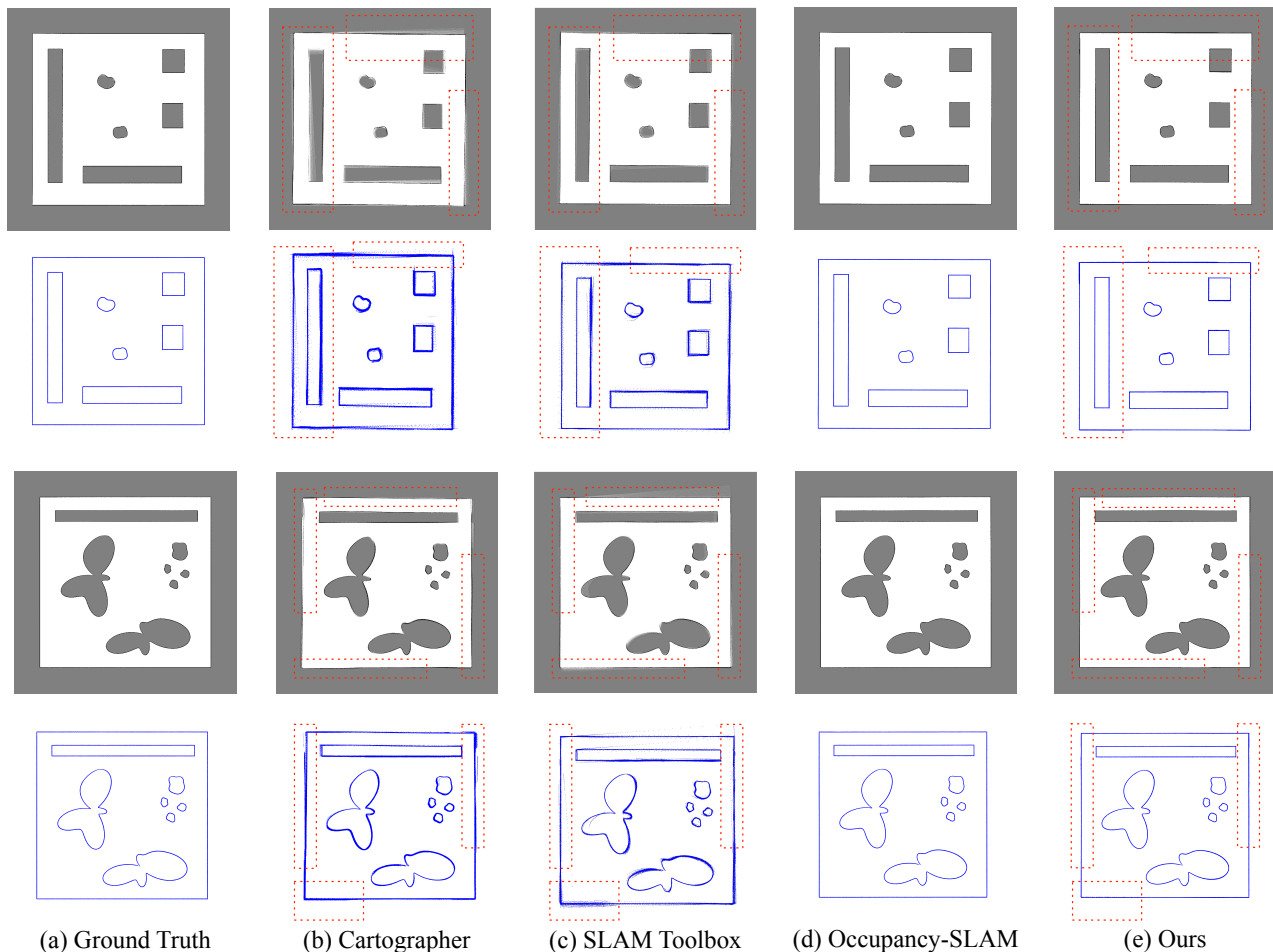


Fig. 3: The OGMs and point cloud maps generated by poses from ground truth, Cartographer, SLAM Toolbox, Occupancy-SLAM and our approach in one dataset for Simulation 1 (first two rows) and Simulation 2 (last two rows).

to perform pose graph optimization, we do not present the results of the SLAM Toolbox for this and subsequent experiments.

Our method can handle large-scale environments well, which is impossible with Occupancy-SLAM due to its increasing computational complexity and tendency to fall into local minima as the trajectory and map size grow. To validate the effectiveness of our algorithm in large-scale environments, three large-scale environment datasets are utilized, namely *Deutsches Museum b0 EG*, *Deutsches Museum b2*, and *C3*, and their map sizes range from $150 \text{ m} \times 125 \text{ m}$ to $250 \text{ m} \times 200 \text{ m}$. Here, we only compared our method with Cartographer, the results of OGMs are shown in Fig. 6, and it can be observed that our results are better than Cartographer, especially for Fig. 6(c) and Fig. 6(f), where the right wall of our results are clearly closer to the straight line whereas that of the Cartographer’s results are somewhat curved.

C. Time Consumption

We use all the six practical datasets to evaluate the time consumption of our approach, Cartographer and Occupancy-

SLAM, the results are presented in Table IV (all elapsed times are evaluated using an Intel Core i7-1370P processor). The huge time consumption of Occupancy-SLAM is due to the high computational complexity of batch optimization using all the scans. Thus, it cannot handle the three large-scale datasets. In contrast, the proposed submap joining algorithm is very efficient, taking less than 1 second for the three normal-scale datasets and around 2 to 6 seconds for the three large-scale datasets. Since Cartographer and Occupancy-SLAM can solve the complete OGM based SLAM problems from scans, we also list the elapsed time of submap building. The total time of submap building and joining is still faster than that of Cartographer on most datasets. It should be mentioned that this total time consumption can be further reduced by using other efficient submap building methods, and the submaps can also be built out-of-core as in [26].

The high efficiency of the proposed map joining method is mainly due to the independent property of our formulation. Because of this, the pose-only GN method can be used to solve the NLLS problem efficiently. We compare our pose-only GN with the full GN where both poses and global map are solved at the same time, and the results are depicted in

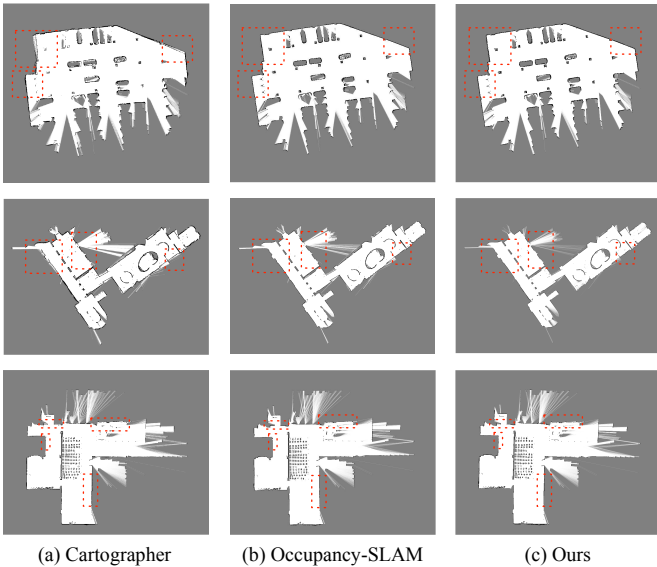


Fig. 4: *OGMs from Cartographer, Occupancy-SLAM and our method.* The first to third rows are *Car Park, Deutsches Museum b0 1G* and *C5* dataset, respectively.

TABLE IV: Time Consumption (*in seconds*) of Algorithms

Dataset	Carto	Occupancy-SLAM	Ours		
			Submap	Joining	Total
Car Park	168	17138	133	0.8	133.8
Museum b0 1G	152	26901	113	0.6	113.6
C5	146	27452	157	0.6	157.6
Museum b2	1424	-	1248	2.4	1250.4
Museum b0 EG	615	-	539	1.7	540.7
C3	610	-	737	5.2	742.2

Table V. It is clear that the calculation speed of our pose-only algorithm is much faster than the full GN algorithm.

TABLE V: Time Consumption (*in seconds*) of Each Iteration*

Dataset	Ours	Full GN	Acceleration Rate
Car Park	0.062	0.174	281%
Museum b0 1G	0.044	0.098	223%
C5	0.036	0.058	161%
Museum b0 EG	0.157	0.419	267%
Museum b2	0.226	0.378	167%
C3	0.096	0.174	181%

*Since our algorithm is equivalent to the full GN algorithm and thus has the same number of iterations, we only compare the time consumption per iteration.

V. CONCLUSIONS

This paper formulates the grid-based submap joining problem as a non-linear least squares problem to simultaneously optimize the global occupancy map and local submap frames, which is different from other submap joining approaches that only optimize poses by constructing a pose graph. More importantly, we show that for our formulation the existence of a special independent property helps solve this problem very efficiently by a pose-only Gauss-Newton algorithm. Our

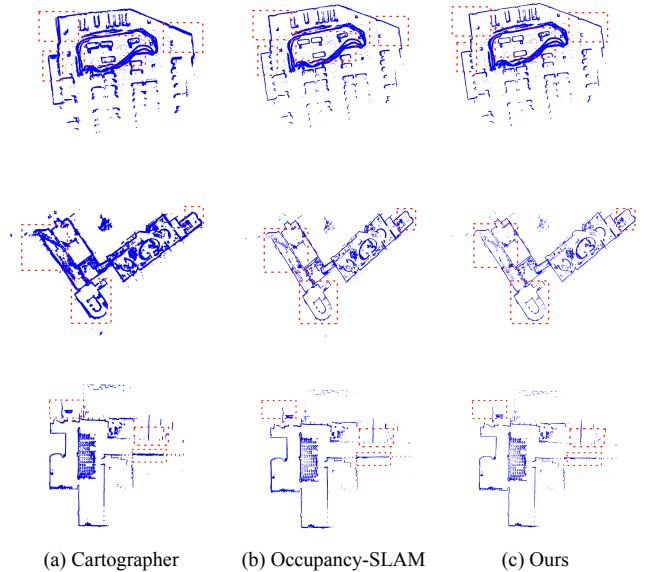


Fig. 5: *Point cloud maps from Cartographer, Occupancy-SLAM and our method.* The point cloud maps are generated by projecting the scan endpoints using poses.

local submap joining approach is evaluated using datasets generated from two simulated environments and six practical datasets, demonstrating that it outperforms state-of-the-art algorithms in terms of speed and accuracy.

This paper is the first to simultaneously optimize the global occupancy map and local map coordinate frames for the grid-based submap joining problem. In the experiments of this paper, we have used 3-10 local submaps depending on the scale of datasets. The best number of local maps for map joining (to achieve the best efficiency) and the best way to partition the whole laser dataset for building different high-quality local maps require further investigation, and these are left to our future work. We also plan to extend our approach to 3D cases to perform large-scale 3D grid map based SLAM more accurately and efficiently.

REFERENCES

- [1] H. Moravec and A. Elfes, "High resolution maps from wide angle sonar," in *Proceedings of 1985 IEEE International Conference on Robotics and Automation*, vol. 2. IEEE, 1985, pp. 116–121.
- [2] H. P. Moravec, "Sensor fusion in certainty grids for mobile robots," in *Sensor Devices and Systems for Robotics*. Springer, 1989, pp. 253–276.
- [3] A. Elfes, *Occupancy grids: A probabilistic framework for robot perception and navigation*. Carnegie Mellon University, 1989.
- [4] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous Robots*, vol. 34, no. 3, pp. 189–206, 2013.
- [5] S. Huang, K. Pan, and G. Dissanayake, "Perception in complex and unstructured infrastructure environments," *Infrastructure Robotics: Methodologies, Robotic Systems and Applications*, pp. 33–58, 2024.
- [6] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit *et al.*, "Fastslam: A factored solution to the simultaneous localization and mapping problem," *AAAI/IAAI*, vol. 593598, 2002.
- [7] G. Grisetti, C. Stachniss, and W. Burgard, "Improving grid-based slam with rao-blackwellized particle filters by adaptive proposals and selective resampling," in *Proceedings of the 2005 IEEE international conference on robotics and automation*. IEEE, 2005, pp. 2432–2437.

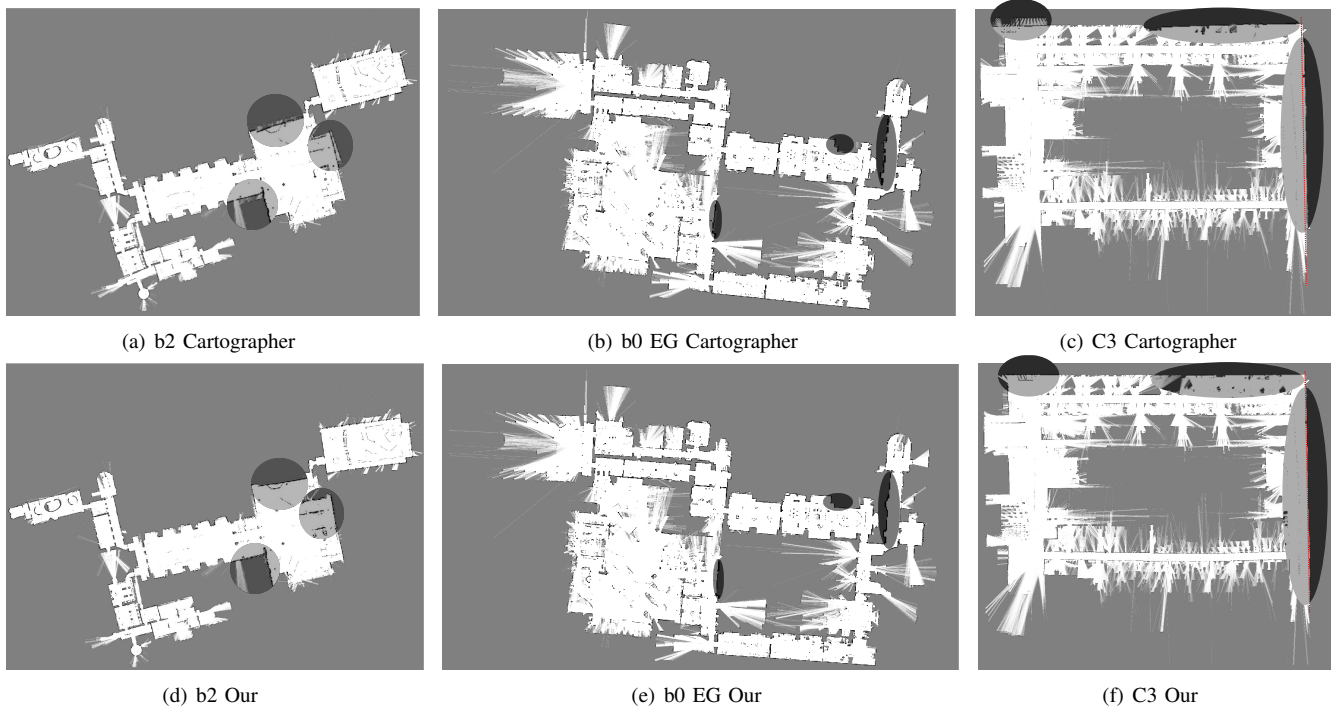


Fig. 6: Comparison of the results of our method and Cartographer in three large-scale environments. The highlighted areas show our method outperforms Cartographer. The red dotted lines in (c) and (f) are reference lines, and it can be seen that the right wall of Cartographer’s result is more curved while our result is closer to the straight line. Occupancy-SLAM fails with such large-scale environments.

[8] —, “Improved techniques for grid mapping with rao-blackwellized particle filters,” *IEEE transactions on Robotics*, vol. 23, no. 1, pp. 34–46, 2007.

[9] S. Kohlbrecher, O. Von Stryk, J. Meyer, and U. Klingauf, “A flexible and scalable slam system with full 3d motion estimation,” in *Proceedings of 2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*. IEEE, 2011, pp. 155–160.

[10] K. Konolige, G. Grisetti, R. Kümmerle, W. Burgard, B. Limketkai, and R. Vincent, “Efficient sparse pose adjustment for 2d mapping,” in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2010, pp. 22–29.

[11] W. Hess, D. Kohler, H. Rapp, and D. Andor, “Real-time loop closure in 2d lidar slam,” in *Proceedings of 2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2016, pp. 1271–1278.

[12] S. Macenski and I. Jambrecic, “Slam toolbox: Slam for the dynamic world,” *Journal of Open Source Software*, vol. 6, no. 61, p. 2783, 2021.

[13] S. Agarwal, K. Mierle, and T. C. S. Team, “Ceres Solver,” 10 2023. [Online]. Available: <https://github.com/ceres-solver/ceres-solver>

[14] L. Zhao, Y. Wang, and S. Huang, “Occupancy-SLAM: Simultaneously Optimizing Robot Poses and Continuous Occupancy Map,” in *Proceedings of Robotics: Science and Systems*, New York City, NY, USA, June 2022.

[15] F. Dellaert and M. Kaess, “Square root sam: Simultaneous localization and mapping via square root information smoothing,” *The International Journal of Robotics Research*, vol. 25, no. 12, pp. 1181–1203, 2006.

[16] A. Rosinol, A. Violette, M. Abate, N. Hughes, Y. Chang, J. Shi, A. Gupta, and L. Carlone, “Kimera: From slam to spatial perception with 3d dynamic scene graphs,” *The International Journal of Robotics Research*, vol. 40, no. 12-14, pp. 1510–1546, 2021.

[17] V. Reijnders, A. Millane, H. Oleynikova, R. Siegwart, C. Cadena, and J. Nieto, “Voxgraph: Globally consistent, volumetric mapping using signed distance function submaps,” *IEEE Robotics and Automation Letters*, vol. 5, no. 1, pp. 227–234, 2019.

[18] L. Zhao, S. Huang, and G. Dissanayake, “Linear slam: A linear solution to the feature-based and pose graph slam based on submap joining,” in *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 2013, pp. 24–30.

[19] J. Wang, J. Song, L. Zhao, S. Huang, and R. Xiong, “A submap joining algorithm for 3d reconstruction using an rgb-d camera based on point and plane features,” *Robotics and Autonomous Systems*, vol. 118, pp. 93–111, 2019.

[20] M. C. Martin and H. P. Moravec, “Robot evidence grids.” The Robotics Institute, Carnegie-Mellon University, Tech. Rep., 1996.

[21] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. MIT Press, 2005.

[22] F. Zhang, *The Schur complement and its applications*. Springer Science & Business Media, 2006, vol. 4.

[23] L. Zhao, Z. Mao, and S. Huang, “Feature-Based SLAM: Why Simultaneous Localisation and Mapping?” in *Proceedings of Robotics: Science and Systems*, Virtual, July 2021.

[24] J. Zhao, L. Zhao, S. Huang, and Y. Wang, “2d laser slam with general features represented by implicit functions,” *IEEE Robotics and Automation Letters*, vol. 5, no. 3, pp. 4329–4336, 2020.

[25] A. P. Bradley, “The use of the area under the roc curve in the evaluation of machine learning algorithms,” *Pattern recognition*, vol. 30, no. 7, pp. 1145–1159, 1997.

[26] K. Ni, D. Steedly, and F. Dellaert, “Tectonic sam: Exact, out-of-core, submap-based slam,” in *Proceedings 2007 IEEE International Conference on Robotics and Automation*. IEEE, 2007, pp. 1678–1685.