SDF-Loc: Signed Distance Field based 2D Relocalization and Map Update in Dynamic Environments

Mingming Zhang, Yiming Chen, and Mingyang Li

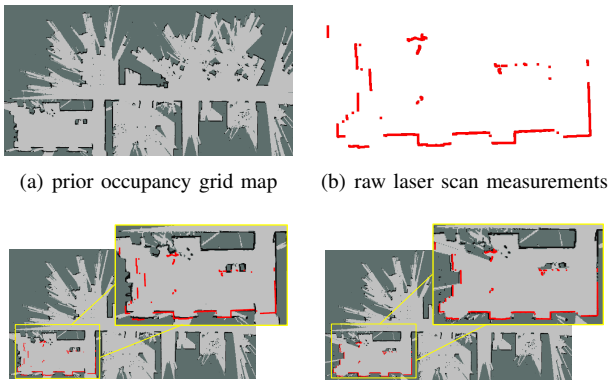
***Abstract*— To empower an autonomous robot to perform long-term navigation in a given area, a concurrent localization and map update algorithm is required. In this paper, we tackle this problem by providing both theoretical analysis and algorithm design for robotic systems equipped with 2D laser range finders. The first key contribution of this paper is that we propose a hybrid signed distance field (SDF) framework for laser based localization. The proposed hybrid SDF integrates two methods with complementary characteristics, namely Euclidean SDF (ESDF) and Truncated SDF (TSDF). With our framework, accurate pose estimation and fast map update can be performed simultaneously. Moreover, we introduce a novel sliding window estimator which attains better accuracy by consistently utilizing sensor and map information with both scan-to-scan and scan-to-map data association. Real-world experimental results demonstrate that the proposed algorithm can be used for commercial robots in various environments with long-term usage. Experiments also show that our approach outperforms competing approaches by a wide margin.**

I. Introduction

Accurate global pose estimation is necessary for mobile robots to navigate their surroundings effectively. In robotics community, this *localization* problem is widely studied in recent 20 years, especially for the robots that are equipped with range sensors like lidar or sonar [1][2][3][4][5]. Most of the existing algorithms can be categorized into two groups: localization and mapping in an *unknown* environment[2][3] or in previously visited areas with pre-built localization maps [4][5]. This paper contributes to the latter one, enabling mobile robots to perform autonomous navigation repetitively under different conditions without human intervention. This overcomes one critical challenge for long-term autonomy of mobile robots and will make their low-cost commercial usage possible.

To achieve this goal, a properly designed algorithm should satisfy all following requirements. First of all, pose estimation with respect to an existing localization map should be of high precision and also be robust under different working conditions, e.g., surrounded with moving objects, different reflection properties of building materials, and so on. Secondly, although the operation area is previously visited and explored, in real-world applications it is inevitable that the environment changes over time, e.g., furniture re-arrangement in indoor spaces. Therefore, the capability of online map verification and update determines the performance of mobile robots over time. Finally, for wide real-world application and commercialization, both high-precision pose estimation and map update components should be able to operate in real time on low-cost platforms.

The authors are with Alibaba DAMO Academy AI Labs, Hangzhou, China. {ximing.zmm, yimingchen, na.lmy}@alibaba-inc.com.



(a) prior occupancy grid map (b) raw laser scan measurements

(c) localization results on a prior map (d) dynamic update for the prior map

Fig. 1: Representative building blocks of our proposed algorithm. With a prior occupancy grid map (a), and raw laser scan measurements (b), our approach estimates robot pose with respect to the global map (c) and perform dynamic map update (d). Note that, the occupancy grid map here is only used for visualization, and the pose estimation depends on the proposed SDF map.

The existing algorithms have certain shortcomings. First, existing map-based algorithms either require heavy computation as the ones that employ Monte Carlo approaches [4][5], or need well-tuned initial pose estimates to guarantee estimator convergence[3]. Moreover, we notice that most algorithms do not explicitly model the over-time changes of the operation environment (neither the corresponding localization map). Instead, they simply probabilistically reject the sensor measurements corresponding to the changed regions[6]. For small map changes this leads to reduced estimation accuracy and for large changes this can result in localization failures.

To address these limitations, in this paper, we present SDF-Loc, a concurrent localization and map update algorithm, based on hybrid signed distance field. The hybrid SDF scheme integrates ESDF and TSDF, which have complementary characteristics. For laser based localization problem, we show that ESDF leads to better estimation accuracy and convergence properties (see Section IV for details) but computationally expensive to be updated online to adapt to the environment changes. On the other hand, TSDF can be

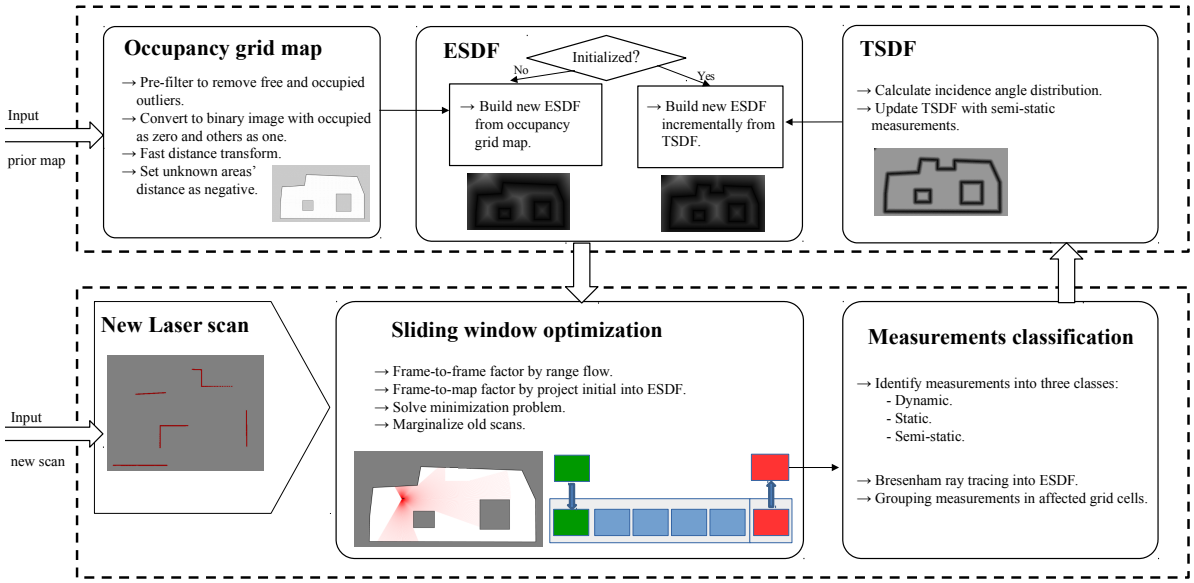


Fig. 2: The system overview of SDF-Loc

updated efficiently, but less robust in numerical optimization. This motivates us to combine them together. The main characteristics and contribution of the paper can be characterized as follows:

- A novel laser map representation by ESDF as well as the corresponding scan-to-map optimization function. The ESDF map can either be built incrementally or be converted from other map format, e.g., the widely used occupancy grid map [7].
- An nonlinear optimization based sliding-window estimator, which composes of sensor and map information in both scan-to-scan and scan-to-map manners to improve estimation accuracy and convergence properties.
- We also propose an efficient online map update method, which utilizes TSDF framework to merge new semi-static measurement points and subsequently build ESDF incrementally from updated TSDF.
- Finally, we provide real-world experimental results to demonstrate the accuracy and effectiveness of the proposed algorithm, as well as the comparison to competing approaches.

Fig. 1 shows the representative building blocks of our algorithm.

II. Related Work

In this section, we will give a brief review on the related work. The existing work on laser based localization can be categorized by the methods of processing laser scans and parameterizing laser maps. Generally, there are three types of approaches: sparse feature parameterization based ones [8][9][10], dense representation based ones [11][3][2][12], and recently introduced end-to-end deep learning based ones[13][14]. The first type of algorithms seeks to detect and geometrically parameterize repeatable features from laser scans, including but not limited to straight lines[8], corner points[10], and blobs[9]. Given detected sparse features, a data association step will be performed across frames, relying on either features' geometric information[8] or their corresponding descriptors[9]. Once that is done, robot poses can be estimated either jointly with those features or conditioned on offline computed ones. On the other hand, dense algorithms process entire laser scan without pre-selecting information. Occupancy grid mapping[11] is one of the most common dense approaches used nowadays, which presents 2D map via grid cells and computes occupancy probability for each cell. Various localization algorithms have been developed under this framework, and majority of them use either Particle filter estimation[11][3] or nonlinear iterative optimization[2]. Finally, in recently years, a couple of deep neural network based algorithms have also been proposed for tacking problems like laser odometry[13] or scan matching[14].

Recently, among above mentioned algorithms, dense based ones are more preferred in commercial applications. This is due to the fact that this type of algorithms use entire laser scan information which provides better accuracy and robustness guarantees. However, high-precision real-time pose estimation in occupancy grid framework[11] is problematic. Particle filter based approaches[11][3] can attain good accuracy, but its computational cost increases linearly as the number of particle increases. On the other hand, nonlinear optimization based approaches can be employed[3][2], but due to the sparsity of occupancy grid map representation, the corresponding estimators can only converge when good initial pose estimates are provided. This might not be always possible for long-term robotic operations. Borrowing the concept of TSDF from computer vision problems, Fossel [12] proposed a TSDF based localization algorithm, attaining better accuracy compared to traditional occupancy grids. Recent work also shows that TSDF can be directly used for

mapping, navigation, and path planning [15] [16]. However, TSDF is highly compressed map representation, by only keeping meaningful values near structural surfaces. As a result, when initial pose estimates are of large errors, nonlinear iterative minimization via TSDF still has large possibility to fall into local minimum solutions. To this end, we introduce ESDF[17][18] in this paper for improving the localization performance.

Another important topic of robot localization is to deal with dynamic environments, to detect and reject moving objects and to merge newly appeared static object into maps. The first problem is typically solved via frame-to-frame dense data association, by minimizing geometric consistency cost functions with respect to rigid motion assumption[19] [20]. Alternatively, moving objects can be detected and classified using convolutional neural networks[21]. On the other hand, to update localization maps, [22] utilizes temporary local maps to keep track of the observations caused by unexpected objects and uses both local maps and prior map for localization. [23] introduced a method named dynamic pose graph SLAM by introducing time as a parameter in the optimization problem. Specifically, when a robot operates in the same area multiple times, the pose graph will choose keyframes dynamically from each run for computing a map that best represents the current environment. Moreover, Krajník [24] [25] proposed a spatial-temporal occupancy grid approach in which each grid stores information about both occupancy persistency and periodicity. Based on that, these approaches can predict the probability of grid cells at different timestamps. However, occupancy persistency and periodicity are different in different environmental conditions, and those algorithms are not generic enough for off-the-shelf commercial usage.

III. Proposed approach

As mentioned previously, the focus of this paper is to perform concurrent localization and map update in previsited areas. Therefore, throughout this paper we assume a pre-built localization map is available. We do not place any limitation on the mapping algorithms that generate localization maps, as different state-of-the-art algorithm can be applied, e.g. [2] [23]. This is due to the fact that we do *not* require to have a perfect prior map since we can perform probabilistically map update to refine map details over time.

The overview of our system is shown in Fig. 2. Since most mapping algorithms generate occupancy grid maps, to make our algorithm convenient for usage, we first describe a pipeline for converting occupancy grid maps into ESDF. Next, we describe our sliding-window estimator by firstly introducing the basic formulation of our nonlinear optimization cost function, including both scan-to-scan term and scan-to-map term. Subsequently we will present both terms with mathematical details. Finally, we describe our proposed online map verification and update method.

A. Map Representation

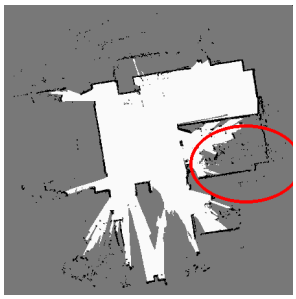
The main problem of using occupancy grid map for nonlinear optimization is its sparsity in the corresponding gradient map. Specifically, nonlinear optimization approaches require computing gradients with respect to the map on different positional axis. But due to the occupancy grid sparsity, the majority of those elements will be identically zero. As a result, when initial pose estimates are not highly accurate, which is common in environments with large structural changes or many moving objects, global optimal convergence cannot be reached. To solve this problem, we propose to use ESDF. ESDF is also a grid based map, in which every grid contains its euclidean distance to the nearest obstacle. We note that TSDF is another commonly used SDF based representation, which calculates projective distance to obstacles within *short radius* for each cell. It is clear that, compared with occupancy grid and TSDF, ESDF does not suffer from the 'vanishing gradient problem' due to its denser map representation.

Algorithm 1 Map pre-processing: Convert occupancy grid map into ESDF map.

GIVEN occupancy grid map P ▷ prebuilt
for every grid cell c in P **do**
  if $P[c]$ == **Occupied** **then**
    Count the freespace ratio in its surroundings: $r(c)$
    if $r(c) < r_{threshold}$ **then**
      $P[c] \leftarrow$ **Unknown**
    end if
  end if
end for
Find connected components for free space elements.
Set cells with small number of connected components as **Unknown**.
Perform fast Euclidean distance transformation.
Set **Unknown** cells' distances to be negative.
OUTPUT ESDF ▷ further usage

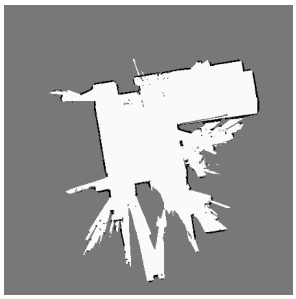
When an occupancy grid map is provided, the first step of our proposed method is to convert it into ESDF representation. We experimentally find that, many off-the-shelf mapping algorithms do not perform well with respect to some commonly seen materials, e.g., glasses and mirrors. Therefore, before performing the occupancy grid map to ESDF map conversion, we first apply a pre-processing filter to remove ill-conditioned map regions. Note that, this step is not necessary for *all* mapping algorithms, but it will help seamlessly connect the proposed algorithm to most widely used open-source implementations.

Specifically, we perform pre-processing based on the following criteria: all free-space cells should form a connected component and unknown-space cells should be separated from free-space cells by occupied-space cells. To present the filter details, we introduce the following notation **Occupied**, **Free**, and **Unknown**, to represent the corresponding cells. For an **Occupied** cell, we compute the percentage of **Free**



(a) Occupancy grid map before pre-processing.

(b) Euclidean distance map without pre-processing.



(c) Pre-processed occupancy grid map.

(d) Pre-processed Euclidean distance map.

Fig. 3: Representative map conversions case from occupancy grid map to ESDF maps. (a) Occupancy grid map before pre-processing step. The region corresponding to mirrors is shown with a red circle. (b) ESDF map without pre-processing. (c) Pre-processed occupancy grid map. (d) ESDF map converted from pre-processed occupancy grid map.

cells in its surroundings. If that is smaller than a given threshold, this **Occupied** cell cannot be trusted and we set it to **Unknown**. On the other hand, for each **Free** cell, we apply connected-components analysis to reject unsatisfied **Free** cells, and set those as **Unknown**. Once this pre-processing is finished, we propose to perform fast occupancy grid map to ESDF map conversion, following the idea of [26]. The complete process is described in Algorithm 1, and a representative map conversion case can be seen in Fig. 3.

B. Estimator Formulation

In laser based localization, we seek to estimate 2D laser poses with respect to the map coordinate frame over time. Mathematically, at timestamp k, the 2D pose $\mathbf{p}_k$ is:

$$\mathbf{p}_k = \begin{bmatrix} x_k, y_k, \theta_k \end{bmatrix}^T \tag{1}$$

where x_k and y_k represents the positional elements expressed in the map frame, and θ_k is the heading angle. By denoting the laser measurements at timestamp k as $\mathbf{s}_k$, and the prior map as $\mathbf{M}$, a probabilistic algorithm for computing $\mathbf{p}_k$ can be formulated as

$$\mathbf{p}_k^\star = arg\min_{\mathbf{p}_k} \mathcal{L}\left(\mathbf{p}_k; \mathbf{s}_k, \mathbf{M}\right) \tag{2}$$

where $\mathcal{L}(\cdot)$ is for computing scan to map difference as a function of pose $\mathbf{p}_k$, conditioned on the map $\mathbf{M}$ and scan $\mathbf{s}_k$. [3] is a representative algorithm of doing this. However, this equation solves the highly nonlinear optimization problem only relying on information of a single laser scan. As a result, the volume of the used information is not enough, and it can inevitably lead to local minimum solutions (See experimental section for validation details).

To improve the estimation accuracy, we propose to use a sliding window based algorithm, whose state at timestamp k is:

$$\mathbf{x}_k = \begin{bmatrix} \mathbf{p}_k^T, \mathbf{p}_{k-1}^T, \cdots, \mathbf{p}_{k-N+1}^T \end{bmatrix}^T \tag{3}$$

where N is the sliding window size. We formulate our estimation algorithm as:

$$\mathbf{x}_k^\star = arg\min_{\mathbf{x}_k} \Bigg(\sum_{i=k-N+1,\cdots,k-1} \mathcal{I}\left(\mathbf{p}_i, \mathbf{p}_{i+1}; \mathbf{s}_i, \mathbf{s}_{i+1}\right) + \sum_{i=k-N+1,\cdots,k} \mathcal{L}\left(\mathbf{p}_i; \mathbf{s}_i, \mathbf{M}\right) \Bigg) \tag{4}$$

where $\mathcal{I}(\cdot)$ is a function representing difference between two scans based on their corresponding poses. By introducing both scan to scan constraints as well as the sliding-window optimizer, the localization problem can be solved with improved performance.

Next, we will present laser frame-to-frame cost function $\mathcal{I}(\cdot)$ and frame-to-map cost functions $\mathcal{L}(\cdot)$ in details.

C. Cost functions

We first present frame-to-map cost function. Since each cell of ESDF map represents the distance to the closest surface, we directly seek to minimize the sum of distance values for all laser measurement points. Thus, we formulate the cost function as:

$$\mathcal{L}\left(\mathbf{p}_i; \mathbf{s}_i, \mathbf{M}\right) = \sum_{j=1,\cdots,K} \mathcal{F}\big(\mathcal{D}\left(\mathbf{m}_k^i\right)\big) \tag{5}$$

where K is the number of scan points, $\mathbf{m}_k^i$ is the range measurement expressed in map coordinate frame,

$$\mathbf{m}_k^i = \begin{bmatrix} \cos\theta_k & -\sin\theta_k \\ \sin\theta_k & \cos\theta_k \end{bmatrix} \mathbf{s}_k^i + \begin{bmatrix} x_k \\ y_k \end{bmatrix} \tag{6}$$

and $\mathcal{F}(\cdot)$ is the Cauchy M-estimator

$$F(a) = \frac{\tau^2}{2} ln\left(1 + \left(\frac{a}{\tau}\right)^2\right) \tag{7}$$

where τ is the robust estimation parameter. Solving Eq. 5 requires computing the Euclidean distance value $D(\mathbf{m}_k^i)$ as well as its corresponding gradient, evaluated at the map coordinate (m_x^i, m_y^i). This can be approximated by using bilinear interpolation on four grid cells from the latest ESDF, which will be incrementally updated from TSDF (see Section III-E for details).

On the other hand, to introduce frame-to-frame cost function, we present the continuous time laser range flow

equation[19]:

$$\rho(\xi) = (s_x^i \sin\beta - s_y^i \cos\beta^i - R_\alpha^i k_\alpha) w_\theta + R_t^i + \left(\cos\beta^i + \frac{R_\alpha^i k_\alpha \sin\beta^i}{r}\right) v_x + \left(\sin\beta^i - \frac{R_\alpha^i k_\alpha \cos\beta^i}{r}\right) v_y = 0 \tag{8}$$

where $\xi = (v_x, v_y, w_\theta)$ represents both positional and rotational velocities, β^i is the angle of the laser point $\mathbf{s}^i$ in laser scan $\mathbf{s}$, k_α is a laser intrinsic parameter representing angular resolution of laser beams, and R_α^i and R_t^i are the derivatives of $\mathbf{s}^i$ with respect to range and time.

We note that, in the above equation, $\rho(\xi)$ is *linear* in ξ. Therefore, if we denote Δt the time interval between two consecutive scan, $\delta = (d_x, d_y, d_\theta)$ is the relative transformation between them, we can write:

$$\rho(\delta) \simeq \rho(\xi \Delta t) = \rho(\xi)\Delta t = 0 \tag{9}$$

The first approximation equation conforms the linear motion assumption. For robots moving on 2D plane within short time interval between consecutive laser scans (e.g., 75 ms), this assumption is of small errors. Thus, the frame-to-frame cost can be formulated by minimizing $\rho(\delta)$ with a robust cost function, for all laser points in consecutive frames.

D. Sliding window optimization

With frame-to-frame and frame-to-map cost functions being formulated, we can rewrite the estimation problem Eq. 4 as:

$$\mathbf{x}_k^\star = arg\min_{\mathbf{x}_k} \left(\sum_{i=k-N+1,\cdots,k-1} \sum_{j=0,\cdots,K} \mathcal{F}\big(\rho(\delta)\big) + \sum_{i=k-N+1,\cdots,k} \sum_{j=0,\cdots,K} \mathcal{F}\big(\mathcal{D}\left(\mathbf{m}_k^i\right)\big) \right) \tag{10}$$

We solve this non-linear minimization problem by applying Levenberg Marquardt method, which iteratively linearizes the above cost function at each iteration. When a new pose is obtained, we propose to marginalize the oldest pose from the sliding window, and the oldest pose will be used for map update.

It must be noted that a typical consistent sliding window estimator must introduce a prior marginalization factor to minimization problem, to avoid information loss, either in information form [27] or covariance form [28] [29]. However, in our system, due to the existence of the global map, we can simply remove the oldest frame since it is conditional independent to other frames given the offline map.

E. Dynamic map update

Before we update the map, we must point out that there are different styles of dynamic objects in an environment. Similar to [6], we categorize the objects into three classes, based on their dynamics:

- **Static objects:** Objects that will not change their poses in the environments, like walls, fixed desks, etc. They can be used as strong cues for localization.
- **Semi-static objects:** Objects that change their poses with a relative low frequency, e.g., re-arranged furnitures. In this paper, we further assume these objects will not change their positions when they are in laser's FOV, but they can move afterwards. Those objects should be dynamically updated in localization maps.
- **Dynamic objects:** Objects that will change their positions when they are observed, e.g., moving people, moving trolley and so on. They are neither used for localization nor map update. They should be classified as outliers.

Algorithm 2 Map update algorithm

```
GIVEN marginalized laser scan s
for every measurement s_i in s do
    if Residue[s_i] > R_thre then
        Set it as semi-static measurement.
        Bresenham ray tracing into the ESDF.
        Push back affected grid cells into vector V_c.
        Group measurements for affected grid cells.
    end if
end for
Laser scan number N_scan ← N_scan + 1
if N_scan > N_thre then
    N_scan ← 0
    for every grid cell c_i in V_c do
        Calculate incidence angles' range A_r
        if A_r > A_thre then
            Perform TSDF update
        end if
    end for
    Build ESDF from TSDF incrementally
    OUTPUT ESDF                          ▷ further usage
    OUTPUT Occupancy grid map            ▷ further usage
end if
```

Static objects and semi-static objects can be identified by calculating the Euclidean distance residual values on the corresponding cells. If the distance is larger than preset threshold R_{thre}, the corresponding measurement will be labeled as semi-static measurement, otherwise as static measurement. Dynamic objects can be detected and rejected simply by the range flow calculation. Only semi-static measurements will be used in map update.

Bresenhams ray tracing algorithm[30] is used to project these semi-static measurements into grid cells, and the potential affected grid cells are pushed into a vector. To further speed up the update process, we use a method similar to [31] and [16]. For each point in a laser scan, we project its position to the corresponding grid cell, and group it with all other points that are mapped to the same grid cell.

Different from depth sensors, laser scanners usually have very large FOV, e.g., more than 180 degrees, leading to

large projective errors for SDF. Here we assume local planar objects, and the error can be expressed as:

$$r_p(\gamma) = d - dsin(\gamma) \tag{11}$$

where d is the measured distance, and γ is the angle between laser ray and the surface observed in the laser scanner. In fact, γ can vary from 0 to $\pi/2$, leading to maximum error: $r_{max} = d$.

To cope with this problem, we first project semi-static measurements in every laser scan to the grid cells, but only update TSDF when the laser scan number exceed N_{thre}. For certain grid cell, the TSDF values only need to be updated when it has laser scan measurements from many different incidence angles to alleviate the error caused by small incidence angle. Once the TSDF are updated, 2D modification of a fast method similarly to [16] is applied to update the ESDF incrementally. More details on map update procedure can be found in Algorithm 2.

IV. Implementation and Experiments

A. Implementation and Setup

We used C++ to implement the source code, in which Ceres solver [32] was used for our proposed nonlinear optimization framework. The core source code is ROS-free, but ROS interface was also implemented in our experiment for visualizing the occupancy grid maps.

In order to evaluate our proposed approach, we conducted experiments in four different indoor places. [2] is used for offline mapping, and the map grid size is 5cm. As mentioned previously, it is important to note the proposed algorithm does *not* place any limitation on the choice of the mapping algorithm and we used [2] due to its open-source implementation. The computed occupancy grid map of the four environments are showed in Fig 4, and example images are also provided in Fig 5.

B. Results

1) Convergence and Accuracy: We first compare the proposed approach against [7] (termed AMCL), which is a Monte Carlo localization approach [7] with KLD-sampling [33]. The Monte Carlo based algorithms are widely used in robotics and autonomous driving applications. The online open source implementation [34] is used in our experiments. For quantitative evaluation, we use the metrics described in [35] to compare the pose errors (transitional and rotational elements).

The first experiment is to evaluate the capability of optimization convergence. Specifically, we tested whether different methods can converge (generate pose estimates with small errors), when the input pose estimates are of large errors. To do this, we manually added positional and rotational errors to initial pose estimate for the proposed method and [34], and tested the maximum allowable errors for both. To decouple the effect of transition and rotation, we separated them in input pose estimates. For example, for evaluating the transitional components, we only added errors to themselves and left rotational elements unchanged.

TABLE I: Convergence evaluation: Maximum allowable added errors for both methods that can still operate (yield acceptable localization performance).

	SDF-Loc	[34]
OFFICE		
Max. allowable initial position err. (m)	2.72	1.31
Max. allowable initial rotation err. (deg.)	30.2	10.3
RESTAURANT		
Max. allowable initial position err. (m)	1.69	0.94
Max. allowable initial rotation err. (deg.)	20.7	10.9
HOTEL		
Max. allowable initial position err. (m)	2.13	1.02
Max. allowable initial rotation err. (deg.)	38.1	21.5
HALL		
Max. allowable initial position err. (m)	1.52	1.06
Max. allowable initial rotation err. (deg.)	23.8	11.7

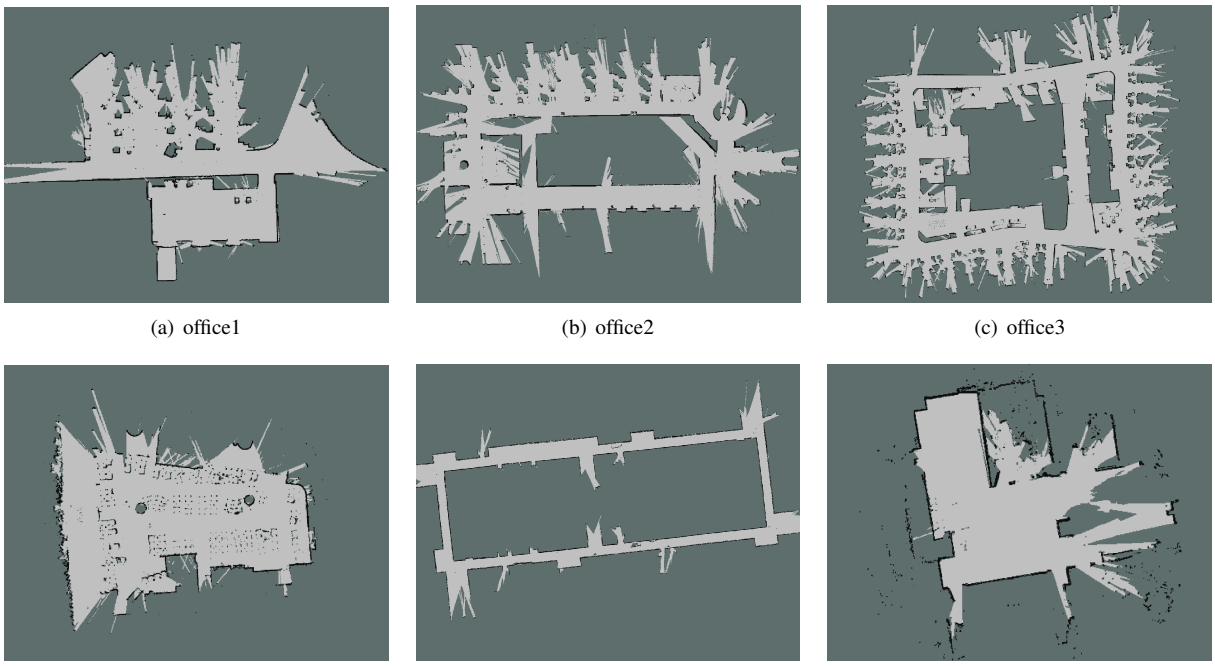
Also, since we will show the necessity and performance of the sliding window optimiser later, in this test, we constantly set the sliding window size to be 1, to focus on the map representation and ESDF scan-to-map cost function. Table I shows that the proposed algorithm outperforms competing method by a wide margin. In all four tested environments, the proposed algorithm can all cope with large initial errors. This indicates that, the proposed map formulation and cost function is better suited for the purpose of localization.

We also compared the general performance of the proposed algorithm and AMCL, when both algorithms are provided with identical reasonable initial pose estimates. We set the maximum particle number to be 400, which can vary over time. Additionally, we set the sliding window size to be 7, and the detailed analysis of sliding window size can be found in the next section.

The experimental results can be seen in the Table II. Our method clearly performs better in terms of both localization accuracy and runtime. We note that, this is achieved by both the hybrid SDF map representation as well as sliding window estimator formulation. On one hand, the hybrid SDF map representation makes itself more suitable for nonlinear optimization, leading to better convergence. On the other hand, the sliding window estimator consistently combines frame-to-frame and frame-to-map constraints, results in better tightly coupled information fusion.

2) Sliding Window Formulation: We also conducted an experiment to evaluate the performance of the proposed algorithm with different sliding window sizes. It is important to note that, when the sliding window size equals 1, the frame-to-frame constraint won't be effectively used. As a result, the estimator formulation becomes similarly to [3], with different map representation though.

The experimental results of accuracy and runtime are shown in Fig. 6. We first note that, when the window size is 1, the largest error is obtained. This indicates the necessity of having more frames for joint optimization as well as the need of frame-to-frame constraint. Also, due to limited space in the paper, we did not present experimental results against [3], while our tests also indirectly indicate that our proposed formulation can better deal with the problem of laser based



(a) office1 (b) office2 (c) office3

(d) restaurant (e) hotel (f) hall

Fig. 4: Occupancy grid map in three office localizations, a local restaurant, a hotel, and a hall. Black color represents **Occupied** space, white color represents **Free** space, and gray color shows **Unknown** space.

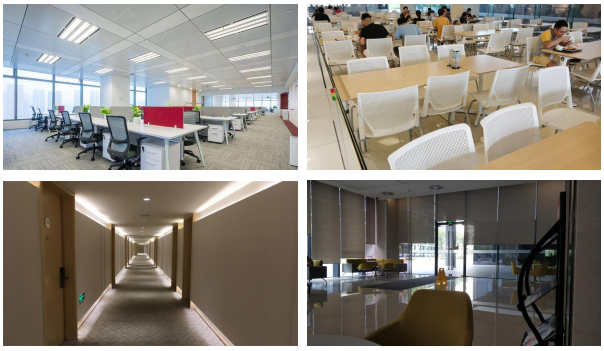


Fig. 5: Four typical images for indoor scenarios, *office* (top left), *restaurant* (top right), *hotel* (bottom left), *hall* (bottom right)

TABLE II: Localization error and runtime

	Proposed Algorithm	AMCL[34]
OFFICE		
Absolute transitional (m)	0.040379	0.051379
Absolute rotational (degree)	0.396035	0.843618
runtime (ms)	10.62	27.58
RESTAURANT		
Absolute transitional (m)	0.049022	0.051961
Absolute rotational (degree)	0.507419	0.63468
runtime (ms)	9.20	17.73
HOTEL		
Absolute transitional (m)	0.046132	0.057461
Absolute rotational (degree)	0.407419	0.507419
runtime (ms)	9.93	20.57
HALL		
Absolute transitional (m)	0.047557	0.064310
Absolute rotational (degree)	0.524398	0.671429
runtime (ms)	10.51	22.72

localization. Moreover, we note that, when the number of sliding window poses increases, the accuracy will increase, but there exists a 'cut-off' number: when 7 poses are used for optimization, introducing more poses leads to less accuracy gain.

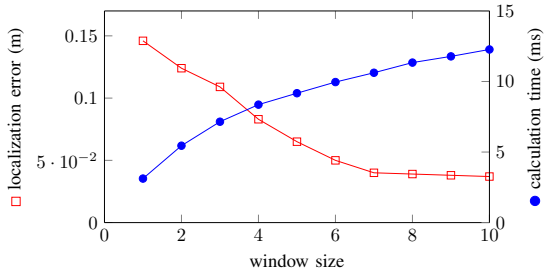


Fig. 6: Localization error and runtime for different sliding window size ($N = 1 \sim 10$) in office scenarios.

3) Map update: Although it is difficult to directly evaluate map update performance, we are still able to compare the localization accuracy with and without map updates. This will show the importance of conducting the map update process. To do that, we conducted two experiments: 1) indoor localization with re-arranged furnitures, and 2) office localization with moved large planar objects.

Results are shown in Table III. We can clearly find that, when map update is performed, it better characterizes the current environmental conditions and leads to better localization accuracy. This also indicates that, to have real-world robotic system deployment, this module is absolutely

TABLE III: Localization accuracy with and without map update

	with original map	with updated map
Re-arranged furniture test		
Position error (m)	0.052473	0.042187
Rotation error (deg.)	0.63121	0.53901
Moved large planar object test		
Position error (m)	0.055269	0.041254
Rotation error (deg.)	0.66398	0.50032

necessary, since the dynamic scene changes are inevitable.

REFERENCES

[1] F. Dellaert, D. Fox, W. Burgard, and S. Thrun. Monte carlo localization for mobile robots. In *IEEE International Conference on Robotics and Automation*, volume 2, pages 1322–1328 vol.2, 1999.

[2] Wolfgang Hess, Damon Kohler, Holger Rapp, and Daniel Andor. Real-time loop closure in 2d lidar SLAM. In *IEEE International Conference on Robotics and Automation*, pages 1271–1278, 2016.

[3] Stefan Kohlbrecher, Oskar Von Stryk, Johannes Meyer, and Uwe Klingauf. A flexible and scalable SLAM system with full 3d motion estimation. In *IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR)*, pages 155–160. IEEE, 2011.

[4] Philipp Ruchti, Bastian Steder, Michael Ruhnke, and Wolfram Burgard. Localization on Openstreet map data using a 3d laser scanner. In *IEEE International Conference on Robotics and Automation*, pages 5260–5265, 2015.

[5] Alexander Schiotka, Benjamin Suger, and Wolfram Burgard. Robot localization with sparse scan-based maps. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 642–647, 2017.

[6] Daniel Meyer-Delius, Jürgen Hess, Giorgio Grisetti, and Wolfram Burgard. Temporary maps for robust localization in semi-static environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5750–5755, 2010.

[7] Dieter Fox, Wolfram Burgard, Frank Dellaert, and Sebastian Thrun. Monte carlo localization: Efficient position estimation for mobile robots. *AAAI/IAAI*, 1999(343-349):2–2, 1999.

[8] Joel A. Hesch, Faraz M. Mirzaei, Gian Luca Mariottini, and Stergios I. Roumeliotis. A laser-aided inertial navigation system (l-ins) for human localization in unknown indoor environments. pages 5376–5382, 2010.

[9] F. Kallasi and D. L. Rizzini. Efficient loop closure based on FALKO lidar features for online robot localization and mapping. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1206–1213, Oct 2016.

[10] Y. Li and E. B. Olson. Extracting general-purpose features from LIDAR data. In *IEEE International Conference on Robotics and Automation*, pages 1388–1393, May 2010.

[11] Adam Milstein. Occupancy grid maps for localization and mapping. In *Motion Planning*. InTech, 2008.

[12] Joscha-David Fossel, Karl Tuyls, and Jürgen Sturm. 2d-sdf-slam: A signed distance function based slam frontend for laser scanners. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1949–1955. IEEE, 2015.

[13] Austin Nicolai, Ryan Skeele, Christopher Eriksen, and Geoffrey A Hollinger. Deep learning for laser based odometry estimation. In *RSS workshop Limits and Potentials of Deep Learning in Robotics*, 2016.

[14] J. Li, H. Zhan, B. M. Chen, I. Reid, and G. H. Lee. Deep learning for 2d scan matching and loop closure. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 763–768, Sept 2017.

[15] Helen Oleynikova, Alexander Millane, Zachary Taylor, Enric Galceran, Juan Nieto, and Roland Siegwart. Signed distance fields: A natural representation for both mapping and planning. In *RSS 2016 Workshop: Geometry and Beyond-Representations, Physics, and Scene Understanding for Robotics*. University of Michigan, 2016.

[16] Helen Oleynikova, Zachary Taylor, Marius Fehr, Roland Siegwart, and Juan Nieto. Voxblox: Incremental 3d Euclidean signed distance fields for on-board MAV planning. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[17] Matt Zucker, Nathan Ratliff, Anca D Dragan, Mihail Pivtoraiko, Matthew Klingensmith, Christopher M Dellin, J Andrew Bagnell, and Siddhartha S Srinivasa. Chomp: Covariant hamiltonian optimization for motion planning. *The International Journal of Robotics Research*, 32(9-10):1164–1193, 2013.

[18] Boris Lau, Christoph Sprunk, and Wolfram Burgard. Improved updating of euclidean distance maps and voronoi diagrams. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 281–286, 2010.

[19] Mariano Jaimez, Javier G Monroy, and Javier Gonzalez-Jimenez. Planar odometry from a radial laser scanner. a range flow-based approach. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 4479–4485, 2016.

[20] Mariano Jaimez, Javier Monroy, Manuel Lopez-Antequera, and Javier Gonzalez-Jimenez. Robust planar odometry based on symmetric range flow and multi-scan alignment. *IEEE Transactions on Robotics*, 2018.

[21] Aseem Behl, Despoina Paschalidou, Simon Donné, and Andreas Geiger. Pointflownet: Learning representations for 3d scene flow estimation from point clouds. *arXiv preprint arXiv:1806.02170*, 2018.

[22] Peter Biber, Tom Duckett, et al. Dynamic maps for long-term operation of mobile service robots. In *Robotics: science and systems*, pages 17–24, 2005.

[23] Aisha Walcott-Bryant, Michael Kaess, Hordur Johannsson, and John J Leonard. Dynamic pose graph SLAM: Long-term mapping in low dynamic environments. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1871–1878, 2012.

[24] Tomáš Krajník, Jaime Pulido Fentanes, Marc Hanheide, and Tom Duckett. Persistent localization and life-long mapping in changing environments using the frequency map enhancement. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4558–4563, 2016.

[25] Tomáš Krajník, Jaime P Fentanes, Joao M Santos, and Tom Duckett. Fremen: Frequency map enhancement for long-term mobile robot autonomy in changing environments. *IEEE Transactions on Robotics*, 33(4):964–977, 2017.

[26] Pedro F Felzenszwalb and Daniel P Huttenlocher. Distance transforms of sampled functions. *Theory of computing*, 8(1):415–428, 2012.

[27] T. Dong-Si and A. I. Mourikis. Motion tracking with fixed-lag smoothing: Algorithm and consistency analysis. In *IEEE International Conference on Robotics and Automation*, pages 5655–5662, May 2011.

[28] Mingyang Li and Anastasios I. Mourikis. High-precision, consistent Ekf-based visual-inertial odometry. *The International Journal of Robotics Research*, 32(6):690–711, 2013.

[29] Mingyang Li, Hongsheng Yu, Xing Zheng, and Anastasios I Mourikis. High-fidelity sensor modeling and self-calibration in vision-aided inertial navigation. In *IEEE International conference on Robotics and Automation*, pages 409–416, May 2014.

[30] Jack E Bresenham. Algorithm for computer control of a digital plotter. *IBM Systems journal*, 4(1):25–30, 1965.

[31] Matthew Klingensmith, Ivan Dryanovski, Siddhartha Srinivasa, and Jizhong Xiao. Chisel: Real time large scale 3d reconstruction onboard a mobile device using spatially hashed signed distance fields. In *Robotics: science and systems*, volume 4, 2015.

[32] Sameer Agarwal, Keir Mierle, and Others. Ceres solver. http://ceres-solver.org.

[33] Dieter Fox. Kld-sampling: Adaptive particle filters. In *Advances in neural information processing systems*, pages 713–720, 2002.

[34] Ros implementation of amcl. http://wiki.ros.org/amcl.

[35] Rainer Kümmerle, Bastian Steder, Christian Dornhege, Michael Ruhnke, Giorgio Grisetti, Cyrill Stachniss, and Alexander Kleiner. On measuring the accuracy of slam algorithms. *Autonomous Robots*, 27(4):387, 2009.