Robotica (2022), **40**, pp. 3786–3810
doi:10.1017/S0263574722000613

RESEARCH ARTICLE

Stable keypoints selection for 2D LiDAR based place recognition with map data reduction

Lounis Douadi[1],*, Yohan Dupuis[2] and Pascal Vasseur[3]

[1]Université de Rouen Normandie, LITIS (laboratoire d'informatique, du traitement de l'information et des systèmes), Saint-Étienne-du-Rouvray 76800, France, [2]CESI (centre des études supérieures industrielles), LINEACT (laboratoire d'innovation numérique pour les entreprises et les apprentissages au service de la compétitivité des territoires), Paris La Défense 92074, France, and [3]Université de Picardie Jules Verne, Laboratoire MIS (modélisation, information, systèmes), Amiens 80080, France
*Corresponding author. E-mail: lounis.douadi@gmail.com

Received: 18 October 2021; **Revised:** 27 January 2022; **Accepted:** 24 March 2022; **First published online:** 25 April 2022

Keywords: place recognition, 2D LiDAR, keypoint selection, map data reduction, global localization, loop closure, SLAM

Abstract

This paper presents a new feature based approach for place recognition using 2D LiDAR (Light Detection And Ranging) data. The main contribution lies in the mapping process. It includes a keypoint selection strategy to model places with persistent keypoints from concatenated LiDAR scans. Our objective is to achieve map data reduction while maintaining good place recognition performace. LiDAR scans are concatenated with a registration algorithm and keypoints are extracted from each scan. Based on a regular grid, our approach measures the occurrence of similar keypoints in each region of interest defined by a grid cell. Only keypoints with occurrences beyond a threshold, qualified as stable keypoints, are kept in the place model called submap. The environment is therefore modeled as a collection of submaps, which constitutes the global map. Place recognition consists in submap matching followed by a two steps geometric verification. In the first stage, optimal parameters are set using corrected data. Mapping parameters satisfy six criteria among which is the spatial distribution distance, which represents another contribution of our work. It gives a measure on how well keypoints are distributed in space. Place recognition optimal parameters are set through global localization. In the second stage, the approach is evaluated using raw data in the contexts of global localization, and loop closure detection in a SLAM framework. The results obtained using popular real data sets show that our approach achieves significant map data reduction (up to 92%) while maintaining good place recognition performance, comparable to state-of-the-art methods.

1. Related work

The work presented in this article is in the area of place recognition, which is an important component of localization and mapping fields. Using perceptual data, place recognition is the process of modeling the current robot's environment called place, in order to detect it, in a later stage, as an already visited place. This information is crucial in applications such as loop closure in SLAM or global localization. As an active research topic, place recognition literature is constantly growing, especially for image and 3D LiDAR modalities. Because of richer data provided by 3D LiDARs and cameras, and since the former sensors are becoming more accessible nowadays, place recognition literature is more furnished with works using these sensors and references on place recognition with 2D LiDAR data remain insufficient. Nevertheless, 2D LiDAR data present the interest of being considerably lighter and 2D sensors are still widely spread as compared to their 3D counterparts.

There is no unique classification of place recognition approaches in the literature, but they can coarsely be grouped into two major categories. There are methods based on photometric data and methods based on geometric data. In the literature, methods of the first category are often qualified as appearance based, although this term is sometimes used for the second category as well. In addition, since there are works using both data types thanks to sensors such as depth cameras [1, 30], the limit between the two categories is not always clear.

Early works on modeling the appearance with photometric data use bag of visual words [2, 3]. One popular algorithm in this context is FAB-MAP for fast appearance based mapping, introduced by Cummins and Newman [4, 5], a probabilistic approach for place recognition in the context of SLAM. The authors model the appearance of a place with bags of words (BoW), based on a variant of the SURF descriptor [6]. In ref. [7] loop closure is achieved with image retrieval using BoW in a Bayesian scheme. The authors in ref. [8] obtain better performance than FAB-MAP thanks to feature-covisibility which exploits spatial information between features allowing better robustness to visual aliasing. seqSLAM introduced in ref. [9], models the appearance with image sequences. In a comparison to FAB-MAP the authors show the ability of seqSLAM to cope with changing environments. Due to sequence matching, the original algorithm is time consuming and an improved version is proposed in ref. [10].

Geometric data based approaches can be split into two groups, depending whether they use 2D or 3D data. Recently, 3D sensors became relatively affordable and less bulky. They offer a rich information on the environment and are, therefore, more attractive. Magnusson et al. [11, 12] exploit the NDT (Normal Distribution Transform) surface representation for 3D based place recognition for loop detection in SLAM. They use feature histograms based on surface orientation to describe a place. In ref. [13] rotation invariant 3D features are defined and combined through machine learning using AdaBoost. The authors perform real-time loop closure where no a priori knowledge of the robot's displacement is required. Muhammad and Lacroix [14] introduce small sized signatures for 3D cloud description applied to loop closure in outdoor environments. Local surface normals quantized in an histogram form these signatures. Place recognition is achieved by signature comparison using histogram distance. In ref. [15] fast point features histogram features are used in FAB-MAP instead of visual features for loop closure detection in difficult environments, such as austere tunnels. The authors found that their solution shows less performance in terms of recall as compared to FAB-MAP with visual data.

Because 2D LiDAR data are lighter than 3D LiDAR data they are suitable for applications such as large-scale mapping. However, this also means that 2D LiDAR data is less descriptive of the environment. Research works in this field aim at solving this dilemma by effectively exploiting this data type. One of the earliest works on 2D LiDAR based place recognition is that of Bosse and Zlot [16]. The authors developped a method for large-scale outdoor mapping that allows them to produce the largest real-world SLAM map at the time of their publication, with almost 30 km length. Their method uses range data provided by a pair of 2D LiDARs mounted back-to-back on a vehicle's roof. The method proceeds with two levels, local and global. In the local stage, laser scans are concatenated pairwise to form local maps of a certain size, called submaps. Submaps are described with histogram based signatures. During the global stage, each newly built submap is compared to previous submaps thanks to their signatures, in order to detect loop closure events. The classification based approach in ref. [19] shows better performance than Bosse's and Zlot [16] algorithm on smaller maps, and requires no manual parameter settings or heuristics. Tipaldi et al. [20] introduce a new scan signature called GFP for geometrical FLIRT [21] (fast laser interest region transform) phrases. Similarly to bag of words, a vocabulary of FLIRT words is learnt from a learning database. Every scan is translated into a clockwise-ordered sequence of FLIRT words. FLIRT is a 2D LiDAR operator introduced by the same authors in ref. [21], and which is used in our work for keypoint detection and description. In [23] Himstedt and Maehle propose a new scan signature called GSR (geometric surface relations) that outperforms GFP [20]. It is based on GLARE (geometrical landmark relations) operator [24] from the same authors. This signature makes use of geometric relations within cooccuring features in a scan. Feature ordering is also exploited in ref. [25], in addition to neighbor relational information relating neighboring scans in the SLAM graph. Here, place recognition is addressed as a BoW based document retrieval problem. In ref. [26] the authors introduce geometric relation distribution, a new signature for place recognition in landmark based maps. This signature encodes pairwise geometric relations (angles and distances) among landmarks of a local map into a probability density.

https://doi.org/10.1017/S0263574722000613 Published online by Cambridge University Press

We have briefly reviewed some popular works on place recognition using photometric or geometric data. Most of them rely on local features that are quantized into signatures. Some approaches model the place with an individual acquisition while others require more adjacent acquisitions resulting in a sequence of images or local maps, depending on data modality. In our case, local features from cumulated scans are combined to model a place.

The remainder of this paper is organized as follows. Section 2 provides a summary on our approach and presents the data sets used in our evaluations. The mapping and the place recognition processes are detailed in Sections 3 and 4, respectively. Parameter setting for each process is given in Section 5. In Section 6, we evaluate our approach and discuss the obtained results. Finally, our conclusions and perspectives are given in Section 7.

2. Introduction

We are interested in performing 2D LiDAR place recognition based on local features, namely 2D keypoints. Local feature based methods, such as in refs. [20–25], model a place with keypoints extracted from individual laser scans. With such a representation of the environment, map data keeps growing with the number of laser readings. Our objective is to achieve map data reduction while maintaining high place recognition performance. Instead of modeling each place with features from one scan, a place can be modeled with features extracted from multiple concatenated scans. This principle was used in refs. [16–18], where the authors model a place with keypoints extracted from adjacent scans registered to form local maps or submaps. We follow a similar reasoning, however, we proceed differently for keypoint selection and submap construction and matching. In our case, keypoints are detected from every new scan inserted in the submap using the state-of-the-art operator FLIRT [21]. As scans are concatenated, some regions of interest are more prone to produce similar detector responses than others. We want to identify regions with higher probability of producing similar detector responses, and select from each of these regions a representative keypoint. For every region of interest, we count the occurrences of similar keypoints from the concatenated scans. This occurrence count, normalized with the number of scans reaching the region of interest, constitutes the keypoint stability score in our keypoint selection strategy. Only keypoints selected from regions with a high keypoint stability score are kept in the place model. These keypoints, which constitute the submap, have more chances of being extracted in a later stage, with close descriptor values. Submap termination is based on two major criteria, among which is the spatial distribution of keypoints and which represents another contribution of our work. To the best of our knowledge, such stability score estimated online, for 2D LiDAR place recognition with map data reduction was not done before. The results obtained using popular real data sets show that our approach achieves significant map data reduction (up to 92%) while maintaining good place recognition performance, comparable to state-of-the-art methods.

The mapping process consists of two levels of hierarchy, local and global. As schematized in the dataflow of Fig. 1, the local mapping models places from cumulated scans to build local maps, also called submaps. The global map is the list of submaps from all mapped places. Depending on the application, the global map can be built online, for instance in the case of SLAM, or offline as it is the case for global localization.

Place recognition dataflow is depicted in Fig. 2. It consists in a submap matching process followed by a geometric verification step. For a submap describing the current location, place recognition looks for a matching submap in the global map. The global pose can therefore be infered from this information. In the subsection below, we give details on the data sets used in our experiments.

2.1. The data sets

For all experiments in this paper, we use real data from popular 2D LiDAR data sets. More specifically, three data sets corresponding to different indoor environments are used. The first two data sets were

https://doi.org/10.1017/S0263574722000613 Published online by Cambridge University Press

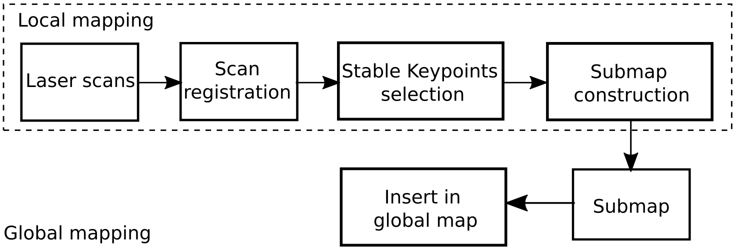


Figure 1. *Global and local mapping dataflow.*

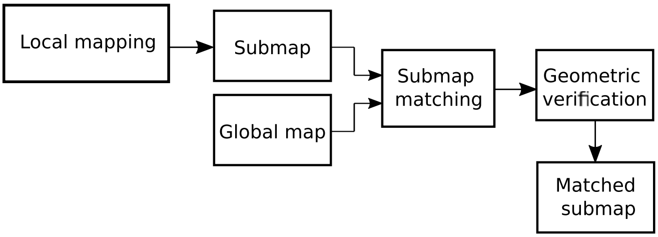


Figure 2. *Place recognition dataflow.*

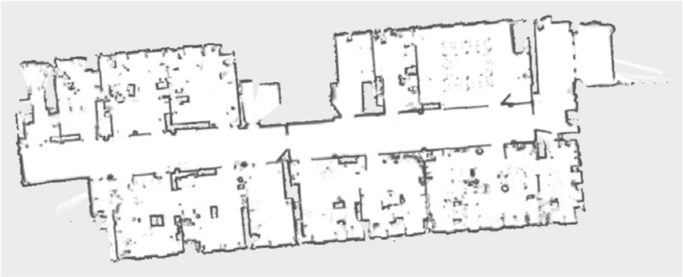


Figure 3. *Map using corrected data FR*079.

acquired, respectively, in Freiburg university buildings 79 and 101, while the third was acquired in Seattle's Intel Research Laboratory. For each data set, we need the raw data as well as the corrected data. Raw data contain all acquired laser scans together with odometry poses. Corrected data contain a minimal subset of laser scans together with corrected transforms provided by SLAM. We consider these corrected transforms as ground truth transforms, which we also qualify in this document as the provided transforms. Figures 3–5 show maps obtained from the corrected data for these data sets.

Corrected data sets are named *FR*079, *FR*101, and *Intel*, corresponding respectively to Freiburg building 79, Freiburg building 101, and Seattle's Intel Research Laboratory. *FR*079 and *Intel* are provided in ref. [35], and *FR*101 is available in ref. [31]. Similarly, raw data are named *FR*079*_r*, *FR*101*_r*, and *Intel_r*, all are available in Freiburg university's database [31]. The number of scans in each data set is given in Table I. Each environment has its characteristics. Building *FR*079 contains a long corridor which is subject to visual aliasing. Building *FR*101 is an open space, with less geometric features and can be assimilated to an outdoor environment. Intel Research Laboratory also presents a long corridor, but more importantly it features many symmetries. On the other hand, the raw version of this data set contains the largest number of scans. The evaluations in this paper were all run on a personal computer with *Intel i*3 − 3217*U* CPU at 1.80 GHz, and 8 GB of RAM. Our approach is detailed, in the next two sections, through its two major components that are the mapping and the place recognition processes.

https://doi.org/10.1017/S0263574722000613 Published online by Cambridge University Press

Table I. *Number of scans in corrected and raw data.*

Dataset	*FR*079	*FR*079_*r*	*FR*101	*FR*101_*r*	*Intel*	*Intel_r*
Number of scans	1464	4934	292	4758	2672	13,631

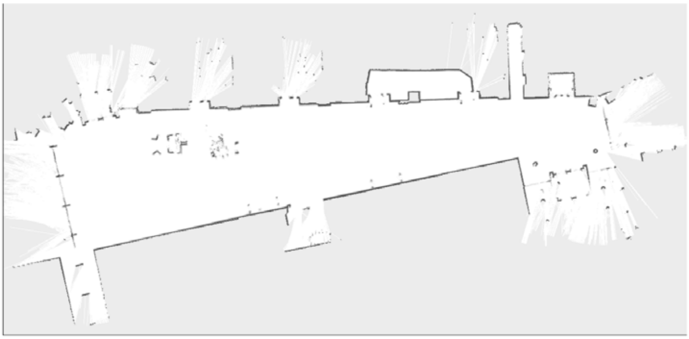


Figure 4. *Map using corrected data FR*101.

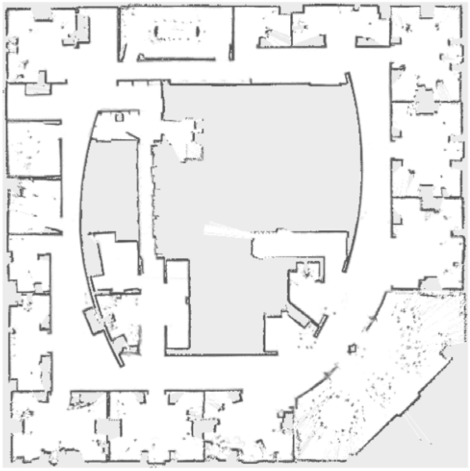


Figure 5. *Map using corrected data Intel.*

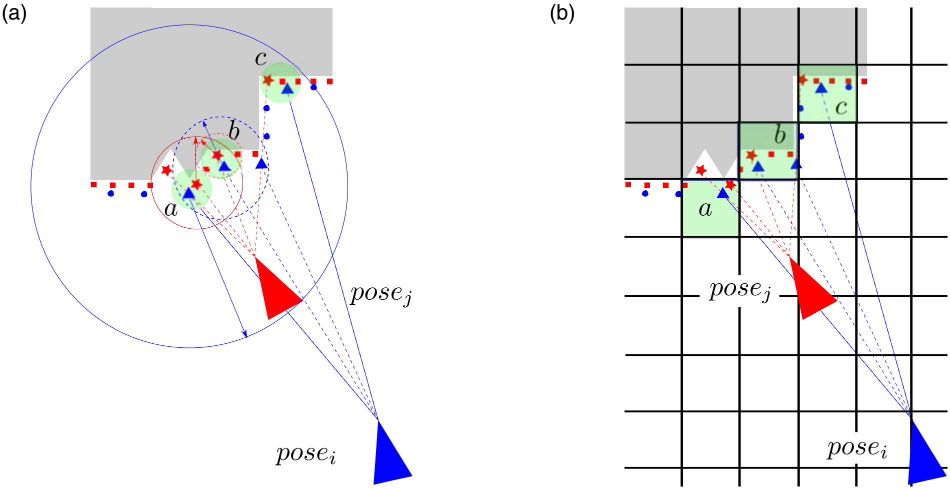
3. Mapping process description

The mapping process models the environment as a collection of places. We define a place as the immediate environment surrounding the robot. Therefore, a place corresponds to a local map, also called submap. Our mapping approach is hierarchical with two levels of hierarchy (see Fig. 1). The lowest level is the local mapping, which is in charge of submap construction. The highest level is the global mapping, which collects all submaps produced by the local mapping while exploring the environment, in a list representing the global map. In the following subsection we expose the local and global mapping algorithms.

3.1. Local mapping

Here, we are interested in modeling places with local maps of the environment using 2D LiDAR local features. Instead of modeling a place with features extracted from a single laser reading, our approach locally collects information from multiple scans in order to describe a place with persistent features only.

https://doi.org/10.1017/S0263574722000613 Published online by Cambridge University Press



2D features from two laser readings.
Regions of interest are represented by green discs

Grid based region of interest represented by green cells

***Figure 6.** Illustration of the mapping process. Two 2D LiDAR scans acquired from two positions i and j corresponding to blue and red colors.*

We use FLIRT [21] operator to extract 2D keypoints from individual scans. This operator combines a curvature-based detector with a beta-grid descriptor. It has shown good repeatability and precision performance in refs. [21, 22]. Its multiscale capacity allows to cope with varying laser points density. In addition, this operator is orientation invariant. More details on FLIRT operator are found in refs. [21, 22]. Figure 6(a) shows two scans corresponding to sensor poses i and j, represented by the blue and red dots, respectively. FLIRT features are symbolized with blue triangles and red stars, accordingly. For instance, the region of interest a contains two keypoints (one from each scan) and detected with different scales. They have different descriptor values (defined by support points spatial distributions in each neighborhood, delimited by the blue and red circles, respectively). Conversely, region b contains two keypoints with close descriptors at different scales. We qualify such keypoints as stable or persistent. With an appropriate keypoint selection we can obtain a compact and yet consistent representation of the environment. For instance, by keeping one keypoint instance from regions with persistent keypoints only (such as in region b). Stable keypoint selection strategy is, therefore, key component of our work. It is detailed in the next two subsections.

3.1.1. Keypoint stability score

In order to identify regions with high probability of producing similar detector responses, we need to cumulate several scans and count, for each region of interest, the number of keypoint occurrences having similar descriptor values. The submap is, then, built through online selection of stable keypoints among all keypoints from concatenated scans. Scans are registered with ICP [29] based on libPointMatcher open source library [32]. As stated before, we use FLIRT operator to extract and describe 2D keypoints from individual scans. The FLIRT implementation used in this work is provided by the open source library FLIRTLib [33].

A region of interest or ROI is a region of the environment containing at least one keypoint. In order to count keypoint occurrences in different ROIs we need to define this region in the 2D space. One convenient way to do this is through a regular grid G. Keypoints projected on the grid are grouped into individual cells. In Fig. 6(b), an example of such grid is shown. The grid size adapts with cumulated scan limits in a way that does not affect ROIs’ content. Details of how G is built and updated, as laser

https://doi.org/10.1017/S0263574722000613 Published online by Cambridge University Press

scans are concatenated, are given below.

$$G(i,j) = \{H, h, m, ns\} \tag{1}$$

$G(i,j)$ is a cell of G at row i and column j. For the sake of simplicity, we will mention row and column indexes for elements of $G(i,j)$ only when necessary. h and m are, respectively, the laser beam hit and miss counts in this cell, while ns counts the number of laser scans reaching the cell. It is incremented each time a scan produces a value of h greater than zero. H is a list that contains keypoints history for the cell $G(i,j)$. Its elements are keypoint sets having similar descriptor values, such as

$$H = \{H_0, H_1, \cdots, H_{n-1}\} \tag{2}$$

H_k with k in $[0, n-1]$ is a set of keypoints with similar descriptor values, called history cluster. Based on the work of Tipaldi et al. [22], descriptors are similar when the symmetric χ^2 distance between them is within the value 0.4. For every new keypoint falling in $G(i,j)$, we look for a match with keypoints in every history cluster of H. If a match is found, the keypoint is inserted in the corresponding cluster. Otherwise, a new history cluster is created with the new keypoint.

Keypoint basic stability score $s_b(i,j)$ for every ROI delimited by $G(i,j)$ is then easily defined as

$$s_b(i,j) = \frac{\text{occ}}{ns} \tag{3}$$

occ corresponds to the occurrence count in the largest history cluster in H, such as

$$\text{occ} = |H_q| \tag{4}$$

with q being the index of the largest history cluster in H. In practice, H is sorted by decreasing cluster size. In that case, $q = 0$. Also, this means that for a new keypoint projected in $G(i,j)$, the largest history clusters are the first to be considered for a match. In order to use keypoints from areas with high occupancy probability, the stability score in Eq. (3) can be weighted with the density d as follows.

$$s(i,j) = d \times s_b(i,j) \tag{5}$$

where d is the occupancy density defined by $d = h/(h+m)$. In addition, one can set a lower bound $ns_{\min}$ for ns (if $ns < ns_{\min}$ then $ns = ns_{\min}$). This can be used to impose a minimal scan number reaching a ROI before considering its contribution fully. We use the score defined in Eq. (5) with $ns_{\min} = 4$. The following subsection details the submap construction based on the keypoint stability score exposed here.

3.1.2. Stable keypoints selection for submap construction

Thanks to the stability score above, we can now select stable keypoints to be included in our submap as scans are concatenated. The data structure of a submap S is defined below.

$$S = \{P, F\} \tag{6}$$

where $P = \{p_0, \cdots, p_{n_p-1}\}$ are the n_p global poses corresponding to the concatenated scans used during the construction of S. F is the set of stable keypoints expressed in the local frame of S named p_{ref} and which corresponds to the first element of P. The local mapping algorithm is given below.

Algorithm 1 builds a submap S starting at scan index i_{init} from a data set containing N laser scans and their corresponding global poses. Functions *ReadLaserData* and *GetPose* respectively output the current local laser reading $\mathcal{L}$ and the corresponding global pose p. Depending whether we use ground truth transforms for scan concatenation or a registration algorithm, *GetPose* may simply read the provided transforms, or employ scan registration to infer the global pose. p_s is the current pose with regard to the submap reference frame p_{ref}. G and S are updated with the functions *UpdateG* and *UpdateS* as follows.

- $UpdateG(G, p_s, \mathcal{L})$
  - Transform $\mathcal{L}$ data in the local frame of S thanks to p_s such as $\mathcal{L}_s = p_s \times \mathcal{L}$
  - Update dimensions of G according to $\mathcal{L}_s$

https://doi.org/10.1017/S0263574722000613 Published online by Cambridge University Press

Algorithm 1 Local mapping

```
Set $G = \{\}$, $P = \{\}$, $F = \{\}$, $p_{ref} = Identity(3)$
for $index = i_{init}$ to $N$ do
    $\mathcal{L} \leftarrow ReadLaserData(\text{index})$
    $p \leftarrow GetPose(\text{index})$
    if $P = \{\}$ then
        $p_{ref} = p$
    end if
    $p_s = p_{ref}^{-1} \times p$
    $G \leftarrow UpdateG(G, p_s, \mathcal{L})$
    $S \leftarrow UpdateS(S, p, G)$
    if $S$ is complete then
        Break
    end if
end for
```

- Update h, m, and ns counters in G with $\mathcal{L}_s$
- Extract FLIRT keypoints from $\mathcal{L}_s$ and save them in the current keypoint list K
- For every keypoint in K
  - ∗ Find its corresponding cell in G, $G(i,j)$ and update its keypoint history H as described in Section 3.1.1.
  - ∗ Compute the corresponding stability score $s(i,j)$ according to Eqs. (3)−(5)

- *UpdateS*(S, p, G)
  - Insert global pose p in the list P
  - For every cell $G(i,j)$ with $H(i,j) \neq \{\}$
    - ∗ If $s(i,j) \geq s_{\min}$ select a keypoint from $H_q(i,j)$ (for instance, the most recent keypoint) and save it in the stable keypoints list K_s
  - Overwrite the list of keypoints F with K_s

3.1.3. Submap completion criteria

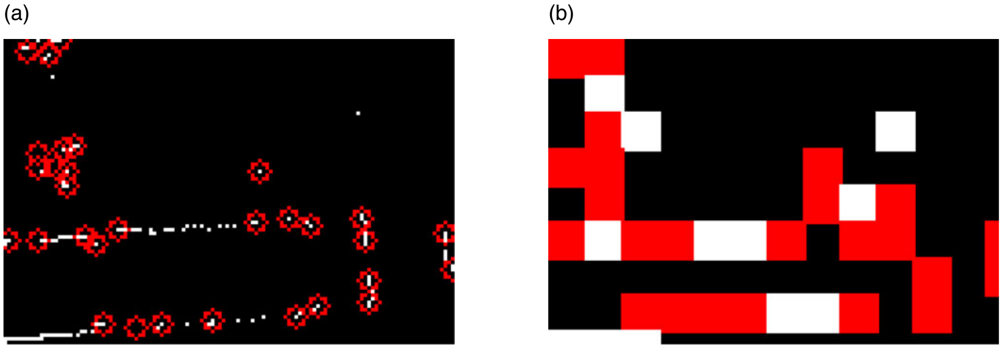
During the mapping process, while exploring the environment, we consider that a robot has finished visiting the current place when overlap ratio ρ between the current and initial submap areas drops below a threshold. There are different ways to estimate this overlap ratio. For instance, it can easily be computed from occupancy grid map as follows.

$$\rho = \frac{a_{\text{int}}}{a_{\text{uni}}} \tag{7}$$

where a_{int} and a_{uni} are respectively, the intersection and the union areas between initial and current occupancy grids. The current occupancy grid map is available in G. This method requires to keep in memory the initial occupancy grid corresponding to the grid built with the first scan only, during submap construction. For the computation of a_{int} and a_{uni} it is enough to use binary grids. Hence, keeping the initial grid will have negligible memory footprint. In a study not reported in this document, we empirically set the overlap ratio threshold $\rho_{\min}$ to 0.75. In addition to this criterion, we introduce a submap quality measure that can serve as an additional criterion for submap termination. It is defined in the the following paragraph.

Keypoint spatial distribution: a submap quality measure. As scans are concatenated, the population of stable keypoints in S evolves. Their spatial distribution with regard to the distribution of laser points in

https://doi.org/10.1017/S0263574722000613 Published online by Cambridge University Press



Example of keypoints from a LiDAR scan

Laser points and keypoints quantified spatial distribution. Red squares contain keypoints and laser points, white squares contain laser points only

Figure 7. *Illustration of spatial distribution quantification for keypoints and laser points.*

the submap is an interesting feature to evaluate the submap quality. In some way, this parameter measures the descriptiveness of the submap. It is defined as follows.

$$\delta = \mathrm{dist}(h_{lp}, h_{kp}) \tag{8}$$

where dist is an histogram distance function, in our case it corresponds to the symmetric χ^2 distance. h_{lp} and h_{kp} correspond to the laser points histogram and the keypoints histogram, respectively. They are 2D binary histograms obtained from the occupancy grid map extracted from G. Figure 7(a) shows the occupancy grid map with laser points in white and keypoints surrounded with red circles. We use another grid D to divide space into cells of size l. In our case l is set to 0.5 m, which represents a suitable spatial discretization step for indoor environments. Laser points histogram h_{lp} is a 2D matrix with the same dimensions as D. When a cell $D(i, j)$ contains a laser point, $h_{lp}(i, j)$ takes value 1, which corresponds to all non black cells in Fig. 7(b), otherwise it is set to 0. The same reasoning is applied to h_{kp}, for which cells take value 1 for keypoints only, corresponding to the red cells in Fig. 7(b). The histograms are normalized with the respective population in each category of points. Thus, the maximum value for the histogram ditance δ is 1.

The submap S is complete if the overlap ratio ρ, defined in Eq. (7) becomes inferior to $\rho_{\min} = 0.75$. Furthermore, the spatial distribution distance defined in Eq. (8) can be used as an additional submap completion criterion, as we will see in Section 5.2.

3.2. Global mapping

We have seen, so far, how to build a local map, corresponding to a place. The global map is a collection of places and is, therefore, represented by a list of submaps with a certain overlapping between successive submaps. The following pseudo-code (Algorithm 2) summarizes the global mapping process that results in the global map M, using N laser scans.

The function *LocalMapping* calls the local mapping Algorithm 1. It takes as argument the index i_{init} and builds a new submap S starting from scan index i_{init}. When S is complete ($S \neq \{\}$) it is inserted as a new element of the list M. The boolean variable *Success* is then set to true, and i_{init} is updated such that the next submap starts at the scan positioned after the last scan in S. Therefore, i_{last} corresponds to the size of P, which gives the number of poses (and the number of scans) in S. The global mapping ends when *LocalMapping* returns an empty submap. When this happens, it means that *LocalMapping* has reached final scan index N in the data set, without being able to complete the submap S. In practice, this occurs when i_{init} becomes too close to N (or becomes larger than N), in such a way that there is not enough scans (or no more scans) left to complete a submap.

https://doi.org/10.1017/S0263574722000613 Published online by Cambridge University Press

Algorithm 2 Global mapping

```
$M = \{\}, S = \{\}, i_{init} = 1$
do
    $Success = False$
    $S \leftarrow LocalMapping(i_{init})$
    if $S \neq \{\}$ then
        $M \leftarrow insertS(M, S)$
        $Success = True$
        $i_{init} = i_{last} + 1$
    end if
while $(Success)$
```

4. Place recognition process description

The previous section detailed our modeling of places with submaps. Place recognition sums up to submap matching. This matching process is followed by a geometric verification step intended to filter out wrong place matches. The following two subsections describe the submap matching process and the geometric verification method.

4.1. Submap matching

A submap consists of a list of stable keypoints. For submap matching we can either match individual keypoints or use a submap signature, for instance in the form of an histogram. In this work, we follow the first approach based on a nearest neighbor pairing. The second approach will thoroughly be studied in another work.

To describe our submap matching algorithm, let us consider two submaps S_r and S_c. The first is the reference submap belonging to the global map M, while the second is the currently built submap and for which we look for a match in M. Our matching algorithm proceeds as follows.

Algorithm 3 Submap matching

```
$S_r = \{P_r, F_r\}, S_c = \{P_c, F_c\}, pairs = \{\}, Success = False$
for every keypoint $kp_c$ in $F_c$ do
    $pair \leftarrow FindUniqueMatch(kp_c, F_r)$
    $pairs \leftarrow Insert(pair)$
end for
if $|pairs| \geq \mu$ then
    $Success = True$
end if
```

where *pair* is a pair formed by keypoint kp_c from F_c and its matched keypoint kp_r from F_r. Keypoints kp_c and kp_r are matched if the distance between their descriptors is below 0.4 and if none of them was already matched before. The set *pairs* contains all unique matches between keypoints from S_r and keypoints from S_c. We remind that coordinates of keypoints in *pairs* are expressed in their respective submap coordinate systems, corresponding to the first elements of P_r and P_c. The two submaps are considered a successful match if there is enough unique matches between them. We will see later how we set the threshold μ.

For the submap S_c, there may be more than one match in M. All successfully matched submaps from M are stored in the candidates set C. This set is sorted by increasing average descriptor distance, between

https://doi.org/10.1017/S0263574722000613 Published online by Cambridge University Press

matched keypoints in the set *pairs*. Hence, the first element in C is the best match with S_c in terms of average descriptor distance.

4.1.1. Pose estimation

Depending whether we perform global localization or SLAM, the global map M is either complete or in progress. In both cases poses in P_r are known and expressed in the global frame. The poses in P_c for submap S_c are expressed with regard to $P_c[0]$ (in the case of global localization) or global but inaccurate (in the case of SLAM). The estimated global pose $\hat{p}$ is obtained using RANSAC [28] which takes as inputs the local keypoints in F_r and F_c, and returns the rigid transform $\hat{T}_r^c$ expressing the pose $P_c[0]$ with regard to $P_r[0]$ ($X[0]$ symbolizes the first element of the set X). $\hat{p}$ is given by Eq. (9)

$$\hat{p} = P_r[0] \times \hat{T}_r^c \times p_c \tag{9}$$

where p_c corresponds to the last element of P_c expressed with regard to $P_c[0]$.

We use RANSAC implementation provided in the FLIRT open source library from ref. [33]. The algorithm randomly selects a pair of descriptor based matches and infers a rigid transform relating their 2D coordinates. Applying this transform to bring the two keypoint sets in the same frame, the algorithm counts the number of inliers having an Euclidean distance below an acceptance threshold (set to 0.1 m in our case, based on the works in ref. [22]). The random selection is repeated for a number of iterations and the consensus corresponds to the largest inliers set, for which the infered transform $\hat{T}_r^c$ is returned as output.

4.2. Geometric verification

In the literature, place recognition usualy includes a geometric verification step, to eliminate wrong place matches. Indeed, the appearance of a place may have more than one match in the global map, which do not necessarily correspond to the same location. This is mainly due to visual aliasing and to environments featuring symmetries. This issue is even worsen in the case of 2D LiDAR data, because of its poor nature. The main idea behind geometric verification is to verify the validity of the place matching using geometric criteria, such as the consistency of the rigid transform relating the matched places. In our case, this verification is applied on the first n_c elements of the set C defined in Section 4.1. We set n_c equal to 500 in order to consider most of the candidate matches during geometric verification. Our geometric verification, described below, is twofold.

4.2.1. First geometric verification

The first level of verification imposes a minimal size for the consensus in RANSAC [28]. In ref. [22], Tipaldi et al. found that the minimal consensus size is 9 pairs. We apply this constraint as a first geometric verification. If it is satisfied, the algorithm proceeds to the second verification.

4.2.2. Second geometric verification

In the first verification, a minimal size for RANSAC [28] consensus is imposed. However, this is not enough since we are not sure that all elements of the consensus constitute descriptor based matches. Indeed, they are paired using Euclidean geometric distance only. For the second verification, let us consider the case of a perfect submap match, resulting in a correct loop transform $\hat{T}_r^c$. Using this transform to bring the keypoints from both submaps in the same coordinate system, we obtain a keypoint overlap area. Our assumption is that most keypoint matches in this area should be correct matches, not only from a geometric point of view, but also with their descriptors. However, since we are not in an ideal case (the detector and descriptor are not ideal, and the scans are not perfectly aligned), we expect that only a certain proportion γ of keypoints in that area has close neighbors that are also matched based on their descriptors. We qualify these points as "doubly” matched keypoints. A pair of keypoints is said

https://doi.org/10.1017/S0263574722000613 Published online by Cambridge University Press

doubly matched when the symmetric χ^2 distance between their descriptors is within 0.4 and the geometric distance between their positions does not exceed the threshold λ, set in Section 5.2. We define the proportion γ in Eq. (10) below.

$$\gamma = \frac{2 \times n_{dm}^{ol}}{n_r^{ol} + n_c^{ol}} \tag{10}$$

where n_{dm}^{ol} stands for the number of doubly matched keypoints (*dm* subscript) in the overlap area (*ol* superscript), n_r^{ol} and n_c^{ol} are, respectively, the number of keypoints from S_r and S_c in that area. The higher is γ the better is the submap match quality. In the ideal case, every keypoint from one submap in the overlap area has a doubly matched keypoint in the other submap, hence $n_{dm}^{ol} = n_r^{ol} = n_c^{ol}$, meaning that γ would be equal to 1, its maximal value. The submap S_r is finally considered a valid match with S_c if γ is beyond a threshold $\gamma_{\min}$, which value is set in Section 5.2. The overlapping area is approximated by the intersection of the bounding boxes delimited by keypoint coordinates from each submap.

5. Parameter setting

In the previous two sections, we detailed our approach through its mapping and place recognition components. In this section, we show how to set parameters in each component. We have identified a total of five parameters listed below. Parameter setting requires running the mapping and place recognition algorithms for a considerable number of parameter combinations. To save execution time, we use the corrected data sets described in Section I. In that case, instead of running ICP for scan registration during submap construction, scans are directly concatenated using the provided transforms. We will show in Section 6 that the parameters found here can be generalized to raw data.

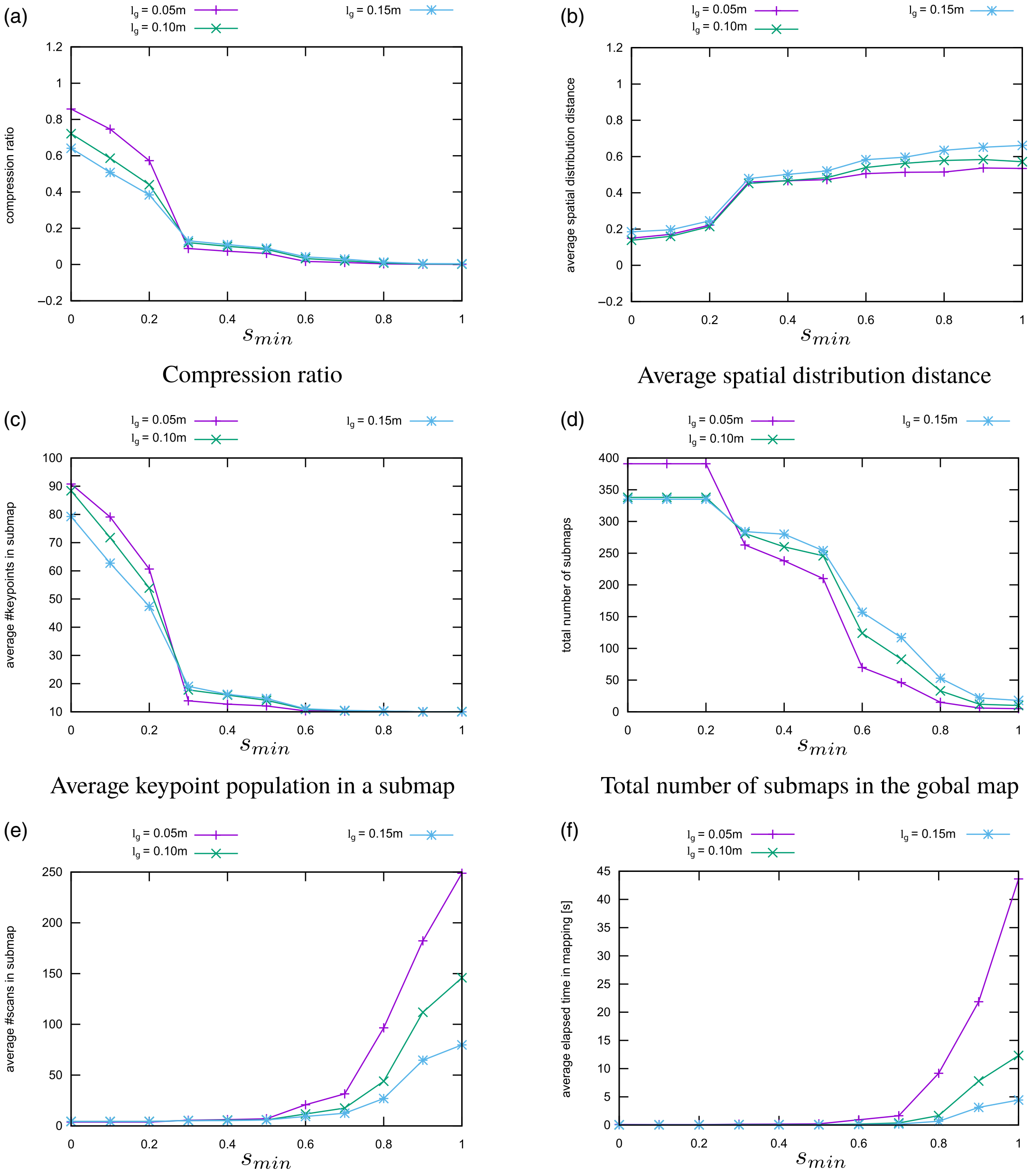
- Mapping parameters
  - l_g : metric cell size in G
  - $s_{\min}$: minimal threshold for keypoint stability score
- Place recognition parameters
  - μ : minimal number of unique keypoint matches required for submap matching
  - $\gamma_{\min}$: minimal ratio of doubly matched pairs in the second geometric verification
  - λ : maximal threshold for geometric distance between doubly matched keypoints in the second geometric verification

5.1. Mapping parameter setting

We assess the mapping algorithm for different values of l_g and $s_{\min}$. In order to find optimal values for these parameters, we define the following quality criteria that the map should meet.

- Compression ratio – The compression ratio is the ratio between the number of stable keypoints in the global map and the total number of keypoints from all scans. The lower it is, the higher is map data reduction.
- Average number of scans in a submap – This criterion measures an average of the number of scans in a submap, from all the submaps in the global map. This gives a sense of how many scans are required to map a place.
- Average number of keypoints in a submap, and
- Average spatial distribution distance – The spatial distribution distance measures the difference between the spatial distribution of keypoints and that of all laser points in a submap, as defined in Eq. (8). The two latter quantities inform us on keypoint population in the submaps. A submap must contain enough keypoints to be distinguishable among other places. On the other hand,

https://doi.org/10.1017/S0263574722000613 Published online by Cambridge University Press



Compression ratio

Average spatial distribution distance

Average keypoint population in a submap

Total number of submaps in the gobal map

Average number of scans required to build a submap

Average elapsed time in building a submap

Figure 8. *Evolution of mapping quality criteria as a function of parameters s_{min} and l_g.*

these keypoints should be well distributed in space. The lower is the spatial distribution distance the better is the submap description of a place.

- Total number of submaps in the global map
- Average elapsed time in building a submap

The six criteria above are evaluated while varying $s_{\min}$ from 0 to 1 with a step of 0.1. This is repeated for three grid cell size values $l_g = 0.05$ m, $l_g = 0.1$ m, and $l_g = 0.15$ m.

https://doi.org/10.1017/S0263574722000613 Published online by Cambridge University Press

Table II. *Mapping criteria results for optimal parameter values, and for the three corrected maps. Compress. corresponds to the average compression ratio, # Scans is the average number of scans required to build a submap, # Keypoints is the average number of stable keypoints in a submap, Distrib. is the average spatial distribution distance, and # Submaps is the number of submaps in the global map. Time is the average spent time in submap construction.*

Data set	**Compress.**	**#Scans**	**#Keypoints**	**Distrib.**	**#Submaps**	**Time (s)**
FR079	0.12	5.20	17.80	0.45	281	0.057
FR101	0.12	4.04	14.79	0.54	72	0.109
Intel	0.15	5.83	16.04	0.46	457	0.075

We want to maximize keypoint population in a submap while minimizing the compression ratio, the spatial distribution distance, the number of scans in a submap and the elapsed time in building a submap. Observing Fig. 8, the values $s_{\min} = 0.3$ and $l_g = 0.1$ represent a good tradeoff to reach this objective. Table II summarizes the results obtained for all data sets using these parameter values. The compression ratio is in the worst case equal to 0.15, corresponding to a reduction of 85% in map data, which is significant. This reduction reaches 88% in the best case. Less than six scans, in average, are needed to map a place. Submap keypoint population is beyond 14 keypoints with spatial distribution distance not exceeding 0.54. This value is acceptable, considering that the maximal value for this distance is 1. Submaps are built with a rate of 9.2 Hz in the worst case and 17.5 Hz in the best case.

5.2. Place recognition parameter setting

There are three main parameters in the place recognition process. The first is the minimal number of unique keypoint matches, μ in the submap matching step described in Section 4.1. The second parameter is the minimal ratio of doubly matched pairs $\gamma_{\min}$ in the second geometric verification described in Section 4.2.2. The third parameter is the maximal geometric distance λ separating a pair of doubly matched keypoints, in the second geometric verification as well. In order to set optimal values for these parameters, we measure performance of our algorithm in terms of precision and recall, in the context of global localization. A global map M is built beforehand using the corrected data. Then, using these data a submap S_c is started from an arbitrary global position, unknown to S_c. In this case, $P_c[0]$ is the identity matrix. Finally, the place recognition algorithm provides an estimate of the global pose $\hat{p}$ by finding for S_c the matching submap S_r in the global map M, as described in Section 4.1.1. In order to avoid automatching with an identical submap from M, if a matched submap S_r is built with the exact same scans as S_c, the match is ignored. A similar evaluation process is used in the literature, such as in refs. [20–22, 25].

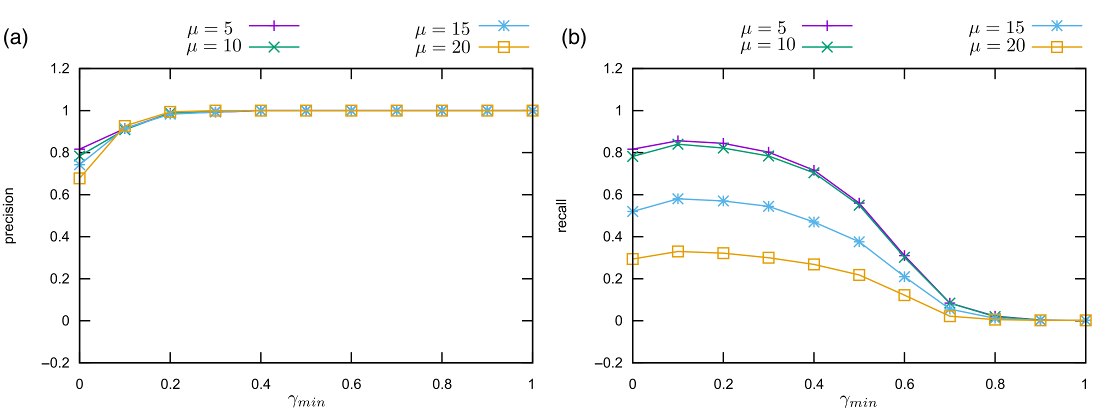
The algorithm is assessed for 4 values of μ and 11 values of $\gamma_{\min}$, respectively $\{5, 10, 15, 20\}$, and $[0, 1]$ with a step of 0.1. This experiment is run for $\lambda = 0.05$ m. In a later stage, the obtained optimal parameters are used for performance evaluation with $\lambda = 0.1$ m. In addition to these parameters, we impose a minimal number of keypoints in a submap to be consistent with the minimal required size for RANSAC's consensus (set to 9). Therefore, we decide to set 10 as a minimal size for the submap, in terms of keypoint population. On the other hand, based on previous analysis on the mapping process given in Section 5.1, in addition to the drop in overlap ratio, we impose another criterion for submap completion that is the spatial distribution distance δ, which should not exceed 0.5, in order to guarantee an optimal spatial distribution of keypoints. We remind that, based on the results obtained in Section 5.1, the average submap's keypoint population is beyond 10 and the average spatial distribution is around 0.5. This evaluation is performed using the data set *FR*079, then the obtained parameters are used for *FR*101 and *Intel* data sets.

The pseudo code for global localization evaluation, used here, is given in Algorithm 4. The function *PlaceRecognition* combines submap matching and geometric verification. The function *Evaluation*

https://doi.org/10.1017/S0263574722000613 Published online by Cambridge University Press

Algorithm 4 Global localization evaluation

```
$M = \{S_{r_0} \cdots S_{r_m}\}, S_c = \{P_c, F_c\}, P_c = \{\}, F_c = \{\}$
for $iteration = 1$ to $n_{pos}$ do
    $i_{init} \leftarrow$ GetStartingIndex$(iteration, n_{pos})$
    $S_c \leftarrow LocalMapping(i_{init})$
    if $S_c \neq \{\}$ then
        $(S_r, \hat{T}_r^c, \hat{p}) \leftarrow$ PlaceRecognition$(M, S_c)$
        (precision, recall) $\leftarrow$ Evaluation$(p, \hat{p})$
    else
        Break
    end if
end for
```



Precision Recall

Figure 9. *Global localization – Precision and recall for FR079, $\lambda = 0.05$ m.*

computes precision and recall by comparing $\hat{p}$ with p, respectively the estimated pose given by Eq. (9), and ground truth pose. The pose estimate $\hat{p}$ is considered correct if the translation and rotation errors, with regard to p, are within 0.5 m and 10°, respectively [20–22, 25, 26]. The global localization is performed n_{pos} times corresponding to equally spaced starting positions for the current submap S_c. In our case, $n_{\text{pos}} = 500$. The average precision and recall is computed over n_{pos}, for all combinations of place recognition algorithm parameters. Precision and recall are defined as

$$\text{precision} = \frac{n_{\text{cor}}}{n_{\text{acc}}} \tag{11}$$

$$\text{recall} = \frac{n_{\text{cor}}}{n_{\text{pos}}} \tag{12}$$

n_{cor} and n_{acc} correspond respectively, to the number of truly correct localizations (true positives), and the number of localizations considered correct (accepted) by the algorithm (true positives + false positives).

Figure 9 shows precision and recall obtained for *FR*079 data set with $\lambda = 0.05$ m. In terms of precision, the results obtained with the four values of μ are similar and reach the maximum precision value 1, although one can notice that the precision is better for larger values of μ and γ_{min} together. The impact of μ and γ_{min} is more visible when observing recall results. For $\mu = 10$ and $\gamma_{\text{min}} = 0.4$ we obtain a recall and precision tradeoff that maximizes precision. Table III summarizes results obtained applying these

https://doi.org/10.1017/S0263574722000613 Published online by Cambridge University Press

Table III. *Global localization results for corrected data, obtained with $\mu = 10$, $\gamma_{min} = 0.4$, and $\lambda = 0.05$ m.*

Data set	Precision	Recall	# Scans	Time (s)
FR079	1	0.71	6.05	0.067
FR101	1	0.67	6.98	0.020
Intel	1	0.73	7.81	0.088

Table IV. *Global localization results for corrected data, obtained with $\mu = 10$, $\gamma_{min} = 0.4$, and $\lambda = 0.1$ m.*

Data set	Precision	Recall	#Scans	Time(s)
FR079	0.99	0.83	5.98	0.065
FR101	1	0.81	6.86	0.021
Intel	0.99	0.82	7.76	0.086

Table V. *Optimal parameters.*

Mapping parameters		Place recognition parameters		
l_g (m)	s_{min}	μ	γ_{min}	λ (m)
0.1	0.3	10	0.4	0.05

values to the other corrected data sets. For all three data sets, we obtain a good recall level at precision of 1, which is particularly recommended for loop closure detection in SLAM.

Table IV gives the results obtained for $\lambda = 0.1$ m. Comparing with the results in Table III, increasing this parameter improves recall results while keeping the precision at a high level. The last two columns in Tables III and IV inform us on the average number of scans required to perform a correct localization and the corresponding spent time.

5.3. Summary of parameter setting

We summarize, here, the results obtained for the five parameters in our approach. Table V gives parameter values for the mapping and place recognition algorithms. These parameters are imposed by mapping quality criteria and place recognition objectives. For the mapping stage, we have defined in Section 5.1 six criteria among which the most important, in our case, is the map data compression ratio. Place recognition parameter setting relies on precision and recall optimization in the context of global localization. We made the choice of maximizing precision while keeping an acceptable recall level.

6. Evaluation and discussions

In this section, we apply the mapping and place recognition algorithms on raw data. Algorithm parameters are set to the values obtained in previous section and summarized in Table V. The results obtained for each process are discussed in the next two subsections.

6.1. Mapping evaluation

We apply the mapping algorithm on the raw data sets $FR079\_r$, $FR101\_r$, and $Intel\_r$, using parameter values obtained from previous section, that is, $s_{\text{min}} = 0.3$ and $l_g = 0.1$. Scans are registered with ICP. The results are given in Table VI. Comparing with the results in Table II, we first notice an increase in the

https://doi.org/10.1017/S0263574722000613 Published online by Cambridge University Press

Table VI. *Mapping criteria results for optimal parameter values and for the three raw data sets.*

Data set	Compress.	#Scans	#Keypoints	Distrib	#Submaps	Time (s)
FR079_r	0.12	7.95	26.93	0.37	620	0.31
FR101_r	0.14	9.95	40.91	0.35	478	0.46
Intel_r	0.15	7.69	20.30	0.40	1772	0.26

Table VII. *Global localization results for raw data, obtained with* $\mu = 10$, $\gamma_{min} = 0.4$, *and* $\lambda = 0.05$ *m.*

Data set	Precision	Recall	#Scans	Time (s)
FR079_r	0.99	0.59	12.74	0.117
FR101_r	1	0.22	9.97	0.041
Intel_r	1	0.60	15.30	0.136

number of scans required to build a submap. This is mainly due to the density of raw data in terms of scans, which also explains the increase in the number of submaps in the global map. As a consequence, the average keypoint population is higher, with a better spatial distributions ($\delta = 0.4$ for the worst case). The compression ratio remains at a very low level, corresponding to 85% of map data reduction in the case of *Intel_r*, and 88% for *FR*079*_r*. Execution time shows that we can build submaps, in the worst case at a rate of 2.17 Hz, and 3.85 Hz for the best case. The mapping process takes more time here, due to scan registration with ICP and to a larger number of scans in the submaps.

6.2. Place recognition evaluation

Place recognition is evaluated using raw data for two applications. The first is global localization and the second is loop closure detection in the context of SLAM.

6.2.1. Global localization

Similarly to Section 5.2, a global map M is built using the corrected data, then starting from an unknown position, a submap S_c is built in order to perform global localization using our place recognition approach. The main difference with the experiment in Section 5.2 is that S_c is built using raw data. Scans in S_c are registered with ICP. Not all raw poses have a corresponding ground truth pose among corrected data. In order to measure precisison and recall, we need a mapping that relates each ground truth pose with its corresponding raw pose. Submap S_c starts at arbitrary positions among raw poses for which a ground truth value exists. ICP is initialized with the odometry provided in raw data. We run the global localization algorithm using parameters from previous section, that is, $\mu = 10$, $\gamma_{\text{min}} = 0.4$, and $\lambda = 0.05$. An additional evaluation is performed for $\lambda = 0.1$. Mapping algorithm parameters are those given in Table V.

Comparing the results in Tables VII and VIII with the results in Tables III and IV, we notice that the precision remains very high (almost 1) in all cases but the recall level decreased. This is particularly true for *FR*101*_r* data set, when using a value of 0.05 m for λ. It takes more scans, in the raw data case, to correctly localize the robot, which is expected since in this case the displacements are smaller. Compared to the case of corrected data, localization takes more time due to higher keypoint population in raw data submaps, and larger global maps in term of submap population.

Comparing the results in Tables VII and VIII, the same conclusion as in Section 5.2 can be drawn regarding how λ impacts recall results. For *FR*101*_r* with $\lambda = 0.1$ m, recall has significantly improved (more than doubled) in comparison with the recall obtained for $\lambda = 0.05$ m. Considering the last column

https://doi.org/10.1017/S0263574722000613 Published online by Cambridge University Press

Table VIII. *Global localization results for raw data, obtained with* $\mu = 10$, $\gamma_{min} = 0.4$, *and* $\lambda = 0.1$ *m.*

Dataset	Precision	Recall	#Scans	Time (s)
FR079_r	0.99	0.80	11.99	0.112
FR101_r	0.99	0.49	10.24	0.044
Intel_r	0.99	0.80	15.21	0.133

in Tables VI−VIII, we can map a place and correctly localize the robot at a rate of 2.55 Hz in the best case, and 1.98 Hz in the worst case.

6.2.2. Loop closure

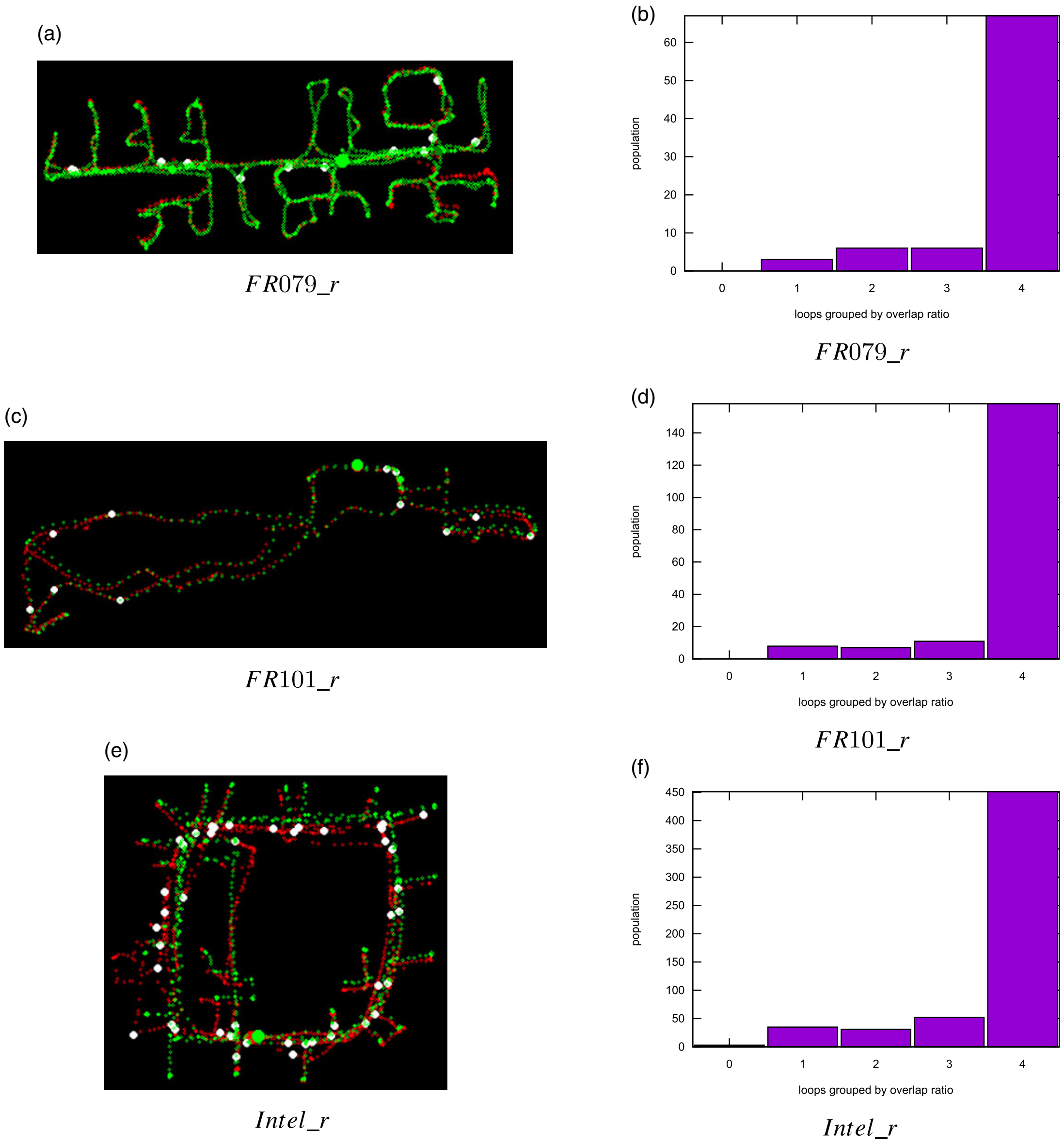
Here, we are interested in evaluating the possibility of integrating our place recognition approach in a SLAM framework, for loop closure detection using parameters in Table V. To this end, we employ a SLAM implementation provided in the GTSAM open source library [34]. It is a pose-graph SLAM variant based on factor graphs. For our evaluation we use this tool with its defaults parameters, except for covariances that we have set empirically. Furthermore, only submap poses are included in the graph and no features are used in the graph optimization process. Therefore, we do not expect to build precise maps. However, with this setup, we aim at evaluating our approach by examining the produced trajectories using raw data. We remind that for these data, not all poses have a corresponding ground truth in the corrected data. There is no guaranty that a submap built with raw data contains a pose for which the ground truth is known. Therefore, in order to decide wether a loop is correct, we rely on overlap rate between matched submaps. The overlap rate is also used in ref. [17] to decide for correct submap matches. Obviously, this assumes a limited drift due to odometry and scan registration errors. Considering this assumption, we reject all matches of submap pairs with no overlap, and we impose a minimal overlap ratio of 0.01 between matched submaps.

Loop closure and mapping analysis. Figure 10(a,c,e) depict the trajectories obtained with SLAM using our place recognition approach for loop closure, respectively for *FR*079*_r*, *FR*101*_r*, and *Intel_r* data sets. The trajectory drawn with red dots correspond to estimated submap poses. As a reference, for visual comparison, we display the trajectory produced by the corrected data with green dots.

For *FR*079*_r* and *FR*101*_r*, the trajectory obtained with SLAM is consistent with the correct trajectory. As expected, due to pose drift and to possible false loop closures, it does not exactly match the ground truth trajectory. In the case of *Intel_r*, the trajectory is globally consistent. However, because of an erroneous loop closure, we notice an inconsistency at the bottom left side of Fig. 10(e). We remind that our pose-graph considers poses only, and does not include features, which impacts the graph SLAM performance. In addition, among all detected loops, only a small subset of loops is used in the graph optimization. Since we update the graph each time a new loop is detected in order to update the global map M, we do not use all the detected loops, as these two update operations are time consuming.

Figure 10(b,d,f) show histograms of overlap rate between all matched submaps for each data set. The histogram has five bins corresponding to overlap rates in the following intervals $[0, 10[$, $[10, 20[$, $[20, 30[$, $[30, 40[$, and $[40, 100]$. For all data sets, most matched submaps overlap with a rate larger or equal to 40%. For *FR*079*_r*, a total of 82 loops are detected while 184 are detected in *FR*101*_r* and 572 in *Intel_r* (among all these loops, only a limited number of 13, 12, and 47 loops are used in the pose graph, respectively for *FR*079*_r*, *FR*101*_r*, and *Intel_r*). In a pessimistic evaluation, if we consider that 30% overlap is required to decide for a correct match, then the precision values are respectively $73/82 = 0.89$, $169/184 = 0.92$, and $503/572 = 0.88$ for the three maps. In a more tolerent evaluation, if 10% overlap rate is enough to decide for a correct loop closure, then the precision values raise to $82/82 = 1$, $184/184 = 1$, and $569/572 = 0.99$, respectively. In both evaluations, *Intel_r* gives the lowest precision value.

https://doi.org/10.1017/S0263574722000613 Published online by Cambridge University Press



*FR*079_*r*

*FR*079_*r*

*FR*101_*r*

*FR*101_*r*

Intel_r

Intel_r

Figure 10. *Left: SLAM trajectory in red versus ground truth trajectory in green. White discs represent positions of loop detections used in the pose graph, large green disc is the initial position, small green disc is the first loop detection position. Right: matched submap overlap histograms. Results for FR*079_*r*, *FR*101_*r*, *and Intel_r are given in each row, respectively.*

Table IX gives map characteristics for the three data sets. Compared to Table VI, all maps feature better map data reduction (up to 92% for *FR*079) with less submaps in the global map. This is due to map correction with SLAM, which also explains the decrease in submap keypoints population. We notice also a slight improvement in keypoints spatial distribution.

Trajectory error analysis. As an additional evaluation, we measure RPE (relative position error) and ATE (absolute trajectory error) for all trajectories. The RPE gives a sense of the drift for different time

https://doi.org/10.1017/S0263574722000613 Published online by Cambridge University Press

Table IX. *Mapping criteria results for optimal parameters, and for the three corrected maps with SLAM using raw data.*

Data set	Compress.	#Scans	#Keypoints	Distrib.	#Submaps	Time (s)
*FR*079*_r*	0.08	9.17	21.52	0.35	530	0.34
*FR*101*_r*	0.11	10.71	34.74	0.34	444	0.54
Intel_r	0.11	9.56	19.10	0.37	1421	0.30

intervals between two poses of the robot. It is defined in Eq. (13)

$$E_i := (Q_i^{-1}Q_{i+\Delta})^{-1}(P_i^{-1}P_{i+\Delta}) \tag{13}$$

with Q_i and P_i being respectively ground truth and estimated poses at time index i. Δ is a fixed time interval. For a trajectory with n poses, we obtain $m = (n - \Delta)$ RPE values along the trajectory [36]. In the literature, authors often evaluate the RMSE (root mean square error) obtained for different values of Δ and define in Eq. (14). Note that in our case, Δ corresponds to data index interval.

$$\mathrm{RMSE}(E_{1:n}, \Delta) := \left(\frac{1}{m}\sum_{i=1}^{m}\left(\|\mathrm{Trans}(E_i)\|^2\right)\right)^{\frac{1}{2}} \tag{14}$$

The operator Trans returns the translation component of error matrix E_i. One can also obtain the RMSE of the rotation RPE by infering the rotation from E_i.

The ATE measures absolute translation and rotation errors between the estimated poses and the corresponding ground truth poses. It is defined as follows.

$$F_i := Q_i^{-1}AP_i \tag{15}$$

where A is the transform that maps the estimated trajectory on the ground truth trajectory [36]. Figure 11 shows RMSE of RPE results for each data set. For *FR*079*_r* and *FR*101*_r* translation and rotation errors do not exceed 1.3 m and 9°, respectively. In comparison, the case of *Intel_r* data set exhibits higher error levels, especially for the rotation. Figure 12 shows ATE errors for each data set, as a function of path length. Similar conclusions can be drawn for the quality of trajectories that is better for *FR*079*_r* and *FR*101*_r*, compared to *Intel_r*. It is interesting to notice the peaks in ATE errors for *Intel_r*, especially between path lengths 150 and 200 m. This corresponds to erroneous loops which consequence on the trajectory can be observed on Fig. 10(e) (bottom left side). This explains the high RPE errors for this map. This also shows that the algorithm manages to find new correct loops that help to rectify the trajectory.

6.3. Discussions with regard to state-of-the-art

In the previous subsections, we have shown that we can achieve a high level of map data reduction while maintaining good performance in terms of precision and recall. In the literature, most local feature based methods in this field encode a place with a single scan, such as in refs. [20–25]. In that case, the global map contains all features from all scans. As a consequence, the map data size inevitably keeps growing with the number of laser readings. For larger environments, it is of great interest to reduce the map data. The work of Bosse and Zlot [16, 17] is the most similar to ours as they model places with local maps or submaps. Submaps contain keypoints selected from concatenated laser scans. The authors provide no significant results on map data reduction. They terminate a submap when the state vector (used in local mapping through a Kalman filter) reaches a certain size. In our work, a keypoint stability score allows to detect areas with persistent features from concatenated scans, and achieve considerable map data reduction. We use the state-of-the-art operator FLIRT, which has shown interesting performance in refs. [21, 22]. Regarding submap termination, we introduce a submap quality measure based on the spatial distribution of keypoints. This metric together with the overlap rate drop between the current and

https://doi.org/10.1017/S0263574722000613 Published online by Cambridge University Press

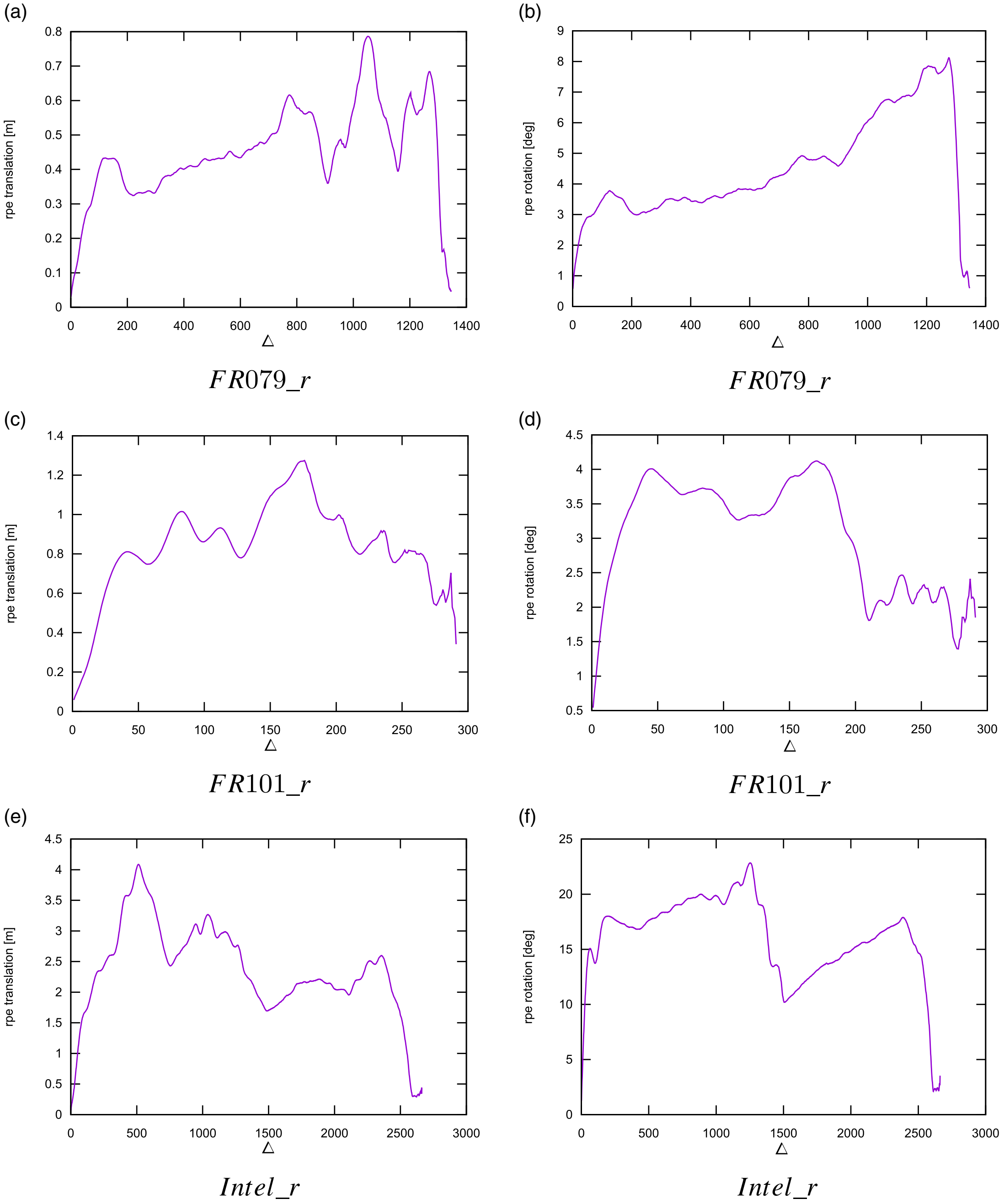
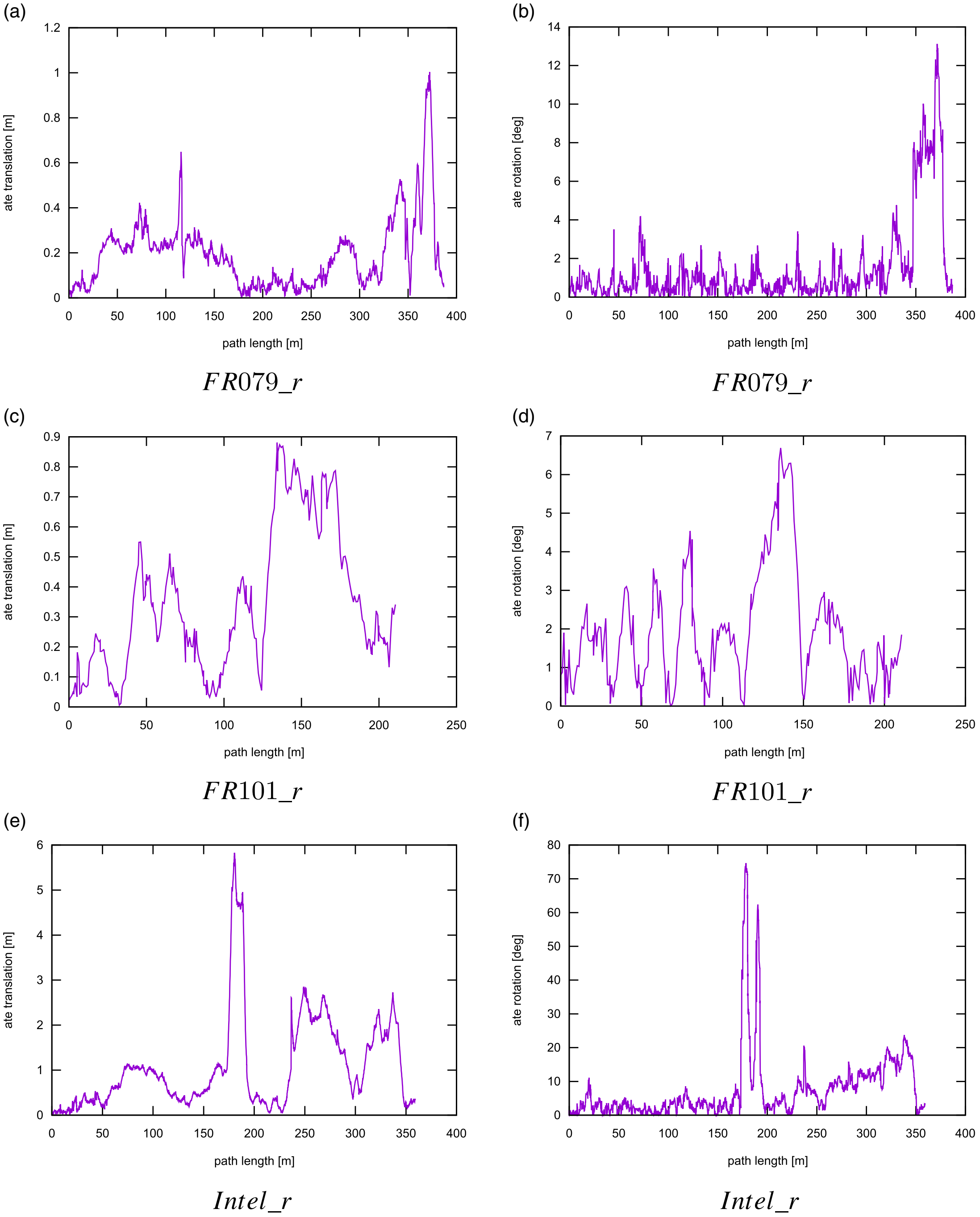


Figure 11. *RMSE of RPE as a function of* Δ. *Left: translation error. Right: rotation error. Results for FR*079_*r, FR*101_*r, and Intel*_*r are given in each row, respectively.*

initial submap areas, serve as criteria for submap completion. Instead of a rigid criterion, our solution is adaptive. It allows to set a minimal overlap between consecutive submaps, and insures that a place is well described before starting a new submap.

Among prominent works on local feature based place recognition using 2D LiDAR data, is that of Tipaldi et al. [20–22]. The authors introduced the FLIRT operator used in our work. They measure two



$FR079\_r$ $FR079\_r$

$FR101\_r$ $FR101\_r$

$Intel\_r$ $Intel\_r$

Figure 12. *ATE as a function of path length. Left: translation error. Right: rotation error. Results for FR079_r, FR101_r, and Intel_r are given in each row, respectively.*

metrics for performance evaluation, namely P_{GL} and P_{CL}. The first is the probability of correct global localization, and it corresponds to precision at maximal recall. The second metric is the probability of loop closure given by recall value at 0.95 precision. As stated before, the authors encode a place with features from a single scan and no map data reduction is considered (the global map contains all features

https://doi.org/10.1017/S0263574722000613 Published online by Cambridge University Press

Table X. *Performance comparison with Tipaldi et al. approach [22]. Data reduc. stands for map data reduction rate. Ours-X-Y is our approach where $\mu = X$ and $\lambda = Y$. TIP-FLIRT is Tipaldi's approach. For easy comparison, TIP-FLIRT results and the best values obtained with our approach are shown in boldface.*

	FR079			Intel		
	P_{GL}	P_{CL}	Data reduc. (%)	P_{GL}	P_{CL}	Data reduc. (%)
OURS-5-0.05	0.88	0.80	**88**	0.90	0.83	**85**
OURS-5-0.1	**0.91**	**0.84**		**0.98**	**0.86**	
OURS-10-0.05	0.89	0.79		**0.98**	**0.80**	
OURS-10-0.1	**0.91**	**0.83**		0.91	0.83	
TIP-FLIRT	**0.98**	**0.98**	**0**	**0.96**	**0.96**	**0**

from every scan). We have evaluated P_{GL} and P_{CL} for data sets $FR079$ and *Intel*, since these data sets are also used in ref. [22]. This evaluation corresponds to the global localization performed in Section 6.2.1. Table X summarizes the results obtained for our approach in comparison with Tipaldi's residual based approach from ref. [22]. We remind that for our evaluation we compare a submap built using raw data (scans registered with ICP) with a global map built from corrected data, whereas Tipaldi's evaluation compares a scan from the corrected data with the remaining scans from the same corrected data. Despite this condition, the results show that even with a considerable map compression rate (88% for $FR079$ and 85% *Intel* data sets, respectively) our approach maintains high loop closure and global localization performance. This conclusion is also valid when comparing to the work of Himstedt et al. [23, 24] on GLARE features and GSR signature. The authors obtain good recall and precision levels but require complex signature calculations and do not consider map data reduction the way it is done here. Finally, approaches such as GFP [20] and in ref. [25] encode scans with words similarly to bag of words, and therefore require training data sets to generate vocabularies, which is not the case of our approach.

7. Conclusion and future works

We have prensented in this article a new approach for place recognition with 2D LiDAR data. Our method allows to achieve considerable map data reduction while maintaining high place recognition performance. Places are modeled with submaps obtained from multiple registered laser scans. A submap describes the local environment with stable keypoints. Our keypoint selection strategy extracts persistent keypoints online among all keypoints from the registered scans. This process eliminates unstable keypoints, and significantly reduces data redundency in the global map, by keeping only one keypoint instance from regions of interest with persistent keypoints.

Submap termination criteria constitute another contribution of our work. In addition to the drop in overlap between the current and initial submap areas, we decide that a submap is complete when stable keypoints are well distributed in space. This is done thanks to the spatial distribution distance, which measures the difference between keypoints and laser points distributions.

The main components of our approach are the mapping and the place recognition processes. The place recognition process includes a submap matching step followed by a two steps geometric verification. In a first stage, parameter setting for each process is performed using corrected version of three popular real data sets $FR079$, $FR101$, and *Intel*, corresponding to Freiburg University buildings 079 and 101, and Seattle's Intel Research Laboratory, respectively. For the mapping process, parameters were optimized with regard to six map quality criteria among which are the map data reduction and keypoints spatial

https://doi.org/10.1017/S0263574722000613 Published online by Cambridge University Press

distribution. For the place recognition process, parameters were optimized as a function of recall and precision in the context of global localization.

In a second stage, the mapping and place recognition processes were evaluated with raw versions of the three data sets above with the same parameter values obtained in the parameter setting stage. The mapping results show high data reduction levels (up to 92%) with consistent spatial distribution of keypoints. Regarding place recognition performance, the approach is evaluated for global localization and loop closure. The first evaluation shows that even with considerable map data reduction, the performance in terms of precision and recall is comparable to state-of-the-art methods, which do not consider map data reduction. In the second evaluation we show how our approach can be integrated in a SLAM framework for loop closure detection.

As future works we aim at improving the second geometric verification. Instead of a rigid value, the threshold λ should adapt to data range in order to be less sensitive to rotation errors. For instance, in the case of open spaces (such as in *FR*101), we notice that this threshold has bigger impact on recall performance than in narrow environments (such as *FR*079 and *Intel*). We also aim at improving the algorithm's execution time during submap matching process. This can be achieved by the use of a suitable submap signature allowing for tree-based nearest neighbor search structures, such as kd-trees. Stable keypoints quantified in a signature, like histograms, may improve performance in terms of precision and recall as well.

Author contribution. This paper represents Lounis Douadi's research work under the supervision of Yohan Dupuis, and Pascal Vasseur. This document was written by Lounis Douadi in collaboration (including document organization and reviews) with Yohan Dupuis, and Pascal Vasseur.

Financial support. This work has been funded with the support from the European Union with the European Regional Development Fund (ERDF) and from the Regional Council of Normandie

Competing interest declaration. Lounis Douadi is with Université de Rouen Normandie, LITIS, Avenue de l'Université 76800 Saint-Étienne-du-Rouvray, France. Yohan Dupuis is with CESI, LINEACT Paris La défence, France. Pascal Vasseur is with Université de Picardie Jules Verne, Laboratoire MIS, 33 rue Saint-Leu 80039 Amiens, France.

Ethical considerations. Not applicable.

References

[1] E. Fernández-Moral, W. W. Mayol-Cuevas, V. Arévalo and J. González, "Fast Place Recognition with Plane-Based Maps," **In:** *Proceedings of International Conference on Robotics and Automation*, Karlsruhe, Germany (2013) pp. 2719–2724.

[2] S. Sivic and A. Zisserman, "Video Google: Efficient Visual Search of Videos," **In:** *Toward Category-Level Object Recognition*, vol. **4170**, (2006) pp. 127–144.

[3] D. Nistér, H. Stewénius, "Scalable Recognition with a Vocabulary Tree," **In:** *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, New York, USA*, vol. **2**, (2006) pp. 2161–2168.

[4] M. Cummins and P. Newman, "FAB-MAP: Probabilistic localization and mapping in the space of appearance," *Int. J. Robot. Res.* **27**, 647–665 (2008).

[5] M. Cummins and P. Newman, "Appearance-only SLAM at large scale with FAB-MAP 2. 0," *Int. J. Robot. Res.* **30**, 1100–1123 (2011).

[6] H. Bay, T. Tuytelaars and L. Van Gool, "SURF: Speeded Up Robust Features," **In:** *ECCV 2006* (2006), Graz, Austria, pp. 404–417.

[7] A. Angeli, D. Filliat, S. Doncieux and J. A. Meyer, "A fast and incremental method for loop-closure detection using bags of visual words," *IEEE Transactions on Robotics*, **24**, 1027–1037 (2008).

[8] E. Stumm, C. Mei, S. Lacroix, J. Nieto, M. Hutter and R. Siegwart, "Robust Visual Place Recognition with Graph Kernels," **In:** *CVPR*, Las Vegas, USA (2016) pp. 4535–4544.

[9] M. Milford and G. Wyeth, "RSeqSLAM: Visual Route-Based Navigation for Sunny Summer Days and Stormy Winter Nights," **In:** *Proceedings of IEEE ICRA* (2012), St-Paul, USA, pp. 1643–1649.

[10] S. Siam and H. Zhang, "Fast-SeqSLAM: A Fast Appearance Based Place Recognition Algorithm," **In:** *Proceedings of IEEE ICRA* (2017), Singapore, pp. 5702–5708.

[11] M. Magnusson, H. Andreassonn, A. Nuchter and A. J. Lilienthal, "Appearance-Based Loop Detection from 3D Laser Data Using the Normal Distributions Transform," **In:** *Proceedings of IEEE ICRA* (2009), Kobe, Japan, pp. 23–28.

https://doi.org/10.1017/S0263574722000613 Published online by Cambridge University Press

[12] M. Magnusson, H. Andreasson, A. Nuchter and A. J. Lilienthal, “Automatic Appearance-Based Loop Detection from 3D Laser Data Using the Normal Distributions Transform,” **In:** *Proceedings of IEEE ICRA* (2009), Kobe, Japan, vol. 26, pp. 892–914.
[13] K. Granström and T. Schön, “Learning to Close the Loop from 3D Point Clouds,” **In:** *Proceedings of IEEE IROS* (2010), Taipei, Taiwan, pp. 2089–2209.
[14] N. Muhammad and S. Lacroix, “Loop Closure Detection Using Small-Sized Signatures from 3D LiDAR Data,” **In:** *Proceedings under IEEE International Workshop on Safety, Security, and Rescue Robotics* (2011), Kyoto, Japan.
[15] J. G. Rogers III and J. M. Gregory, Have I Been Here Before? A Method for Detecting Loop Closure With LiDAR, Tech. Rep. DEC-TR-506, U.S. Army Research Laboratory (2015).
[16] M. Bosse and R. Zlot, “Map Matching and Data Association for Large-Scale Two-Dimensional Laser Scan-Based SLAM,” **In:** *IJRR (International Journal of Robotics Research)*, vol. 27 (2008) pp. 667–691.
[17] R. Zlot and M. Bosse, "Place Recognition Using Keypoint Similarities in 2D Lidar Maps," **In:** *ISER (International Symposium on Experimental Robotics)* (2008), Athens, Greece.
[18] M. Bosse and R. Zlot, “Keypoint design and evaluation for place recognition in 2D lidar maps,” *Robot. Auton. Syst.* **57**(12), 1211–1224 (2009).
[19] K. Granström, J. Callmer, F. Ramos and J. Nieto, “Learning to Detect Loop Closure from Range Data,” **In:** *Proceedings of IEEE of ICRA* (2009), Kobe, Japan, pp. 15–22.
[20] G. D. Tipaldi, L. Spinello and W. Burgard, “Geometrical FLIRT Phrases for Large Scale Place Recognition in 2D Range Data,” **In:** *Proceedings of IEEE of ICRA* (2013), Karlsruhe, Germany, pp. 2693–2698.
[21] G. D. Tipaldi and K. O. Arras, “FLIRT - Interest Regions for 2D Range Data,” **In:** *2010 IEEE International Conference on Robotics and Automation* (2010), Anchorage, USA, pp. 3616–3622.
[22] G. D. Tipaldi, M. Braun and K. O. Arras, *FLIRT: Interest Regions for 2D Range Data with Applications to Robot Navigation*, ISER (2010), New Delhi, India.
[23] M. Himstedt and E. Maehle, “Geometry Matters: Place Recognition in 2D Range Scans Using Geometrical Surface Relations,” **In:** *Proceedings of IEEE ECMR* (2009), Mlini/Dubrovnik, Croatia, pp. 1–6.
[24] M. Himstedt, J. Frost, S. Hellbach, H. Bohme and E. Maehle, “Large Scale Place Recognition in 2D LiDAR Scans Using Geometrical Landmark Relations,” **In:** *Proceedings of IEEE IROS* (2014), Chicago, USA, pp. 5030–5035.
[25] J. Deray, J. Solà and J. A. Cetto, “Word Ordering and Document Adjacency for Large Loop Closure Detection in 2D Laser Maps,” **In:** *IEEE Robotics and Automation Letters* (2017).
[26] D. L. rizzini, F. Galasso and S. Caselli, “Geometric relation distribution for place recognition,” *IEEE Robot. Autom. Lett.* **4**(2), 523–529 (2019).
[27] T. Shan, B. Englot, F. Duarte, C. Ratti and D. Rus, Robust Place Recognition using an Imaging LiDAR. ArXiv, abs/2103.02111.
[28] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Commun. ACM* **24**(6), 381–395 (1981).
[29] P. J. Besl and N. D. McKay, “A method for registration of 3-D shapes,” *IEEE Trans. Pattern Anal. Mach. Intell.* **14**(2), 239–256 (1992).
[30] S. Naudet-Collette, K. Melbouci, V. Gay-Bellile, O. Ait-Aider and M. Dhome, “Constrained RGBD-SLAM,” *Robotica* **39**(2), 277–290 (2021).
[31] http://www2.informatik.uni-freiburg.de/ stachnis/datasets.html.
[32] https://libpointmatcher.readthedocs.io/en/latest/.
[33] https://openslam-org.github.io/flirtlib.html.
[34] https://gtsam.org/.
[35] https://github.com/tipaldi/flirtlib.
[36] https://www.ybliu.com/2020/07/visual-slam-evaluation-aterte.html.

Cite this article: L. Douadi, Y. Dupuis and P. Vasseur (2022). “Stable keypoints selection for 2D LiDAR based place recognition with map data reduction”, Robotica **40**, 3786–3810. https://doi.org/10.1017/S0263574722000613

https://doi.org/10.1017/S0263574722000613 Published online by Cambridge University Press