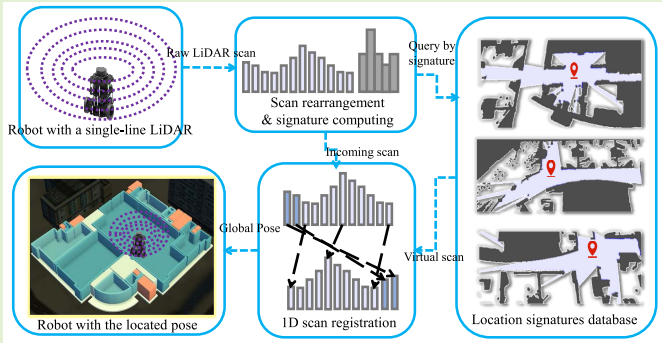
Global Localization With a Single-Line LiDAR by Dense 2D Signature and 1D Registration

Zhong Wang, Lin Zhang, *Senior Member, IEEE*, Shengjie Zhao, *Senior Member, IEEE*, and Shaoming Zhang, *Member, IEEE*

***Abstract*—Global localization is a key problem that needs to be solved for single-line LiDAR based robot navigation since it will directly affect the estimation accuracy of the robot's initial pose and the success rate of recovering the robot's state when it loses its local pose. Existing studies to deal with this problem usually extract feature points from laser beams and then resort to fast retrieval and registration methods to further determine the robot's pose. Although these methods have achieved good results in specific scenes, they often fail to perform well when the robot is far away from the map-building trajectory. It is therefore highly desirable to develop more robust techniques for this problem. In this work, we propose a novel solution which is based on *"Dense 2D Signature and 1D Registration"*, D2S1R for short. By establishing a dense signature database for 2D locations and combining with the fast retrieval technology, the 2D search space is extremely compressed. Furthermore, fast yaw angle determination is achieved by converting scan points to 1D space and measuring the difference of scan contours based on relative entropy. Experimental results on several complex indoor scenes show that D2S1R can complete global localization within 0.03*s* on an ordinary CPU in an area of nearly 4,000m^2. Besides, on the premise of achieving a location accuracy of 0.08 ± 0.04*m* and an orientation accuracy of 0.72 ± 0.60°, it can achieve an average success rate of 95% on all test datasets.**



***Index Terms*—Global localization, mobile robot navigation, single-line LiDAR, scan registration.**

I. Introduction

WHEN a robot is assigned a navigation task, the first problem to be solved is how to quickly and accurately determine its position and posture in the global map so as to further carry out path planning and motion control. When local localization fails or encounters the problem of robot kidnapping during navigation, it is necessary to perform global relocation to restore the state of the robot [1]–[3]. Therefore, global localization is a key link in the navigation technology stack of the mobile robot.

Manuscript received July 16, 2020; revised August 27, 2020; accepted August 29, 2020. Date of publication September 2, 2020; date of current version April 16, 2021. This work was supported in part by the National Natural Science Foundation of China under Grant 61973235 and Grant 61936014 and in part by the Natural Science Foundation of Shanghai under Grant 19ZR1461300. The associate editor coordinating the review of this article and approving it for publication was Prof. Wendy Flores-Fuentes. *(Corresponding author: Lin Zhang.)*

Zhong Wang, Lin Zhang, and Shengjie Zhao are with the School of Software Engineering, Tongji University, Shanghai 201804, China (e-mail: 2010194@tongji.edu.cn; cslinzhang@tongji.edu.cn; shengjiezhao@tongji.edu.cn).

Shaoming Zhang is with the School of Surveying and Geo-Informatics, Tongji University, Shanghai 201804, China (e-mail: zhangshaoming@tongji.edu.cn).

Digital Object Identifier 10.1109/JSEN.2020.3021049

Since global localization estimates the robot's pose in the global scope only through sensor information and a priori map without any position priori, a specific global localization algorithm is closely related to the sensors it uses. Up to now, LiDARs (light detection and ranging) and visual cameras are the two main types of sensors that have been widely studied and applied. Visual sensors have attracted extensive attention in the academic and industry in recent years due to their advantages of low price and abundant information [4]–[6]. However, although many researches have greatly advanced vision-based localization technologies, they are somewhat limited by their sensitivities to illumination, motion blur and high computational complexity. Another kind of sensors commonly used is LiDAR [7], which has the advantages of high sampling rate, high angular resolution, accurate distance measurement capability, adaptability to night environment, etc. Compared with cameras, it is a kind of more stable and reliable sensors. LiDARs have been used as the main sensors of localization algorithms by many autonomous navigation systems. In this work, a global localization solution will also be built based on a single-line LiDAR.

A common solution for global localization with single-line LiDARs is to make use of filters (such as Kalman filter and particle filter) to track the robot [8]–[11]. This kind

of approches recursively estimate the posterior distributions of the poses by fusing motion information and perception data, which is a "golden standard" for the problem for a long time. However, in practical applications, without setting proper initial poses, these methods require the robots to wander around to update the distributions, leading to long convergence time. Without moving the robots, another series of studies directly take the poses with the highest matching degrees as the results. Such schemes carry out global localization in two steps, place recognition and local registration [12]–[17]. Place recognition with 2D range data is mainly derived from loop closure detection approaches based on feature points in visual SLAM [18]–[20]. By establishing a feature-point database of key frames and resorting to efficient retrieval technology, a rough place is determined, and then the accurate pose of the robot is obtained by performing registration locally. For the feature-point databases are built from the map-building trajectories, these place-recognition-based schemes can achieve satisfied results when there are approximate records of the current poses in the databases. However, when the robots are far away from the map-building tracjectories, the detection rates of same feature points will be greatly reduced, leading to localization failures.

For the purpose of achieving a fast global localization without the need for robot movement and overcoming the drawback that the place-recognition-based strategies are easy to be affected by varying perspectives, this paper holds the view that it is necessary to design a more fine-grained place-recognition module and a more efficient registration component. Therefore, we try to search the best matching pose directly from the solution space, and propose a novel global localization scheme based on dense 2D signature and fast 1D registration, which is abbreviated as "D2S1R". Considering that the search space of location is much larger than the orientation, the pose search is carried out in two steps. First, a rotation invariant signature is assigned for each feasible location. Based on a fast retrieval strategy, the 2D search space is compressed to a few candidate locations, and then a novel 1D laser scan registration algorithm is proposed to determine the optimal parameters of the robot's pose.

We validated the performance of D2S1R on three public datasets and made a comprehensive comparison with the most widely used place-recognition-based counterparts. In addition, we also verified the practicability of the scheme in a real-world scene, and combined it with a well-known filter-based approach, AMCL (adaptive Monte Carlo localization) [9], to deal with the localization in ambiguous scenes. Experimental results show that it takes merely 0.03 seconds for D2S1R to complete one operation in a scene of nearly 4000 square meters. Moreover, on the premise of achieving a location accuracy of $0.08 \pm 0.04m$ and an orientation accuracy of $0.72 \pm 0.60°$, it achieved an average success rate of 95% on all test sets.

II. Related Work

As mentioned in Sect. I, the global localization approaches in the existing studies can be classified into two categories, filter-based ones and place-recognition-based ones. A brief review of the related work will be presented here.

A. Filter-Based Global Localization

Jensfelt *et al.* proposed MHL (multi-hypothesis localization) [8] to track the robot. It uses a Gaussian mixture model to construct the state distribution of the robot, in which every hypothesis is tracked by a corresponding extended Kalman filter. With the agent's motion and perception, its parameters are updated. Finally, the weighted distribution is taken as the belief of the state. Another typical filter-based global localization strategy is MCL (Monte Carlo localization) [9], relying on particle filter. At each update step, MCL estimates the posterior distribution of the robot pose by importance resampling. Dellaert *et al.* [10] were the first to use particle filter for mobile robot localization, and then AMCL [9] and MMCL (mixture Monte Carlo localization) [11] were introduced to improve its performance. MCL has the characteristics of multi-modal distribution modeling by means of particle representation. Compared with MHL, it has a stronger ability to express the pose's posterior distribution. Although the filter-based ones can solve the global localization problem in theory, they all require the robots to move randomly before they converge to the optimal poses, which is usually not allowed in the actual navigation applications.

B. Place-Recognition-Based Global Localization

The place-recognition-based schemes perform global localization by two steps of place recognition and local registration, and most of them focus on the former part. To perform place recognition, a line of researches designed hybrid priori map representations by making use of visual data and 3D data. In [12] and [13], while building 2D occupied maps, the image data of key frames were collected, and the rough spatial locations were determined by image retrieval technology. Park *et al.* [14] established a hybrid map representation by fusing topological graphs and local metric maps. Ma *et al.* [15] utilized a Hough transform to represent the priori map, and resorted to ICP to register 3D Kinect data. When there is only 2D range data provided, it becomes more challenging to design the algorithm. Tipaldi *et al*. proposed FLIRT (fast laser interest point transform) [21], a feature detector for 2D laser beams. With FLIRT, location identification can be accomplished by matching key points frame by frame. Later, in order to realize global localization, Tipaldi *et al*. designed GFP (geometrical FLIRT phrases) [16] based on FLIRT. They made use of information retrieval techniques to establish word order descriptions between feature points, achieving the goal of fast retrieval of target locations. Following a similar pipeline, Himstedt *et al*. proposed the GLARE (geometrical landmark relations) model [17], which encodes the relative relationship between different key points in a laser scan. Then, Himstedt *et al.* resorted to an ANN (approximate nearest neighbor) model to figure out location candidates and finally determined the global pose by a RANSAC based scan matching. For the reason that the design principles of GFP and GLARE are to encode the spatial locations by

constructing relationships between feature points, when the query frames and the records in the pre-established databases have approximate perspectives, they can achieve satisfied results. However, the relationship of features can be easily affected by the stability of the feature detection module, which may make the induced global localization schemes less robust.

Generally speaking, among the global localization schemes with 2D range data, the filter-based methods need the robots to take a long time to move randomly before they converge to the best poses. Although the place-recognition-based ones do not share the above-mentioned shortcoming of the filter-based ones, their performance are greatly restricted by the differences between the incoming scans and the key frames extracted from the map-building trajectories.

III. Methodology

A. Framework Overview

As shown in Fig. 1, the proposed global localization scheme D2S1R actually comprises two phases, an offline phase and an online phase. In the offline phase, the priori map $\mathcal{M}$ is patitioned into 2D grids according to a certain resolution. For each traversable grid, its corresponding location signature is generated. Then all resulting signatures are inserted into an ANN tree to obtain a location-related signature search tree, which is denoted by $\mathbb{T}$.

In the online phase, for each incoming laser scan, a signature of the current location is generated in the same way as the location's signature computed in the offline phase. Then the signature is used for retrieving the neighboring candidate locations from $\mathbb{T}$. After that, for each candidate location, a virtual scan is generated clockwise from $\mathcal{M}$, and is registered with the incoming scan. Finally, the location and orientation with the highest matching degree are determined as elements of the global pose.

B. 2D Search Space Compression

During the search of 2D pose parameters in global scope, the orientation space is fixed to 2π, while the location space will increase linearly with the scale of the map. Therefore, the first problem to be solved is to quickly compress the location search space. Inspired by place-recognition-based schemes, we believe that this goal can be achieved by designing recognizable signatures for 2D locations and combining fast retrieval technology. Our pipeline for 2D search space compression is presented in Algorithm 1, and detailed descriptions will be given in Sect. III-B.1 and Sect. III-B.2 respectively.

1) Location Signature: The signature of a 2D location should have the following characteristics:

- It should have a certain degree of location identification capability.
- It should be rotation invariant, so as to ensure that the same location can be retrieved under different heading angles.

In order to meet the aforementioned requirements, the scan's distance histogram is utilized as the signature to encode contour information of the 2D location. Although this signature is only a weak feature, our purpose of this step is to compress the 2D search space fast, and more accurate geometric consistency check can be accomplished in the second phase via scan registration.

Algorithm 1 Algorithm for $\mathbb{T}$'s Construction

Input:
The priori map, $\mathcal{M}$
Output:
The location-related ANN search tree, $\mathbb{T}$

Gridding $\mathcal{M}$ according to a certain resolution, get all passible grids $\mathbb{G}$
$\mathbb{T} \leftarrow \{\}$, set $\mathbb{T}$ as a new empty tree
Set the signature histogram's dimension as $\tilde{d}$
Set the sensor's maximum range as $\tilde{r}$
$\tilde{r}' \leftarrow \tilde{r} + \tilde{\epsilon}$
$\overline{r} \leftarrow \tilde{r}' \div \tilde{d}$
for all $\mathbf{g} \in \mathbb{G}$ **do**
  Scanning at $\mathbf{g}$ from $\mathcal{M}$ to get a virtual scan $\mathbb{S}_v$
  Assign $\tilde{r}'$ to all points that exceed $\tilde{r}$ in $\mathbb{S}_v$
  $\boldsymbol{s} \leftarrow [0, \ldots, 0]$, set siganture $\boldsymbol{s}$ as an array with size of $\tilde{d}$, each element with value of 0
  for all $\mathbf{p} \in \mathbb{S}_v$ **do**
    Get $\mathbf{p}$'s scan distance d
    $i \leftarrow \lfloor d \div \overline{r} \rfloor$
    $\boldsymbol{s}[i] \leftarrow \boldsymbol{s}[i] + 1$
  end for
  Normalize $\boldsymbol{s}$ to $\overline{\boldsymbol{s}}$
  Insert $\overline{\boldsymbol{s}}$ into $\mathbb{T}$
end for
return $\mathbb{T}$

Specifically, in the offline phase, at each passable location of $\mathcal{M}$, a virtual scan $\mathbb{S}_v$ is obtained from $\mathcal{M}$ by simulating the generation of the laser beam, and then its distance histogram is calculated as the signature of the location. In order to ensure that similar candidate locations can be retrieved stably, the following principles should be paid attention to when generating virtual scans. One is that the parameters of $\mathbb{S}_v$ should be consistent with the LiDAR. The other is that points that are out of sight range should also be encoded into signatures, because they also participate in the environmental characterization of the position. In our work, the way to deal with points outside the sight range is to give them a value slightly greater than the maximum sight range of the sensor. Finally, considering that the angular resolution of LiDAR may change, the location signatures need to be normalized uniformly.

Similarly, in the online phase, the computation of the signature from the incoming laser scan $\mathbb{S}_l$ should also follow the aforementioned principles to ensure that locations in $\mathcal{M}$ having similar appearances to the one where $\mathbb{S}_l$ is acquired are more probably retrieved.

2) Fast Candidate Locations Retrieval: By generating signatures for all traversable locations in $\mathcal{M}$, a spatial signature set is acquired. In order to quickly retrieve the neighboring candidate locations from it, this paper resorts to a high-dimensional data retrieval technology to organize these signatures.

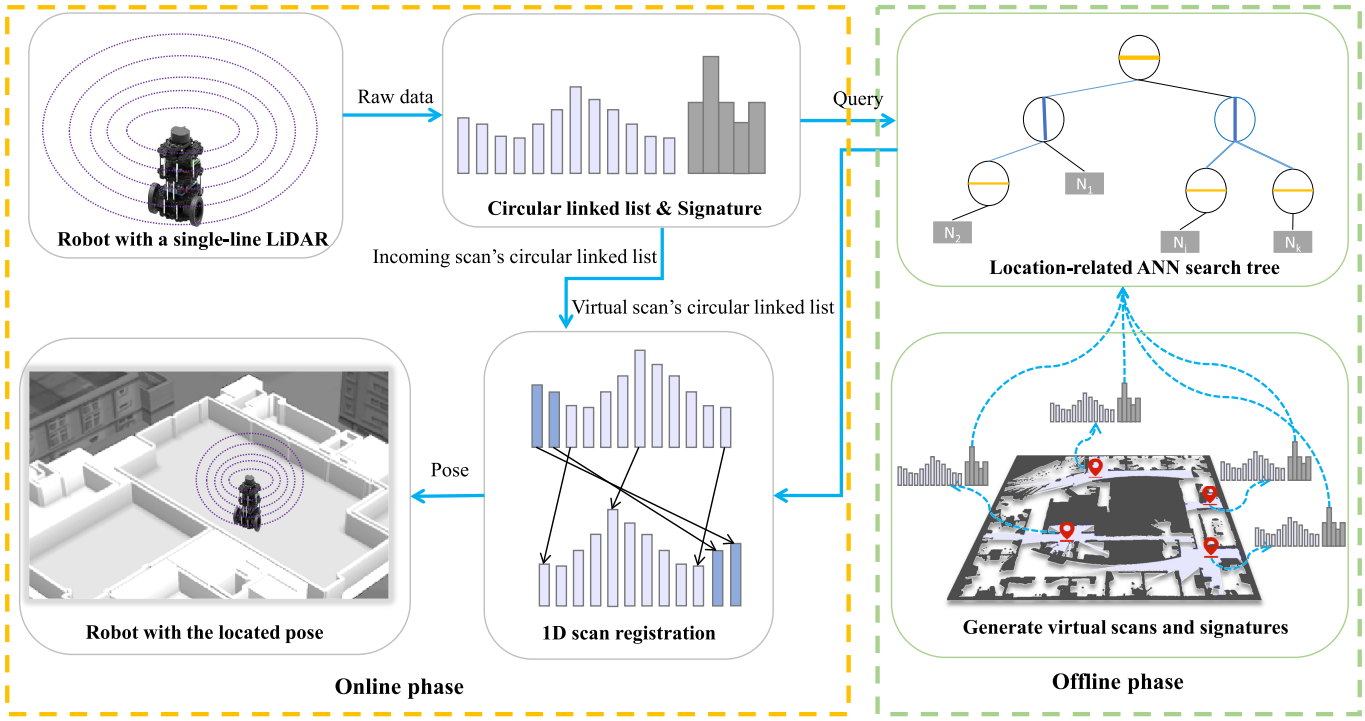


Fig. 1. Framework of our proposed global localization scheme D2S1R. The right part is the offline phase, generating location-related signatures from each free location of the priori map and inserting them into the location-related ANN search tree. The left part is the online phase, querying candidate locations according to the signature of the incoming scan, and then performing scan registration on each candidate location.

One common solution to search nearest neighbors in high dimensional data is KNN (k-nearest neighbors algorithm) [22], [23]. It is a non-parametric method in machine learning for classification and regression. However, computing exact nearest neighbors in high dimension is still a formidable task. A practical alternative is ANN, which computes the nearest neighbors in an approximation way. It has been shown that ANN is possible to achieve significantly faster processing speed with a relatively small error increment compared with KNN. Besides, ANN allows the user to specify a maximum approximation error bound, thus allowing he/she to control the trade-off between precision and running time [24].

To speed up online localization, a balanced ANN search tree of location signatures is pre-established in the offline phase. In the online phase, D2S1R only needs to load it from the hard disk into the memory, which takes negligible time. And the time complexity of querying from ANN has been proved to be a log constant of the tree size [24]. Therefore, the retrieval efficiency and stability of candidate locations can be guaranteed at this stage.

C. Yaw Angle Determination Based on Relative Entropy

Several possible 2D locations can be extracted by the procedure presented in Sect. III-B, but it not the end. Because the location's signature comes from statistics, there is a possibility that different locations have similar signatures. Therefore, it is necessary to further perform a geometric consistency check via scan registration.

Traditional scan registration approaches calculate the matching degree by sampling from the parameter space and converting the scan in the 2D space with the corresponding parameters. Although this line of schemes can efficiently determine the pose, we consider that the single-line LiDAR scan can be directly matched in 1D space to further improve the localization efficiency.

To speed up scan registration, an unconventional procedure to carry out it in 1D space is presented here. First, the incoming laser scan $\mathbb{S}_l$ is rearranged in the angular dimension and arranged into a circular linked list $\mathbf{l}_l$ in clockwise order. Similarly, for the virtual scan $\mathbb{S}_v$ of a candidate location, the corresponding circular linked list $\mathbf{l}_v$ is organized in the same way. Then, by moving the pointer of $\mathbb{S}_l$'s linked list forward step by step and calculating the matching degree of the two 1D vectors which are between the lists' head pointer and tail pointer, the best matching angle between $\mathbb{S}_l$ and $\mathbb{S}_v$ can be obtained. Finally, the above registration is repeated between $\mathbb{S}_l$ and each $\mathbb{S}_v$ of the candidate locations, and the location and orientation with the highest matching degree are taken as the final localization result.

Inevitably, the scanned data will contain a certain amount of noise. The commonly used distance metrics (such as l_1-norm, l_2-norm, etc.) are very sensitive to it. Therefore, in order to improve the robustness of registration, this work draws on the processing approach in information theory and makes use of relative entropy (also known as KL divergence) [25] to measure the similarity between two sequences. For two vectors to be matched, they are first treated with zero and infinity values to avoid seeking logarithmic overflow. Each zero value is set to a minimum value of $\tilde{\epsilon}$. For each infinity value, a $\tilde{r}'$ the same as the one set in offline phase is assigned to it. The

vectors are then normalized and resampled to a dimension of 360 with a resolution of 1 degree. The relative entropy of them is calculated as,

$$E(\boldsymbol{v}_l||\boldsymbol{v}_v)=\sum_{\theta=1^\circ}^{360^\circ}[\boldsymbol{v}_l(\theta)*\log_2\boldsymbol{v}_l(\theta)-\boldsymbol{v}_l(\theta)*\log_2\boldsymbol{v}_v(\theta)], \quad (1)$$

where $\boldsymbol{v}_l(\theta)$ and $\boldsymbol{v}_v(\theta)$ represent the values from $\boldsymbol{v}_l$ and $\boldsymbol{v}_v$ indexed by θ, respectively.

To sum up, the yaw angle determination procedure is formally defined in Algorithm 2. Although D2S1R is given by taking omnidirectional feild of view as an example, it should be noted that Algorithm 2 can still work under local perspectives such as 180 and 270 degrees. The corresponding change that need to be made is to reduce the local registration window accordingly. However, because location signatures will be influenced, therefore, it is necessary for the robot to rotate in situ to obtain a full circle of scanning.

Algorithm 2 Algorithm of Yaw Angle Determination

Input:
The incoming laser scan, $\mathbb{S}_l$
The virtual scan, $\mathbb{S}_v$
Output:
The yaw angle of $\mathbb{S}_l$, θ

Preprocess $\mathbb{S}_l$ and $\mathbb{S}_v$ to get $\mathbf{l}_l$ and $\mathbf{l}_v$
Get vector $\boldsymbol{v}_l$ from $\mathbf{l}_l$'s head pointer to tail pointer
$\theta \leftarrow 0^\circ$
$\hat{\theta} \leftarrow 0^\circ$
$\hat{e} \leftarrow$ inf
while $\hat{\theta} \neq 360^\circ$ **do**
  Step forward $\mathbf{l}_v$'s head pointer and tail pointer
  Get vector $\boldsymbol{v}_v$ from $\mathbf{l}_v$'s head pointer to tail pointer
  Preprocess $\boldsymbol{v}_l$ and $\boldsymbol{v}_v$
  Get relative entropy e between $\boldsymbol{v}_l$ and $\boldsymbol{v}_v$ by Eqa. 1
  if $e \leq \hat{e}$ **then**
    $\hat{e} \leftarrow e$
    $\theta \leftarrow \hat{\theta}$
  end if
  $\hat{\theta} \leftarrow \hat{\theta} + 1^\circ$
end while
return θ

IV. Experiments

A. Setup

As the result of global localization is relative to $\mathcal{M}$, the global poses of samples in the map were taken as the ground truth. For a specific dataset, the mapping pipeline introduced in paper [26] was adopted to build the corresponding priori map with a resolution of 0.05m. We obtained simulated LiDAR scanning data and poses with the help of Stage simulator [27]. Our D2S1R is implemented by C++ language, and we resorted to the ANN library implemented by Muja and Lowe [24] to achieve fast candidates retrieval. All experiments were carried out on an Intel core i7-8750h CPU, and all routines were single threaded.

TABLE I
Datasets Information

Name	Area (m^2)	Number of samples
D_{M_0}	3,853	4,876
D_{I_0}	885	16,465
D_{G_0}	3,644	4,285
D_{M_1}	3,853	3,697
D_{I_1}	885	5,418
D_{G_1}	3,644	9,828
D_T	1,140	1,429

1) Datasets: For examining the actual performance of D2S1R, three public datasets in real scenes were involved for validation, namely the dataset provided by Google corporation in [26], the dataset of campus buildings provided by MIT University in [28], and the dataset provided by Intel Research Lab in [29]. For ease of description, they were referred to as D_{G_0}, D_{M_0} and D_{I_0}, respectively.

In order to unify the data format, the Stage simulator [27] was utilized to resample the scan from the original data. Specifically, we took the corresponding priori map as the Stage's scene, in which a robot model equipped with a single-line LiDAR was placed. Then, the robot was controlled to move in the scene by receiving motion commands from a keyboard. While the robot was moving, the pose and scanning data acquired by the virtual LiDAR from the Stage simulator were recorded.

As discussed above, only verifying the samples on the map-building trajectory cannot thoroughly reflect the performance of the global localization algorithm. Therefore, for each dataset, Stage simulator was utilized to take another set of trajectory and corresponding scan data from each dataset's priori map as a new test set in the same way as the resampling step, which were respectively recorded as D_{G_1}, D_{M_1} and D_{I_1}.

In addition to the public datasets, we collected a dataset comprising data from an indoor scene of Tongji University for real-world experiments, which was named as D_T. Details of the all datasets are listed in Table I.

2) Evaluated Approaches and Metrics: In order to compare the performance of D2S1R with that of the place-recognition-based approaches, two typical global localization schemes based on FLIRT [21] were compared with. One is Sivic and Zisserman's strategy [30], and the other is Tipaldi *et al.*'s approach [16]. As mentioned earlier, the filter-based methods need the robot to move randomly to converge, so we can't compare D2S1R directly with them. Instead, we will have a qualitative comparative analysis with the most commonly used AMCL [9].

To evaluate the speed of D2S1R, the average 2D candidate locations retrieval time t_1, the average yaw angle determination time t_2, and its overall average localization time t_3 were timed. To compare the computational efficiency with place-recognition-based schemes, the average time consumption of Sivic and Zisserman's strategy [30] and Tipaldi *et al.*'s approach [16] were also timed, and recorded them as t_4 and t_5 respectively.

TABLE II
SETTINGS FOR HYPERPARAMETERS OF D2S1R

Parameter	$\tilde{r}$	$\tilde{d}$	$\tilde{\epsilon}$	$\tilde{k}$	$\tilde{\mu}_0$	$\tilde{\mu}_1$	$\tilde{\mu}_2$
Value	30m	60	10^{-3}	10	0.05m	0.05m	1°

TABLE III
TIME CONSUMPTIONS

Data \ **Time**	t_1	t_2	t_3	t_4	t_5
D_{M_0}	0.26	2.14	23.21	9.55	38.60
D_{I_0}	0.21	2.15	22.82	14.42	68.61
D_{G_0}	1.69	2.18	26.30	10.03	40.65

The success rate is also one of the indicators that global localization is concerned with. In this paper, samples with a location deviation of less than 0.2 meters and an angle deviation of less than 5 degrees were regarded as successfully located ones. The ratio of the number of the successful ones to the number of all test samples was taken as the localization success rate.

Localization precision is another indicator that needs attention. Understanding the localization capability and error range at different stages is helpful for parameters setting and improving the localization success rate. Therefore, this paper made a statistical analysis on indexes of average location error (δ_l), standard deviation of location error (σ_l), average orientation error (δ_o) and standard deviation of orientation error (σ_o).

It should be noted that the comparisons of all metrics were carried out under the optimal parameter combination of the corresponding algorithm. The hyperparameters settings of D2S1R are listed in Table II, where $\tilde{r}$ represents the range of the sensor, $\tilde{\epsilon}$ is the minimum value adopted, $\tilde{k}$ represents the number of candidate locations retained after compression of the 2D search space, $\tilde{d}$ is the dimension of the location signature, accordingly, $\tilde{\mu}_0$ represents its division bin size, $\tilde{\mu}_1$ represents the priori map resolution, and $\tilde{\mu}_2$ represents the angular resolution of the registration step. The parameter settings of the other two place-recognition-based schemes followed the configuration provided in the original papers [16], [30] and related open source codes [31], [32]. For AMCL, its parameter settings were kept the same with the default parameters recommended by the ROS navigation stack [33].

B. Results

1) Run Time: Comparing the t_3, t_4, and t_5 columns in Table III, what can be observed is that the overall localization time of D2S1R is on the same order as that of the place-recognition-based methods, and D2S1R is faster than Tipaldi *et al.*'s approach [16] on the three datasets of different sizes experimented. It can be observed that although D2S1R is based on dense location signatures, it is not inferior to the methods based on sparse feature points with respect to time efficiency. The reason is that the fast retrieval technique based on ANN greatly reduces the 2D search space. In addition, the scan registration in 1D space further accelerates the localization. Although D2S1R is slightly slower than Sivic and Zisserman's strategy [30], it will be seen later that D2S1R has higher success rate and stability of localization.

TABLE IV
SUCCESS RATES OF COMPETING METHODS ON DIFFERENT TRAJECTORIES

Method \ **Data**	D_{M_0}	D_{I_0}	D_{G_0}	D_{M_1}	D_{I_1}	D_{G_1}
Sivic&Zisserman [30]	84%	**99%**	**92%**	63%	82%	56%
Tipaldi *et al.* [16]	81%	98%	85%	62%	87%	37%
D2S1R	**94%**	97%	91%	**94%**	**93%**	**97%**

Comparing t_1, t_2, and t_3 from different datasets again, it can be seen that although D_{I_0}, D_{M_0} and D_{G_0} are nearly 5 times different in spatial range, there is no obvious difference in time consumption of the three in different stages. It can be speculated that within a certain range, the localization speed can be controlled at a constant time by D2S1R. In addition, the comparison shows that although scan registration and candidate location retrieval are on the same time level, scan registration with multiple candidate locations becomes the bottleneck of D2S1R's efficiency. And it is not difficult to know that scan registration can be performed in parallel, so relying on multi-core technology, the time consumption of D2S1R can be further reduced.

For comparing localization efficiency with filter-based AMCL, we randomly selected 20 different locations from D_T for experiments. At each location, AMCL was assigned an initial pose with deviation $(0.3m, 0.3m, 30°)$ from the ground truth. When the deviation between the localization result of AMCL and the ground truth was less than $0.2m$, and the angle deviation was less than 10°, AMCL was regarded as converged. Results showed that the convergence time of AMCL ranged from 1.99s to 10.99s, and the average convergence time was 4.39s, which were much longer than place-recognition-based approaches and our D2S1R.

2) Success Rate: From the results in Table IV, it can be observed that the success rate of D2S1R is over 91% on both the map-building and non-map-building trajectories. By contrast, Sivic and Zisserman's strategy [30] and Tipaldi *et al.*'s approach [16] can achieve pleasing success rates only on the map-building trajectories, but their localization performance is greatly reduced when deviating from the map-building trajectory.

One noticeable phenomenon in Table IV is that the place-recognition-based schemes have achieved good results on both types of trajectories of the Intel dataset (D_{I_0} and D_{I_1}). Table I shows that the ratio of the number of samples to map area of D_{I_0} is much larger than that of the other two datasets. It can be inferred that the spatial range of the Intel dataset is relatively small, and the map-building trajectory has enough scan data at any local location, thus improving the localization result. Therefore, one possible way to improve the global localization capability of place-recognition-based methods is to spread the mapping trajectory throughout the mapping space as much as

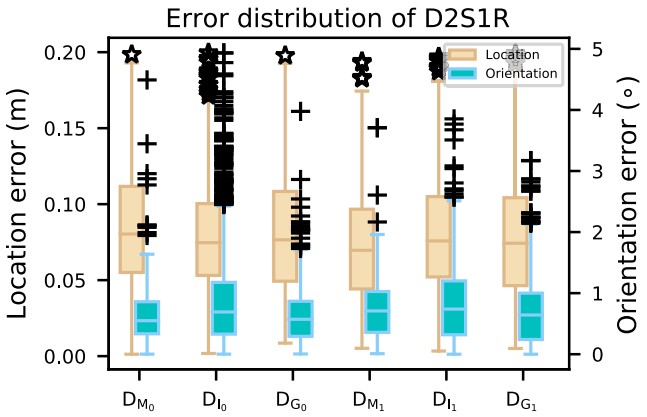


Fig. 2. Localization error distribution of D2S1R. The left axis indicates location error. The right axis indicates orientation error. The black dots indicate outlier samples.

TABLE V
LOCALIZATION PRECISION OF D2S1R

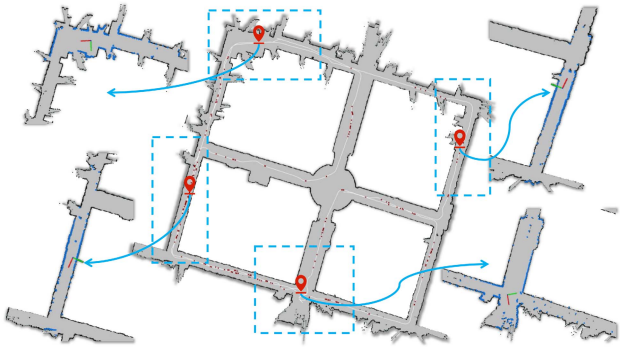
Data \ Metric	$\delta_l(m)$	$\delta_o(^\circ)$	$\sigma_l(m)$	$\sigma_o(^\circ)$
D_{M_0}	0.08	0.61	0.04	0.45
D_{I_0}	0.08	0.89	0.03	0.87
D_{G_0}	0.09	0.63	0.04	0.47
D_{M_1}	0.07	0.71	0.04	0.44
D_{I_1}	0.09	0.81	0.04	0.70
D_{G_1}	0.09	0.71	0.04	0.65
All	0.08	0.72	0.04	0.60

possible. However, this needs to be done at the cost of reducing the time efficiency, which is illustrated in Table III that t_5 of D_{I_0} is much longer than the other two datasets.

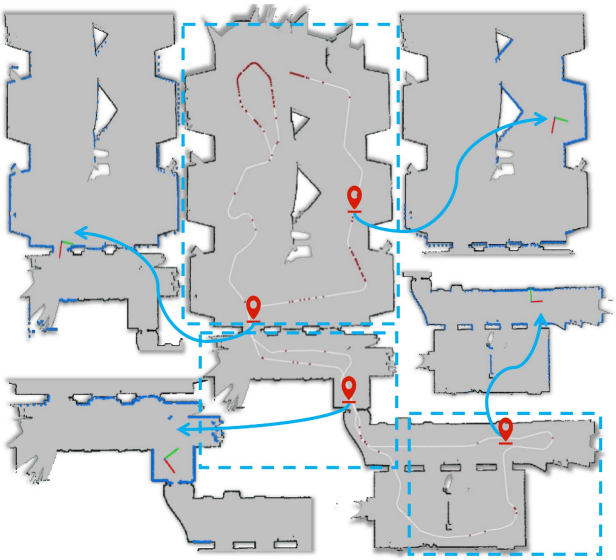
3) Precision: The final localization deviation and fluctuation of D2S1R on different datasets are presented in Table V. The statistics manifest that D2S1R achieves very high localization precision on all datasets, with the average precision of the location being about 0.08 ± 0.04m and the average precision of the orientation being about $0.72 \pm 0.60^\circ$.

The box plot of the error distribution is shown in Fig. 2, from which it can be seen that the localization fluctuation of D2S1R on Intel dataset is larger than that of the other two datasets. The possible reason of this is that the space of this dataset is relatively dense, and the generation of location signature requires higher distance resolution. Otherwise, it is easy to cause that the optimal location does not have sufficient discrimination against the locations nearby, and eventually more outliers appear.

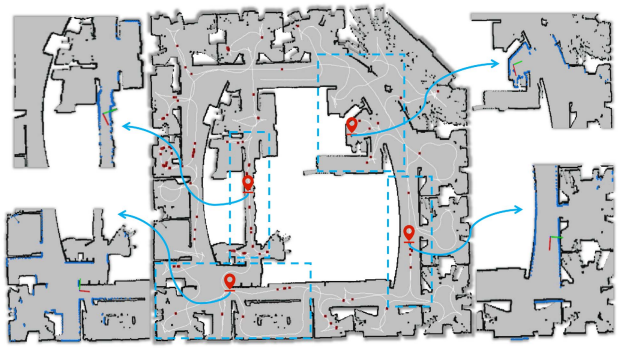
Although the global localization precision of D2S1R is accurate enough for most indoor navigation applications, it is worth noting that the partition resolution of $\mathcal{M}$ in offline phase and the scan registration resolution in online phase have a direct impact on the final localization precision. For this reason, corresponding adjustments should be made according to maps and sensors in practical applications. Besides, on the basis of D2S1R, a more fine-grained local registration can further improve the localization precision.



(a) Matched examples of D_{M_1}



(b) Matched examples of D_{G_1}



(c) Matched examples of D_{I_1}

Fig. 3. Matched examples. In (a), (b), and (c), the white curves are the test trajectories, the dark red points represent the failed samples, and the rest are the successful samples. The blue point clouds in the partially enlarged figures represent the projected scan after the transformation of the localization results.

4) Matched Examples and Failure Cases: The matched examples of different spatial positions in three actual scenarios are shown in Fig. 3(a), Fig. 3(b) and Fig. 3(c), respectively.

It can be discerned from the distribution of successfully located samples in Fig. 3 that D2S1R has high success rate and high accuracy in narrow area, wide space, structured space, irregular area and noise environment, which shows its robustness and practicability in real scenes.

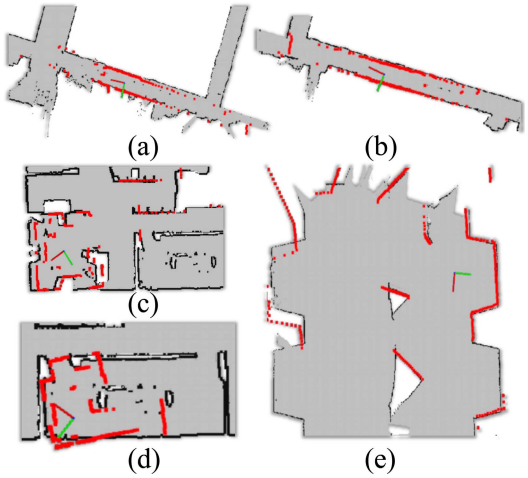


Fig. 4. Typical failure cases. Location features along long corridors ((a) and (b)) have low discrimination. Narrow areas (c) require higher resolution for distance histogram partition. High repetition pattern in office areas (d) causes candidate location deviation. Priori map errors (e) bring matching offset. The red points represent the scan converted by the localization result.

Although D2S1R has achieved remarkable results as a whole, there are still some local failure samples. Several typical failure cases are shown in Fig. 4. By analyzing the samples failed to be located, several typical situations are summarized here:

- Narrow long corridor (Fig. 4(a) and Fig. 4(b)). In such a scene, all locations along the corridor have high matching degrees, which is a typical repetitive pattern problem in localization. Due to the limitation of the measurement range, this problem cannot be completely solved by a single short-range LiDAR, and it is necessary to incorporate other sensors or resort to long-range LiDARs.
- Candidate location deviation (Fig. 4(c)). One of the possible reasons for this situation is that the partition resolution of the distance histogram is too large, resulting in low degree of discrimination of the location against adjacent locations.
- Office area repetition pattern (Fig. 4(d)). It is another typical repetition pattern scenario, i.e. a similar office area appears, resulting in the best candidate location not being within the retrieval range. This can be solved to some extent by increasing the number of candidate locations.
- Priori map error (Fig. 4(e)). The localization result of this situation appears as the local deviation of matching degree, which is caused by the lack of local scanning data. To avoid such errors, when constructing a priori map, someone should expand the whole space as much as possible and add enough loops to improve the accuracy of map construction.

5) Real-World Experiment: Using the robot platform TURTLEBOT2 as shown in Fig. 5, we collected a real-world dataset with a LEISHEN N30103A single-line LiDAR, and established the priori map of the scene as shown in Fig. 6. According to the experimental results, D2S1R achieved a success rate of 89% in this scenario, which fully demonstrated its practicality. The samples failed to be located mainly concentrate near the starting point and in the long corridor. The failures in the long corridor are due to the influence of repetition patterns as discussed in Sect. IV-B.4, while the failures near the starting point can be attributed to the existence of a large area of glass wall.

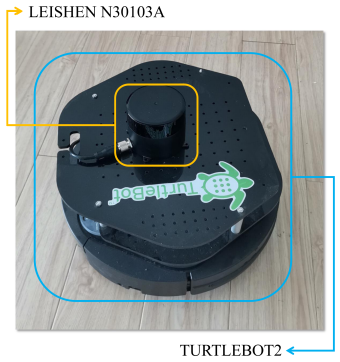


Fig. 5. Real-world experimental platform. Our experimental platform is composed of a TURTLEBOT2 robot platform and a LEISHEN N30103A single-line LiDAR with 360° viewing angle and 30*m* measuring range.

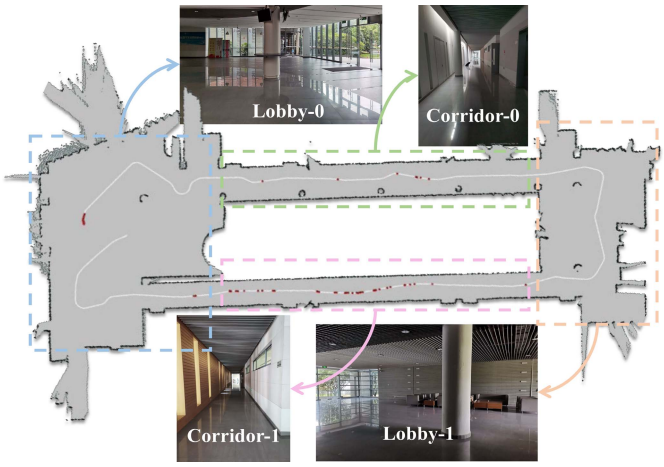


Fig. 6. Real-world experimental result. The white curve is the trajectory of the robot when collecting scanning data, and the crimson points on it indicate the locations where D2S1R failed to locate.

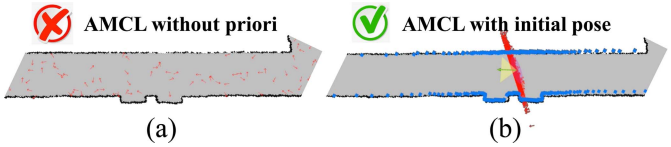


Fig. 7. Green arrow represents the current pose. Blue points represent the transformed scan. Red points represent AMCL's particles. Localization fails without any priori information (a). When the initial pose is specified, AMCL can estimate an appropriate pose, but its particles are not convergent well (b).

6) Localization in Ambiguous Scenes: One of the purposes of designing D2S1R is to achieve global localization without the need for robot movement. However, it is difficult to deal with the localization problem in ambiguous scenes only by the one-shot scanning of a fixed position. Therefore, we will discuss the treatment scheme in this special case.

Filter-based AMCL has the characteristics of multi-modal modeling, and is a common method for global localization in ambiguous scenes. However, it is difficult for AMCL

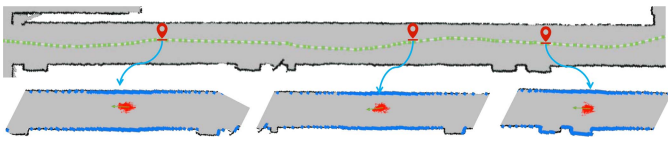


Fig. 8. Localization results in an ambiguous scene by the fused model of D2S1R and AMCL. The green dots on the white trajectory represent the localization results of the fused model. When the results of D2S1R are continuously fed into AMCL as a pseudo odometer, the particles of AMCL can converge well, and the localization success rate is greatly improved in the long corridor.

to converge without a suitable initialization (as shown in Fig. 7(a)). When the localization result of D2S1R is given to AMCL as the initial pose, it can be seen that AMCL can estimate an appropriate pose with probability, although its particles are not completely convergent at this time (as shown in Fig. 7(b)). To further make the particles converge, it is necessary for the robot to move to update its motion and observation. However, when only a single-line LiDAR is equipped, only the observation update is available. Therefore, we consider transforming D2S1R's outputs as a continuous pseudo odometer to feed into AMCL. From the experimental results in an ambiguous scene (as shown in Fig. 8), it can be observed that although AMCL or D2S1R can't individually solve the global localization problem well in this environment, the fused model of the two gives satisfactory localization results. In the experiment of the long corridor, all samples were successfully located, which benefits from the seamless combination of the advantages of D2S1R and AMCL.

V. Conclusion

In this work, aiming at solving the problem of fast global localization relying on a single-line LiDAR in large indoor scenes, a novel localization solution based on dense location signatures and fast scan registration was proposed. In order to overcome the problem that the previous place-recognition-based methods using sparse feature points are greatly affected by varying perspective, a location-related signature database is densely established, and the fast 2D search space compression is achieved by an efficient retrieval technique. Then, the incoming 2D laser scan is transformed back to 1D space. At the same time of rapid geometric verification, the orientation of the scan is determined. The experimental results on different trajectories of actual scenes corroborate that D2S1R has achieved an average success rate of 95%, and has achieved a location precision of 0.08 ± 0.04m and an orientation precision of $0.72 \pm 0.60°$. Besides, in a space of nearly 4,000 square meters, it only takes 0.03 seconds to determine the global pose on a common CPU. For most indoor navigation applications, the method in this paper can meet the actual needs in accuracy, speed and success rate. Furthermore, D2S1R can be seamlessly fused with AMCL to improve its performance in ambiguous scenes.

Although D2S1R solves the global localization problem with a single-line LiDAR to some extent, the sparsity of the single-line sensor data still brings many limitations and difficulties to the algorithm design. To further improve its performance may require more advanced or more kinds of sensors.

References

[1] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, 2nd ed. Cambridge, MA, USA: MIT Press, 2005.

[2] K. L. Ho and P. Newman, "Loop closure detection in SLAM by combining visual and spatial appearance," *Robot. Auto. Syst.*, vol. 54, no. 9, pp. 740–749, Sep. 2006.

[3] B. Williams, M. Cummins, J. Neira, P. Newman, I. Reid, and J. Tardós, "A comparison of loop closing techniques in monocular SLAM," *Robot. Auto. Syst.*, vol. 57, no. 12, pp. 1188–1197, Dec. 2009.

[4] Y. Ling and S. Shen, "Real time dense mapping for online processing and navigation," *J. Field Robot.*, vol. 36, no. 5, pp. 1004–1036, Mar. 2019.

[5] M. Ivanov *et al.*, "Influence of data clouds fusion from 3D real-time vision system on robotic group dead reckoning in unknown terrain," *IEEE/CAA J. Automatica Sinica*, vol. 7, no. 2, pp. 368–385, Mar. 2020.

[6] O. Y. Sergiyenko *et al.*, "Data transferring model determination in robotic group," *Robot. Auto. Syst.*, vol. 83, pp. 251–260, Sep. 2016.

[7] K. Khoshelham and S. Zlatanova, "Sensors for indoor mapping and navigation," *Sensors*, vol. 16, no. 5, p. 655, May 2016.

[8] P. Jensfelt and S. Kristensen, "Active global localization for a mobile robot using multiple hypothesis tracking," *IEEE Trans. Robot. Autom.*, vol. 17, no. 5, pp. 748–760, Oct. 2001.

[9] S. Thrun, D. Fox, W. Burgard, and F. Dallaert, "Robust Monte Carlo localization for mobile robots," *Artif. Intell.*, vol. 128, nos. 1–2, p. 99–141, May 2001.

[10] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte Carlo localization for mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom.*, Detroit, MD, USA, May 1999, pp. 1322–1328.

[11] D. Fox, "Adapting the sample size in particle filters through KLD-sampling," *Int. J. Robot. Res.*, vol. 22, no. 12, pp. 985–1003, Dec. 2003.

[12] Z. Su, X. Zhou, T. Cheng, H. Zhang, B. Xu, and W. Chen, "Global localization of a mobile robot using lidar and visual features," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Macau, China, Dec. 2017, pp. 2377–2383.

[13] X. Zhou, Z. Su, D. Huang, H. Zhang, T. Cheng, and J. Wu, "Robust global localization by using global visual features and range finders data," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Kuala Lumpur, Malaysia, Dec. 2018, pp. 218–223.

[14] J.-H. Park, S. Kim, N. lett Doh, and S.-K. Park, "Vision-based global localization based on a hybrid map representation," in *Proc. Int. Conf. Control, Autom. Syst.*, Seoul South Korea, Oct. 2008, pp. 1104–1108.

[15] Y. Ma, Y. Guo, M. Lu, J. Zhao, and J. Zhang, "Global localization in 3D maps for structured environment," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Beijing, China, Jul. 2016, pp. 6680–6683.

[16] G. D. Tipaldi, L. Spinello, and W. Burgard, "Geometrical FLIRT phrases for large scale place recognition in 2D range data," in *Proc. IEEE Int. Conf. Robot. Autom.*, Karlsruhe, Germany, May 2013, pp. 2693–2698.

[17] M. Himstedt, J. Frost, S. Hellbach, H.-J. Bohme, and E. Maehle, "Large scale place recognition in 2D LIDAR scans using geometrical landmark relations," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Hong Kong, Sep. 2014, pp. 5030–5035.

[18] A. Angeli, D. Filliat, S. Doncieux, and J.-A. Meyer, "Fast and incremental method for loop-closure detection using bags of visual words," *IEEE Trans. Robot.*, vol. 24, no. 5, pp. 1027–1037, Oct. 2008.

[19] A. Angeli, S. Doncieux, J.-A. Meyer, and D. Filliat, "Real-time visual loop-closure detection," in *Proc. IEEE Int. Conf. Robot. Autom.*, Pasadena, MA, USA, May 2008, pp. 1842–1847.

[20] M. Labbe and F. Michaud, "Online global loop closure detection for large-scale multi-session graph-based SLAM," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Pasadena, MA, USA, Sep. 2014, pp. 2661–2666.

[21] G. D. Tipaldi and K. O. Arras, "FLIRT—Interest regions for 2D range data," in *Proc. IEEE Int. Conf. Robot. Autom.*, Anchorage, AK, USA, May 2010, pp. 3616–3622.

[22] K. Q. Weinberger, J. Blitzer, and L. K. Saul, "Distance metric learning for large margin nearest neighbor classification," in *Proc. Adv. Neural Inf. Process. Syst.*, Vancouver, BC, Canada, Dec. 2006, pp. 1473–1480.

[23] K. Fukunaga and P. M. Narendra, "A branch and bound algorithm for computing k-nearest neighbors," *IEEE Trans. Comput.*, vols. C–24, no. 7, pp. 750–753, Jul. 1975.

[24] M. Muja and D. G. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *Proc. Int. Conf. Comput. Vis. Theory Appl.*, Lisboa, Potugal, Feb. 2009, p. 5.

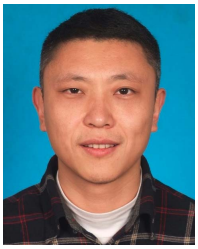
[25] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, 1951.

[26] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2D LIDAR SLAM," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, Stockholm, Sweden, May 2016, pp. 1271–1278.
[27] R. Vaughan, "Massively multi-robot simulation in stage," *Swarm Intell.*, vol. 2, nos. 2–4, pp. 189–208, Aug. 2008.
[28] M. Fallon, H. Johannsson, M. Kaess, and J. J. Leonard, "The MIT stata center dataset," *Int. J. Rob. Res.*, vol. 32, no. 14, pp. 1695–1699, Dec. 2013.
[29] W. Burgard *et al.*, "A comparison of SLAM algorithms based on a graph of relations," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Louis, ON, USA, Oct. 2009, pp. 2089–2095.
[30] Sivic and Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, Nice, France, Oct. 2003, p. 140.
[31] G. D. Tipaldi and K. O. Arras. *Flirtlib*. Accessed: Mar. 21, 2017. [Online]. Available: https://github.com/tipaldi/flirtlib.git
[32] G. D. Tipaldi. *Gflip*. Accessed: Oct. 19, 2015. [Online]. Available: https://github.com/tipaldi/gflip
[33] D. V. Lu, M. Ferguson, and A. Hoy. *Amcl*. Accessed: Sep. 27, 2013. [Online]. Available: http://wiki.ros.org/amcl

Zhong Wang received the B.S. and M.S. degrees from the School of Surveying and Geo-Informatics, Tongji University, Shanghai, China, in 2016 and 2019, respectively, where he is currently pursuing the Ph.D. degree with the School of Software Engineering. His research interests include motion planning of mobile robot, SLAM, and computer vision.

Lin Zhang (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees from the Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China, in 2003 and 2006, respectively, and the Ph.D. degree from the Department of Computing, The Hong Kong Polytechnic University, Hong Kong, in 2011. From March 2011 to August 2011, he was a Research Associate with the Department of Computing, The Hong Kong Polytechnic University. In August 2011, he joined the School of Software Engineering, Tongji University, Shanghai, China, where he is currently a Full Professor. His current research interests include environment perception of intelligent vehicle, pattern recognition, computer vision, and perceptual image/video quality assessment.

Shengjie Zhao (Senior Member, IEEE) received the B.S. degree in electrical engineering from the University of Science and Technology of China, Hefei, China, in 1988, the M.S. degree in electrical and computer engineering from the China Aerospace Institute, Beijing, China, in 1991, and the Ph.D. degree in electrical and computer engineering from Texas A&M University, College Station, TX, USA, in 2004. He is a Professor with the School of Software Engineering, Tongji University, Shanghai, China. In previous postings, he conducted research at Lucent Technologies, Whippany, NJ, USA, and China Aerospace Science and Industry Corporation, Beijing. His research interests include big data, wireless communications, image processing, and signal processing. He is a Fellow of the Thousand Talents Program of China.



Shaoming Zhang (Member, IEEE) received the B.S. degree from the School of Electronic Information Engineering, Tianjin University, Tianjin, China, in 2002, the M.S. degree from the 14th Institute of Electronics, Ministry of Information Industry, Nanjing, China, in 2005, and the Ph.D. degree from the School of Surveying and Geo-Informatics, Tongji University, Shanghai, China, in 2008. In June 2008, he joined the School of Surveying and Geo-Informatics, Tongji University, where he is currently an Associate Professor. From January 2011 to January 2012, he worked as a Postdoctoral Researcher with The Ohio State University. His current research interests include environment perception of intelligent vehicle, pattern recognition, computer vision, and robotics.