Fast Sequence-Matching Enhanced Viewpoint-Invariant 3-D Place Recognition

Peng Yin, Fuying Wang, Anton Egorov, Jiafan Hou, Zhenzhong Jia, *Member, IEEE*, and Jianda Han, *Member, IEEE*

***Abstract*—Recognizing the same place undervariant viewpoint differences is the fundamental capability for human beings and animals. However, such a strong place recognition ability in robotics is still an unsolved problem. Extracting local invariant descriptors from the same place under various viewpoint differences is difficult. This article seeks to provide robots with a human-like place recognition ability using a new 3-D feature learning method. This article proposes a novel lightweight 3-D place recognition and fast sequence matching to achieve robust 3-D place recognition, capable of recognizing places from a previous trajectory regardless of viewpoints and temporary observation differences. Specifically, we extracted the viewpoint-invariant place feature from 2-D spherical perspectives by leveraging spherical harmonics' orientation-equivalent property. To improve sequence-matching efficiency, we designed a coarse-to-fine fast sequence-matching mechanism to balance the matching efficiency and accuracy. Despite the apparent simplicity, our proposed approach outperforms the relative state of the art. In both public and self-gathered datasets with orientation/translation differences or noise observations, our method can achieve above 95% average recall for the best match with only 18% inference time of PointNet-based place recognition methods.**

***Index Terms*—Sequence matching, simultaneous localization and mapping (SLAM), spherical harmonics, viewpoint invariant, 3-D place recognition.**

Manuscript received May 10, 2020; revised May 14, 2020 and December 6, 2020; accepted January 12, 2021. Date of publication February 9, 2021; date of current version October 27, 2021. *(Corresponding author: Peng Yin.)*

Peng Yin is with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA 15213 USA (e-mail: pyin2@andrew.cmu.edu).

Fuying Wang is with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mail: thuwfy15@gmail.com).

Anton Egorov is with the Skolkovo Institute of Science and Technology, Moscow 121205, Russia. He is now with the Autonomous Transportation Systems Laboratory, Innopolis University, Innopolis 420500, Russia (e-mail: anton.egorov@skoltech.ru).

Jiafan Hou is with the School of Science and Engineering, Chinese University of Hong Kong, Shenzhen, Shenzhen 518172, China (e-mail: 116010072@link.cuhk.edu.cn).

Zhenzhong Jia is with the Southern University of Science and Technology, Shenzhen 518172, China (e-mail: jiazz@sustech.edu.cn).

Jianda Han is with the Nankai University, Tianjin 300071, China (e-mail: hanjianda@nankai.edu.cn).

Color versions of one or more figures in this article are available at https://doi.org/10.1109/TIE.2021.3057025.

Digital Object Identifier 10.1109/TIE.2021.3057025

I. Introduction

PLACE recognition plays an essential role in mobile robotics and has been well studied over the past two decades. The capability to relocalize visited areas has enabled multiple applications, such as autonomous vehicles, warehouse automation, rescue-, service-, and delivery robotics, etc. Vision-based place recognition methods [1] usually suffer from illumination variations, while the 3-D LiDAR inputs do not have such issue. The price decline and accurate measurements of LiDAR devices make 3-D point cloud widely applied in simultaneous localization and mapping (SLAM) and navigation tasks. However, 3-D place recognition in the same area under various viewpoints and dynamic scenarios is still a very challenging task. Traditional place recognition methods are mainly based on 3-D registration algorithms [2] or handcraft 3-D feature descriptor [3]–[5]. Achieving efficient place recognition with registration-based methods is difficult in practice since they usually require good initial estimation [2]. Three-dimensional handcraft features can be viewpoint invariant, such as 3-D scale-invariant feature transform [3] and Spin-Image [4], while extracting such features in the real applications is time-consuming [6].

Recent studies on PointNet-based [7] 3-D data association have brought light to the LiDAR-based place recognition task [8]–[10]. These approaches extract place descriptors from the raw point cloud in an end-to-end learning manner and have achieved remarkable performance on public datasets. However, most learning-based 3-D place descriptors are sensitive to viewpoint changes. Furthermore, their dependence on the single observation usually fails to guarantee a correct potential match because the sensor information always contains measurement noises.

To achieve viewpoint-invariant 3-D place recognition while balancing the matching accuracy and searching efficiency simultaneously, we propose a 3-D place recognition framework. As depicted in Fig. 1, our method mainly includes two modules: spherical harmonic place descriptor extraction (SphereVLAD), and fast sequence matching (Fast-Matching). SphereVLAD is an viewpoint-invariant descriptor extraction module, which leverages the orientation-equivalent property of spherical harmonics. It can provide place descriptors with a sequence of spherical projections. Compared with raw 3-D point cloud data, spherical projections can capture sufficient geometric structure for recognition in complex 3-D environment and have an intrinsic advantage in orientation equivalence. Our matching results are conducted on the sequence observation; instead of time-consuming

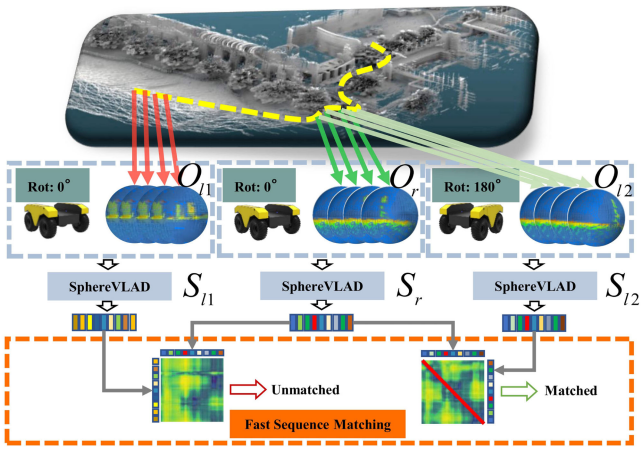

Fig. 1. Given two local 3-D sequences O_{l1} and O_{l2} and a reference sequence O_r, which are observed with different orientations or positions, our method can extract out viewpoint-invariant place descriptors S_{l1}, S_{l2}, and S_r, respectively. Without any initial estimation, we can efficiently detect the feature similarity via our fast sequence-matching procedure.

brute-force searching in the traditional sequence matching [11], [12], Fast-Matching can speed up the matching procedure by 30–50 times.

We conduct an extensive experimental analysis to evaluate the proposed method on both public datasets [13], [14] and self-gathered datasets. Notably, experiment results show that our method is more robust against viewpoint difference than the relative state of the arts [8]–[10], [15], [16]. Additionally, our method consumes less GPU memory and can extract the local place descriptor within 30 ms, making it more suitable for large-scale place recognition and SLAM applications.

II. Related Work

This section will mainly focus on related works of LiDAR-based 3-D place recognition and recent developments in place-matching approaches.

A. Three-Dimensional Place Recognition

Recent 3-D place recognition approaches [8], [9] have made significant progress. Uy and Lee [8] combined the feature extraction ability of PointNet [7] to obtain translation-invariant 3-D place descriptors. Thus, PointNetVLAD [8] has less limitation to the optimal local problem in traditional alignment-based approaches [2]. Based on Uy and Lee's work, LPDNet [9] further improved place recognition accuracy by combining with PointNet++ [17], which is designed to capture more geometric features from raw point clouds. SeqLPD [18] obtains an improvement by incorporating the LPDNet [9] and the sequence-matching module. Recently, PCAN [10] has improved the feature aggregation ability by applying an attention VLAD layer to mark out the essential points in the 3-D point cloud. However, all the above methods are sensitive to viewpoints changes, since PointNet approaches [7], [17] are not designed to be viewpoint invariant.

Kim and Kim [19] proposed a projection-based descriptor called Scan-Context to solve long-term global localization. Yin *et al.* [15] proposed a viewpoint-invariant descriptor from the projections and combined with Monte Carlo localization to achieve a fast global localization. Most recently, Chen *et al.* [16] have introduced an overlapping estimation network to predict the place feature difference and the relative yaw differences simultaneously. However, the viewpoint-invariant ability in the above projection-based methods is rusticity to yaw and nontranslation differences. In the real applications, such as unstructured road status (with changing pitch/roll in viewpoints) and large-scale 3-D environments (with local translation differences on XY plane), such constraints cannot always be satisfied.

Different from the above point-based or projection-based methods, we inference the viewpoint-invariant place descriptors from the spherical harmonic domain [20], which is robust to both 3-D orientations and local translations.

B. Sequence Matching

Traditional place recognition methods usually rely on bag of visual words [21] to encode place descriptors into a tree-like structure and retrieve similar places with one single scan. FABMAP [22] uses a Bayesian filtering approach to achieve long-term place recognition over a 1000-km trajectory with one single scan. Since a single scan usually contains measurement noise and observable texture difference caused by spatial/dynamic scenery differences, SeqSLAM [11] uses a brute-force sequence-matching manner improve the place-matching accuracy. However, brute-force searching is time-consuming in practice. These methods cannot be directly applied to place recognition tasks. Siam and Zhang [23] proposed a Fast-SeqSLAM method, which improved the searching efficiency by utilizing an approximate nearest neighbor (ANN) as the initial estimate for potential matches. Since the ANN in Fast-SeqSLAM still relies on single image feature similarities, the initial search efficiency may decrease when the number of reference sequences is beyond specific amounts.

Our proposed framework balances the recognition efficiency and accuracy by leveraging the sequence matching with a coarse-to-fine searching manner.

III. Our Method

In this section, we will introduce the details of our framework. Given the local and global reference sequences of LiDAR scans, we first generate the multilayer spherical projections, which are then encoded as viewpoint-invariant place descriptors by our SphereVLAD module; finally, we locate the best matches based on our Fast-Matching module. We will investigate the three modules, respectively.

A. Multilayer Spherical Generation

To apply the feature extraction in the SphereVLAD, we need first transform 3-D point clouds into spherical representations. Esteves *et al.* [20] proposed a ray–mash interaction method to project 3-D points onto one spherical mesh. However, this

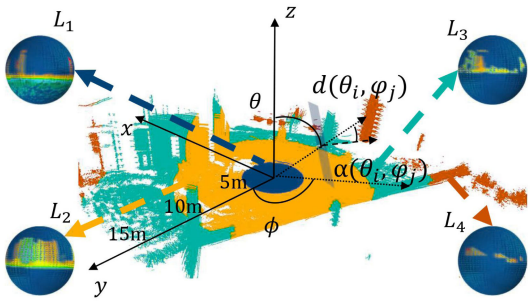

Fig. 2. Multilayer spherical view generation. Given a local point cloud, we project points of different ranges ([0, 5],[5, 10], [10, 15, [15, 20] m) to corresponding spherical views (L_1, L_2, L_3, L_4). Each layer includes two channels, nearest point distance d_{θ_i,ϕ_j}, and direction angle α_{θ_i,ϕ_j} on grid (θ_i, ϕ_j).

projection is unsuitable for naturally dense point clouds, since there may exist several points within one grid on the spherical mesh. To mitigate this problem, we design a multilayer spherical view generation mechanism. As shown in Fig. 2, we divide the raw point cloud into different ranges ([0, 5, 10, 15, 20] m), and each layer projects one range of 3-D data onto a spherical view. Given a desired resolution H, we generate an $H \times H$ grids from the center on the spherical view. On the grid (θ_i, ϕ_j), we set d_{θ_i,ϕ_j} as distance to nearest points within this grid. We also compute the angle α_{θ_i,ϕ_j} between the ray and the surface normal at the intersecting face. In our applications, $H = 64$; this parameter is selected by evaluating the performance and efficiency on different datasets.

B. SphereVLAD

1) Viewpoint-Invariant Feature Extraction: SphereVLAD first extracts the orientation-equivalent features with the spherical convolution operation. Given $g, \psi \in \mathbf{SO}(3) \to \mathbb{R}^K$ on the rotation group, spherical convolution between g and ψ is defined as

$$[g \star_{\mathbf{SO}(3)} \psi](\mathbf{R}) = \int_{\mathbf{SO}(3)} g(\mathbf{R}^{-1}\mathbf{Q})\psi(\mathbf{Q})d\mathbf{Q} \qquad (1)$$

where $\mathbf{R}, \mathbf{Q} \in \mathbf{SO}(3)$. Based on the proof in [24], spherical convolution is shown to be orientation-equivariant:

$$[g \star_{\mathbf{SO}(3)} [L_{\mathbf{G}}\psi]](\mathbf{R}) = [L_{\mathbf{G}}[g \star_{\mathbf{SO}(3)} \psi]](\mathbf{R}) \qquad (2)$$

where $L_{\mathbf{G}}(\mathbf{G} \in \mathbf{SO}(3))$ is the rotation operator for spherical signals. As shown in Fig. 3, the spherical convolution of two signals g and ψ is computed by three steps. We first expand g and ψ to their spherical harmonic basis, then compute the pointwise product of harmonic coefficients, and finally invert the spherical harmonic expansion.

Same as in [8], we leverage a feature clustering operation to convert the output of spherical convolution into a viewpoint-invariant place descriptor. Intuitively, there exists spatial similarity in output local descriptors of spherical convolution. Therefore, we cluster the local descriptors and take a sum of residuals (difference vector between descriptor and corresponding cluster center) as a global place descriptor. The extracted place descriptor is invariant to orientation because the unsupervised clustering property of the VLAD layer [25]. On the other hand, our multilayer spherical projections can improve the geometry feature extraction and reduce the sensitiveness to the translation differences. In experiments, we will analyze the place recognition performance of our SphereVLAD approach under variant viewpoint differences.

2) Learning Metrics: To enable the end-to-end training for our SphereVLAD module, we introduce a "Lazy Viewpoint" loss metric. For the convenience of illustrating the loss functions, the necessary definitions are first described as follows. Each training tuple in our training datasets consists of four components: $\mathcal{S} = [S_a, \{S_{\text{rot}}\}, \{S_{\text{pos}}\}, \{S_{\text{neg}}\}]$, where S_a is the spherical projections of the local 3-D scan onto the ground truth position. $\{S_{\text{rot}}\}$ is a set of spherical representations of 3-D scans manually rotated from $\{S_a\}$, where the rotation angles are random sampled from $([0°, 30°, \ldots 330°])$. $\{S_{\text{pos}}\}$ denotes a set of spherical representations of 3-D scans ("positive"), whose distance to $\{S_a\}$ is within the threshold D_{pos}, and $\{S_{\text{neg}}\}$ denotes a set of 3-D scans ("negative") whose distance to $\{S_a\}$ is beyond D_{net}. In our applications, we set the threshold $D_{\text{pos}} = 5$˜m and $D_{\text{neg}} = 20$˜m. Ideally, we want to minimize two distances, i.e., $\delta_{\text{pos}_i} = d(f(S_a), f(S_{\text{pos}_i}))$ and $\delta^{\text{rot}_j}_{\text{pos}_i} = d(f(S_{\text{rot}_j}), f(S_{\text{pos}_i}))$, while maximizing two distances, i.e., $\delta_{\text{neg}_i} = d(f(S_a), f(S_{\text{neg}_i}))$ and $\delta^{\text{rot}_j}_{\text{neg}_i} = d(f(S_{\text{rot}_j}), f(S_{\text{neg}_i}))$. Here, $S_{\text{rot}_j} \in \{S_{\text{rot}}\}$, $S_{\text{pos}_i} \in \{S_{\text{pos}}\}$ and $S_{\text{neg}_i} \in \{S_{\text{neg}}\}$. $f(.)$ is the function that encodes spherical representations into global descriptors by SphereVLAD, and $d(\cdot)$ denotes the Euclidean distance.

We apply a "Lazy Viewpoint" loss to minimize the distance between $f(S_a)$ and $f(S_{\text{pos}_i})$ and maximize the distance between $f(S_a)$ and $f(S_{\text{neg}_j})$, which is written as

$$L_{\text{Viewpoint}}(\mathcal{T}) = \max_{i,j}([\gamma + \delta_{\text{pos}_i} - \delta_{\text{neg}_j}]_+) + \max_{i,j,k}([\alpha + \delta^{\text{rot}_j}_{\text{pos}_i} - \delta^{\text{rot}_j}_{\text{neg}_k}]_+) \qquad (3)$$

where $[.]_+$ denotes the hinge loss, and γ and α are the constant thresholds to control the margins between $\delta_{\text{pos}_i} \sim \delta_{\text{neg}_j}$ and $\delta^{\text{rot}_j}_{\text{pos}_i} \sim \delta^{\text{rot}_j}_{\text{neg}_k}$, respectively. In our application, both γ and α are set to 0.5.

C. Fast Matching

Given the extracted viewpoint-invariant place descriptor, we apply a Fast-Matching approach to improve place recognition accuracy against the measurement noise. As shown in Fig. 4, given a sequence of global reference descriptors S_g and a sequence of temporary descriptors S_l, we calculate feature differences based on features' Euclidean distances. The proposed Fast-Matching method can locate the best match via a hierarchical searching manner. This searching manner can balance the searching efficiency and accuracy.

Since our Fast-Matching approach follows the transitional particle filter framework, we will introduce the particle initialization, particle/map updating, and complexity analysis, respectively.

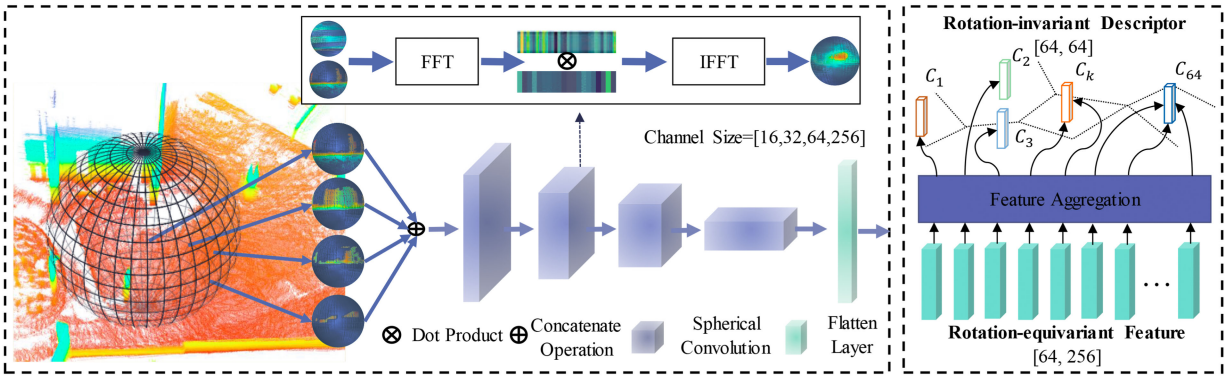

Fig. 3. Network structure of SphereVLAD. Given multilayer spherical perspectives, SphereVLAD can obtain orientation-equivariant local features through the spherical convolution in the harmonic domain and then transform them into viewpoint-invariant features via feature aggregation. Such features are designed to be invariant to heading and roll/pitch differences.

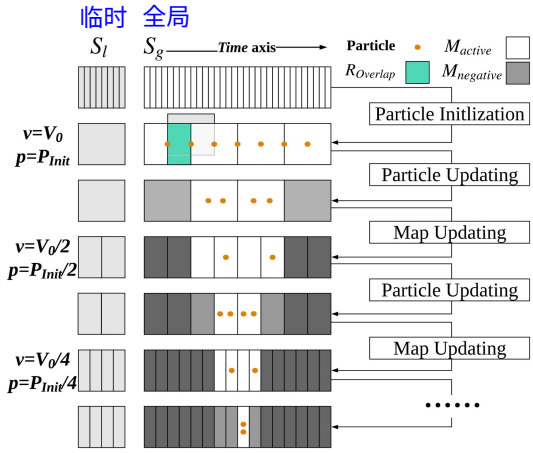

Fig. 4. Fast-Matching method. v is the frame downsampling interval, and p is the number of particles. An area will be marked as "negative" when no active particles within this area. After particles converged on $v = V_0$, new particles are sampled from active area on $v = \frac{V_0}{2}$ level.

1) Particle Initialization: We first define a skipping interval $v = V_0$, i.e., every v frames to take a place descriptor, as shown in the second row of Fig. 4. The particles are generated uniformly within the reference descriptors S_g, where each particle represents a potential match between S_l and S_g. At the lowest resolution level, particles are sampled uniformly along the whole frame sequence, and the sequence length for each particle is $\frac{S_l}{V_0}$. We define an overlapped ratio $R_{\text{Overlap}} \in [0, 1]$ to control the overlapping ratio between two neighbor particles. Then, the initial number of particles P_{init} can be estimated by

$$P_{\text{init}} = \frac{M}{N} \cdot \frac{1}{1 - R_{\text{Overlap}}} = \frac{M}{N}\tau \tag{4}$$

where M and N are the sequence length of reference frames O_g and local frames O_l, respectively. When $R_{\text{Overlap}} = 50\%$, initial particles are $\tau = 2$ times of $\frac{M}{N}$. The entire particle sets have the following format:

$$P = \{p_t^{[1]}, p_t^{[2]}, p_t^{[3]}, \ldots, p_t^{[P_{init}]}\}$$
$$p_t^i = [id_t^i, w_t^i] \tag{5}$$

where id_t^i and w_t^i represent the index of predicted reference sequence and its corresponding weight for particle p_t^i, respectively.

2) Particle and Map Updating: For each particle, we evaluate its corresponding matching score by following the SeqSLAM [11] procedure. Refer to the original paper for detailed explanation. The new particle weighting is obtained by $\hat{\omega}_k^i = \omega_{k-1}^i \times \frac{1}{1+e^{-\text{score}_i}}$. After updating all particles, the particles' weights are further updated with a normalization operation $\omega_k^i = \frac{\hat{\omega}_k^i}{\sum \hat{\omega}_k^i}$. Based on the new particles' weighting, the effectiveness score of new particles P is calculated by $\hat{N}_{\text{eff}} = 1/(\sum(\omega_k^i)^2)$. If $\hat{N}_{\text{eff}}$ is smaller than the given threshold $\text{thresh}_{\text{eff}}$, resampling on the new particles' distribution will be triggered.

As shown in the third row of Fig. 4, the particles will converge to potential matching targets. We determine whether to change the sequence resolution level by evaluating an active coverage score $M_{\text{cover}} = \frac{M_{\text{active}}}{M_{\text{active}} + M_{\text{negative}}}$. If the convergence rate satisfies $M_{\text{cover}} \leq 50\%$, sequences S_{lt} and S_{gr} will be updated into a higher resolution level. Note that we will not generate new particles within the negative areas, and only half of the particles will be kept to avoid the increasing computation consumption for a single particle.

3) Complexity Analysis: Given M reference frames and N temporary frames, for SeqSLAM, the complexity is $O(MN)$. In map resolution level i with P_{init} initial particles, the complexity of our method is $O(\frac{P_{\text{init}}}{2^i} N_i)$, where N_i is the number of testing frames on the ith resolution level. Assume that $l_{\max}$ is the maximum resolution level; we will have $N_i = \frac{N}{2^{l_{\max}-i}}$ testing frames. Then, we have

$$\begin{aligned} \mathcal{C}_{\frac{\text{Seq}}{\text{MRS}}} &= \frac{O(MN)}{O\left(\sum_{i=0}^{l_{\max}} \frac{P_{\text{init}}}{2^i} \cdot N_i\right)} \\ &= \frac{O(MN)}{O\left(\sum_{i=0}^{l_{\max}} \frac{1}{2^i} \cdot \frac{M}{N} \cdot \frac{1}{1-R_{\text{overlap}}} \cdot \frac{N}{2^{l_{\max}-i}}\right)} \\ &= N \cdot (1 - R_{\text{overlap}}) \cdot \frac{2^{l_{\max}}}{l_{\max}} \end{aligned} \tag{6}$$

where $\mathcal{C}_{\frac{\text{Seq}}{\text{MRS}}}$ is the computation complexity ratio between SeqSLAM and our method. If we set $l_{\max} = 3$ and $R_{\text{overlap}} = 0.5$,

Fig. 5. Data recording platform.

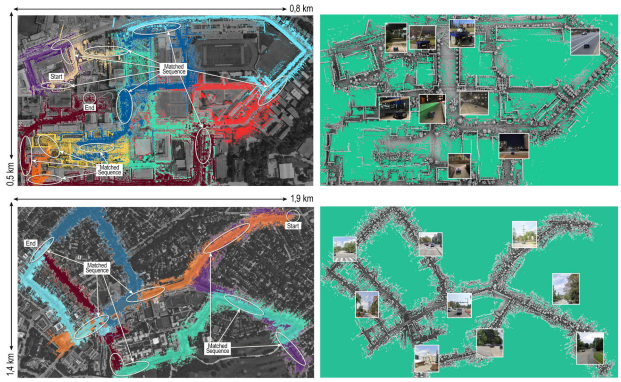

Fig. 6. Data collection for campus and city.

the computation complexity ratio will be $1.33N$. Assume that $N = 50$; ideally, we can speed up by 66.5 times.

IV. Experiments

In this section, we compare the proposed method with current arts in learning-based 3-D place recognition on both public and self-recorded datasets. To record our own datasets, we designed a data collection platform, as shown in Fig. 5, which contains a LiDAR device (Velodyne-VLP 16), an inertial measurement unit (Xsense MTi 30, 0.5° error in roll/pitch, 1° error in yaw, 550 mW), a mini computer ($i7$ Intel NUC i7, 3.5 GHz, 28 W), and a Nivida AGX Xavier (32-GB memory, 30 W). All training and evaluation experiments are conducted on two 1080Ti GPUs with 64-G memory.

A. Dataset Overview

Our experiment is performed on three datasets.

1) *KITTI [13]:* The odometry dataset consists of 21 trajectories generated with Velodyne-64 LiDAR scanner around the mid-size city of Karlsruhe. We use trajectory $\{1–8\}$ for network training and $\{9, 10\}$ for evaluation.
2) *Campus dataset:* We created a *Campus* dataset with 11 trajectories with our recording platform by traversing a 2-km outdoor route in the campus. We use trajectories $\{1–9\}$ for network training and $\{10, 11\}$ for evaluation.
3) *City dataset:* We created a City dataset by mounting the data recording module on the top of a car and traverse 11-km trajectories in the city. We use trajectories $\{1–10\}$ for network training and $\{11, 12\}$ for evaluation.

In Fig. 6, we record the *Campus* and *City* datasets with the LiDAR Odometry [26]. And the ground truth on the self-gathered datasets is estimated with the General-ICP method [2]. The dataset separation for training and evaluation is shown in Table I. We trained and evaluated the performance of our proposed method in three datasets. In the evaluation step, we generate reference and testing sequences in the same trajectory under different orientations to evaluate the place recognition accuracy. Same as PointNetVLAD [8], we define the baseline and refine network with different dataset configurations to verify the matching performance. To further verify the generalization ability, we also evaluate place recognition performance with different trajectories on the campus dataset, where the testing trajectory is slightly different to the reference trajectory. We use the precision–recall curve and the average recall to quantify the place recognition accuracy.

TABLE I
Dataset Splitting for Training/Evaluation

	KITTI	*Campus*	*City*
Training (baseline)	12, 587	13, 682	16, 458
Training (refine)	13, 287	14, 519	17, 826
Evaluation (baseline)	2, 434	3, 512	3, 638
Evaluation (refine)	1, 269	2, 037	2, 392

TABLE II
Average Recall (%) @1% on Different Datasets

	KITTI	*Campus*	*Campus-R*	*City*
PN-STD	0.46	4.20	4.15	3.79
PN-MAX	0.69	2.75	2.64	7.38
PN-VLAD baseline	13.75	17.88	16.17	15.96
PN-VLAD refine	18.93	32.11	32.08	31.16
SPH-VLAD baseline	77.91	89.28	85.19	79.06
SPH-VLAD refine	88.63	91.40	88.28	81.58
PN-STD (seq)	2.27	8.64	8.23	5.76
PN-MAX (seq)	3.02	9.69	9.19	8.15
PN-VLAD baseline (seq)	34.31	20.07	19.54	23.82
PN-VLAD refine (seq)	43.54	56.25	55.87	46.12
SPH-VLAD baseline (seq)	99.70	98.82	96.28	97.01
SPH-VLAD refine (seq)	**99.93**	**98.88**	**98.21**	**99.04**

Note "(seq)" represents sequence matching version.

B. Place Recognition on Single/Sequence Matching

Fig. 7 and Table II show the comparison between single frame matching results and sequence-matching results. For each dataset, we analyze SphereVLAD (SPH-VLAD), original PointNetVLAD (PN-VLAD), PointNet with the max-pool layer (PN-MAX) and PointNet trained with object classification in ModelNet (PN-STD) [7], and the PN-VLAD refined version with the same configuration as in [8]. *Campus-R* in Table II is place recognition results on the *Campus* dataset but under

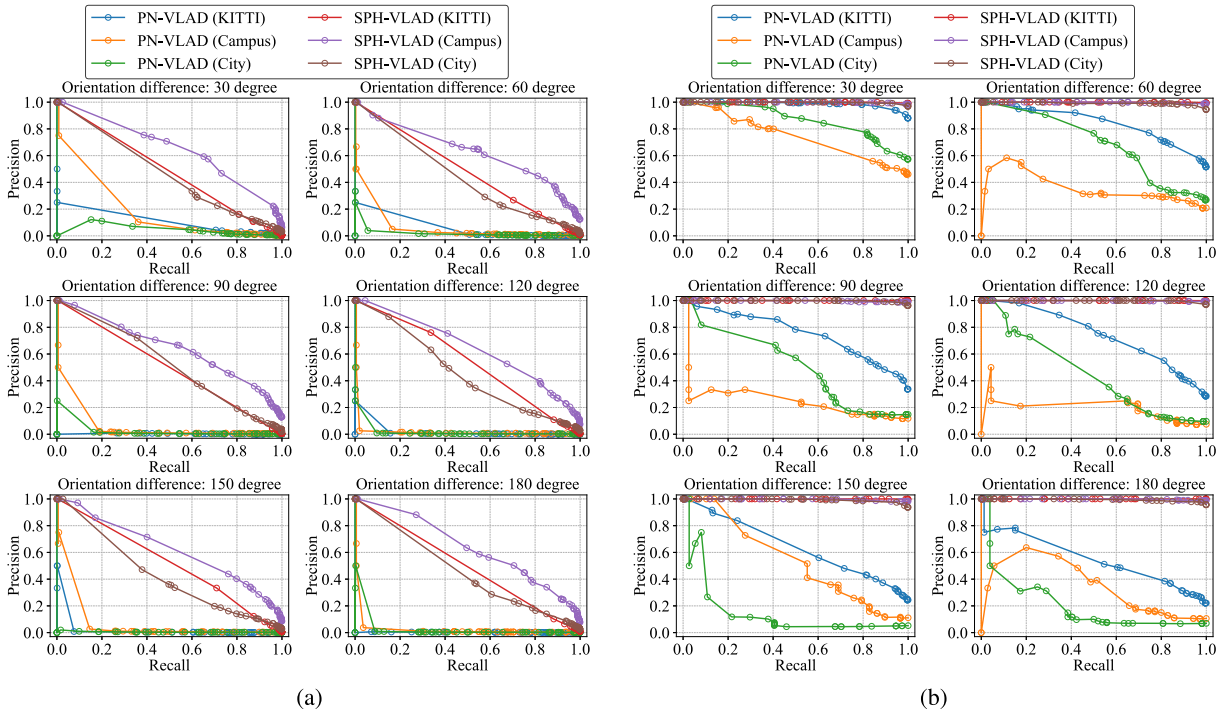

Fig. 7. Precision–recall curves of single frame matching and sequence matching. For both single frame matching and sequence matching, SphereVLAD shows better place retrieval performance than state-of-the-art PointNetVLAD under all six orientation different cases in *KITTI* dataset, *Campus* dataset, and *City* dataset. (a) Single frame matching. (b) Sequence matching.

different real trajectories, whose average translation/orientation differences are within $[-1, 1]$ m and $[-10, 10]^\circ$, respectively. Our method shows robust place recognition performance under various orientation differences. Furthermore, compared to single scan matching, the sequence-matching mechanism can further improve the matching accuracy. The standard sequence matching based on the burst-force searching is accurate but time-consuming. In the next subsection, we will further analyze the matching efficiency of our Fast-Matching and standard sequence matching.

C. Efficiency Analysis

Compared with the original burst-force sequence-matching method SeqSLAM [11], deeper-resolution-based coarse-to-fine searching can improve the initial estimation for the best sequence matching and reduce the matching time. However, in the lowest resolution level case, each particle's sequence features may fail to find the initial estimation. Another critical parameter in the Fast-Matching is τ, which determines the initial number of particles, as shown in (4). As observed in Fig. 8, with the increasing of τ, the particle effectiveness index $\hat{N}_{\text{eff}}$ of first particle updating decreases, which means that there will be more particles converging to the potential optimal. To balance both efficiency and accuracy, we set τ within $(1.5, 2.5)$, depending on the requirement of efficiency. In our experiment, the default τ value is 2.0, i.e., the overlapping ratio between two neighbor particle is $R_{\text{Overlap}} = 66.6\%$. To sum up, with a Fast-Matching approach, we can balance efficiency and accuracy.

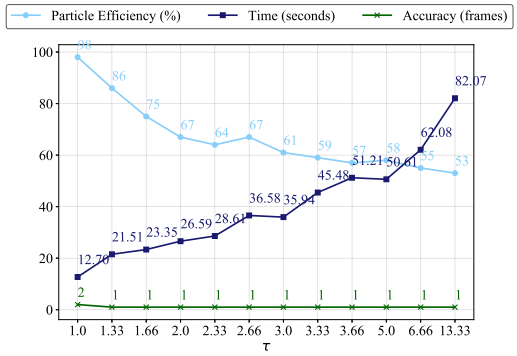

Fig. 8. Matching performance under different overlap area configurations τ for reference sequence $O_r = 9000$ and testing sequence $O_l = 300$. Increase τ will increase sequence-matching time and stabilize the matching performance.

TABLE III
COMPARISON RESULT OF TIME AND MEMORY REQUIREMENTS OF POINTNETVLAD AND SPHEREVLAD

Method	Training GPU memory	Run-time per frame
PointNetVLAD	$7711M$	$55.00ms$
SphereVLAD	**2459M**	**10.50ms**

Table III shows the GPU memory usages and runtimes in the training procedure of our SphereVLAD method and PointNetVLAD. And our method takes only 10.5 ms time for extracting place descriptor for a local 3-D map. The lightweight framework of our method enable its employing on real robots.

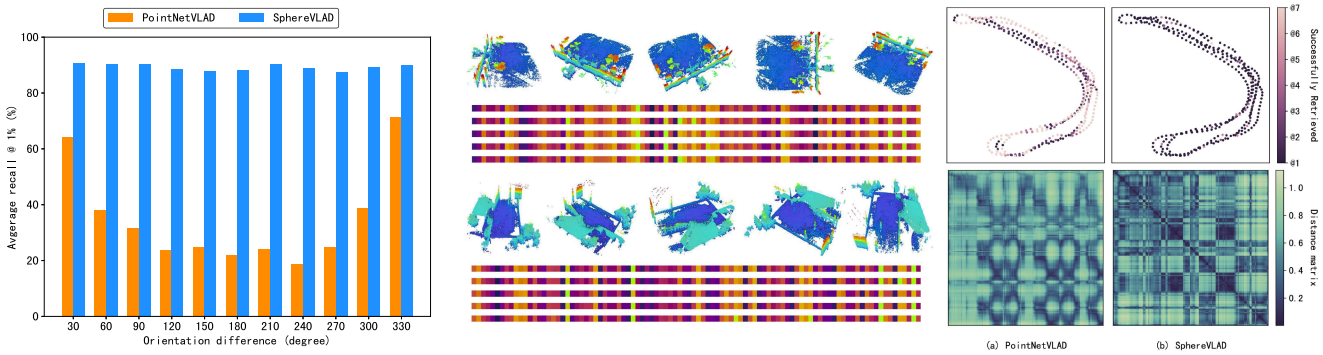

Fig. 9. Place recognition performance under various orientation difference. (Left) Average recall (%) of PointNetVLAD and SphereVLAD at top 1% (@1%) under different orientations (30–330°). (Middle) SphereVLAD features for the same place under different orientations ($[36, 72, \ldots, 360]^\circ$). (Right) Retrieved map and sequence feature similarity of different methods under random roll (−10–10°) and pitch (−10–10°) difference. @k in retrieved map means this location is successfully retrieved within at least k attempts.

TABLE IV
TOP 1% RECALL OF DIFFERENT METHODS IN THREE DATASETS UNDER RANDOM ORIENTATION yaw $\in [-30\text{–}30^\circ]$

Method	*KITTI*	*Campus*	*City*
PointNetVLAD	18.9%	32.1%	21.2%
LPDNet	20.1%	33.5%	24.4%
PCAN	19.8%	31.7%	23.9%
SphereVLAD(our)	**88.6%**	**73.7%**	**82.6%**

TABLE V
TOP 1 RECALL OF DIFFERENT METHODS ON KITTI SEQUENCE 10

Method	Standard	With ROT	With ROT and TRANS
ScanContext	76.19%	**73.81%**	55.56%
OverlapNet	**89.68%**	3.97%	0.79%
SphereVLAD (our)	70.4%	66.4%	**63.2%**

"ROT" denotes the random orientation difference on roll, pitch $\in [-10 \sim 10^\circ]$, yaw $\in [-15 \sim 15^\circ]$; "TRANS" denotes the random translation difference on x, $y \in [-1, 1]$.

TABLE VI
TOP 1% RECALL OF DIFFERENT CONFIGURATIONS IN SPHEREVLAD'S MULTILAYER SPHERICAL PROJECTIONS EVALUATED UNDER FIXED TRANSLATION (M) OR ORIENTATION (°) DIFFERENCES

Multi-layer	T(5), R(10)	T(5), R(90)	T(10), R(90)
L=1, C={dis,alpha}	68.7%	62.2%	58.9%
L=2, C={dis,alpha}	76.1%	71.5%	64.3%
L=3, C={dis,alpha}	80.5%	79.3%	72.5%
L=4, C={dis,alpha}	**83.1%**	**82.9%**	**78.3%**
L=4, C={dis}	71.2%	70.3%	65.6%

D. Viewpoint-Invariant Analysis

To analyze the place recognition performance under different viewpoints, we compare it with the original PointNetVLAD [8], LPDNet [9], and PCAN [10]. For both LPDNet and PCAN, we use their official implementation on the Github.[1][2] For each dataset, we randomly add orientation difference (yaw $\in [-30\text{–}30^\circ]$) between reference and testing point clouds. Table IV shows the top 1% recall of different methods. It demonstrates that both LPDNet and PCAN are pretty sensitive to orientation difference. On the contrary, our SphereVLAD outperforms all the point-based methods and achieves robust viewpoint-invariant place recognition performance in different datasets.

We further analyze place recognition performance of PointNetVLAD and SphereVLAD on the *Campus* dataset as depicted in Fig. 9. The left figure shows the average recall at the top 1% under various orientation differences. We can see that the matching accuracy of PointNetVLAD quickly declines as rotation difference increases, while SphereVLAD can still guarantee a relatively stable matching accuracy. The middle figure shows that the SphereVLAD features of point clouds belonging to the same place are nearly invariant to input orientations. The right figure shows the retrieved map and sequence feature similarity under random roll (−10–10°) and pitch (−10–10°) differences. We also present the comparison results of our method with ScanContext [19] and OverlapNet [16] in Table V and Fig. 10. We conduct experiments in three experimental setups: standard, with "ROT," and with "ROT/TRANS" on KITTI sequence 10. Under standard manner, the Top 1 accuracy of our approach is worse than others; this is due to the low-resolution inputs for spherical convolutions. However, our method shows more stable performance under variant translation/orientation differences.

E. Place Recognition With Different Multilayer Projection

As depicted in Section III-A, we generate multiple-layer spherical representations to capture geometric information of points within different distance range.

For each layer, we apply two channels on the spherical grid $[\theta_i, \phi_j]$, i.e., the distance channel d_{θ_i,ϕ_j} and the orientation-equivalent surface angles α_{θ_i,ϕ_j}. This subsection further investigates the place recognition performance with different multilayer configurations on the *Campus* dataset. As we can see in Table VI, with the same channel configuration, rich layer configuration can further improve the robustness to local translation difference. And with the same number of layers, the

[1][Online]. Available: https://github.com/Suoivy/LPD-net
[2][Online]. Available: https://github.com/XLechter/PCAN

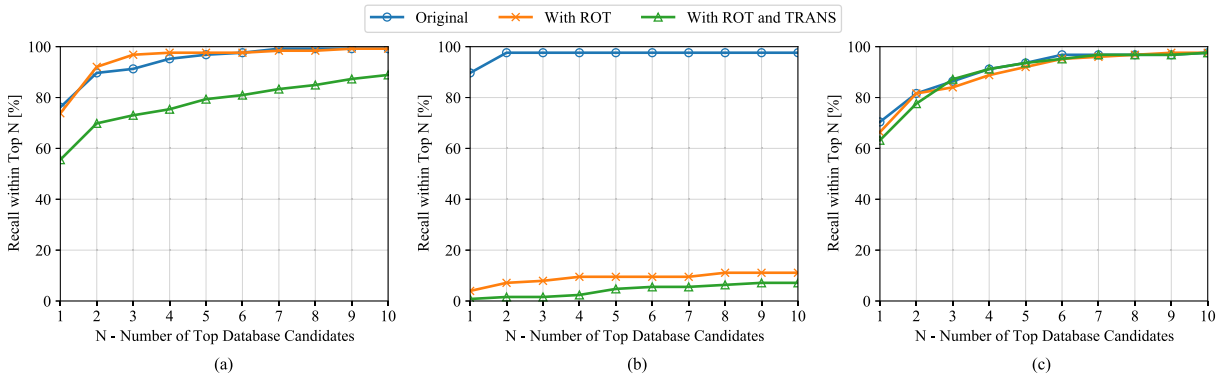

Fig. 10. Average recall of different methods on KITTI sequence 00. "ROT" denotes the random orientation difference of roll,pitch $\in [-10°, 10°]$, yaw $\in [-15°, 15°]$; "TRANS" denotes the random translation difference of $x, y \in [-1, 1]$. (a) ScanContext. (b) OverlapNet. (c) SphereVLAD (our).

additional orientation-equivalent surface angles α_{θ_i,ϕ_j} can help SphereVLAD learn more geometric features, which benefits matching robustness to both translation and orientation differences.

V. Conclusion

In this article, we proposed a Fast-Matching enhanced viewpoint-invariant 3-D place recognition method. Within this framework, we designed the SphereVLAD, which can extract viewpoint-invariant place descriptors from spherical representations of raw point clouds. Given extracted place descriptors, we developed a coarse-to-fine sequence-matching approach to balance the place recognition accuracy and efficiency. The results on both public and self-recorded datasets showed that our method notably outperforms state of the arts in 3-D-point-cloud-based place recognition tasks. We also evaluated the method on our data recording platform; the place recognition ability showed great robustness translation and orientation differences. On the other hand, the lightweight network structure of our method also enabled the large-scale localization task for lower cost mobile robots. In our future work, we will improve the place recognition accuracy by updating our current spherical convolution with higher resolution inputs.

References

[1] C. Cadena *et al.*, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *IEEE Trans. Robot.*, vol. 32, no. 6, pp. 1309–1332, Dec. 2016.

[2] A. Segal, D. Haehnel, and S. Thrun, "Generalized-ICP," in *Proc. Robot.: Sci. Syst. Conf.,* Seattle, WA, USA, vol. 2, 2009, p. 435.

[3] P. Mondal, J. Mukhopadhyay, S. Sural, and P. P. Bhattacharyya, "3D-SIFT feature based brain atlas generation: An application to early diagnosis of Alzheimer's disease," in *Proc. Int. Conf. Med. Imag., m-Health Emerg. Commun. Syst.,* Nov. 2014, pp. 342–347.

[4] Y. Mei and Y. He, "A new spin-image based 3D map registration algorithm using low-dimensional feature space," in *IEEE Int. Conf. Inf. Autom.*, Aug. 2013, pp. 545–551.

[5] R. Rusu, G. Bradski, R. Thibaux, and J. Hsu, "Fast 3D recognition and pose using the viewpoint feature histogram," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.* 2010, pp. 2155–2162.

[6] X. Han, S. Sun, X. Song, and G. Xiao, "3D point cloud descriptors in hand-crafted and deep learning age: State-of-the-art," in *Proc. Comput. Vis. Pattern Recognit.*, 2018.

[7] C. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 652–660.

[8] M. A. Uy and G. H. Lee, "PointNetVLAD: Deep point cloud based retrieval for large-scale place recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4470–4479.

[9] Z. Liu *et al.*, "LPD-Net: 3D point cloud learning for large-scale place recognition and environment analysis," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2019, pp. 2831–2840.

[10] W. Zhang and C. Xiao, "PCAN: 3D attention map learning using contextual information for point cloud based retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 12436–12445.

[11] M. Milford and G. Wyeth, "SeqSLAM: Visual route-based navigation for sunny summer days and stormy winter nights," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2012, pp. 1643–1649.

[12] Z. Chen *et al.*, "Deep learning features at scale for visual place recognition," in *Proc. IEEE Int. Conf. Robot. Autom.*, 2017, pp. 3223–3230.

[13] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *Int. J. Robot. Res.*, vol. 32, no. 11, pp. 1231–1237, 2013.

[14] N. Carlevaris-Bianco, A. K. Ushani, and R. M. Eustice, "University of Michigan North Campus long-term vision and LIDAR dataset," *Int. J. Robot. Res.*, vol. 35, no. 9, pp. 1023–1035, 2016.

[15] H. Yin, Y. Wang, X. Ding, L. Tang, S. Huang, and R. Xiong, "3D LiDAR-based global localization using siamese neural network," *IEEE Trans. Intell. Transp. Syst.*, vol. 21, no. 4, pp. 1380–1392, Apr. 2020.

[16] X. Chen *et al.*, "OverlapNet: Loop closing for LiDAR-based SLAM," in *Proc. Robot.: Sci. Syst. Conf.*, 2020.

[17] C. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 5099–5108.

[18] Z. Liu *et al.*, "SeqLPD: Sequence matching enhanced loop-closure detection based on large-scale point cloud description for self-driving vehicles," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2019, pp. 1218–1223.

[19] G. Kim and A. Kim, "Scan context: Egocentric spatial descriptor for place recognition within 3D point cloud map," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, 2018, pp. 4802–4809.

[20] C. Esteves, C. Allen-Blanchette, A. Makadia, and K. Daniilidis, "Learning SO (3) equivariant representations with spherical CNNs," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 52–68.

[21] H. Jegou, F. Perronnin, M. Douze, J. Sánchez, P. Perez, and C. Schmid, "Aggregating local image descriptors into compact codes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1704–1716, Sep. 2011.

[22] M. Nowakowski, C. Joly, S. Dalibard, N. Garcia, and F. Moutarde, "Topological localization using Wi-Fi and vision merged into FABMAP framework," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Sep. 2017, pp. 3339–3344.

[23] S. Siam and H. Zhang, "Fast-SeqSLAM: A fast appearance based place recognition algorithm," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2017, pp. 5702–5708.

[24] T. Cohen, M. Geiger, J. Köhler, and M. Welling, "Spherical CNNS," 2018, *arXiv:1801.10130*.

[25] R. Arandjelovic, P. Gronat, A. Torii, T. Pajdla, and J. Sivic, "NetVLAD: CNN architecture for weakly supervised place recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 5297–5307.

[26] J. Zhang and S. Singh, "LOAM: Lidar odometry and mapping in real-time," in *Proc. Robot.: Sci. Syst. Conf.*, vol. 2, 2014., p. 9.

Peng Yin received the bachelor's degree from the Harbin Institute of Technology, Harbin, China, in 2013, and the Ph.D. degree from the University of Chinese Academy of Sciences, Beijing, China, in 2018.

He is currently a Postdoctoral Researcher with the Department of the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA. His research interests include LiDAR simultaneous localization and mapping, place recognition, 3-D perception, and reinforcement learning.

Dr. Yin has served as a Reviewer for the IEEE International Conference on Robotics and Automation, the IEEE/RSJ International Conference on Intelligent Robots and Systems, and the American Control Conference.

Fuying Wang received the bachelor's degree in electronic engineering from Tsinghua University, Beijing, China.

From 2019 to 2020, he was a Visiting Research Assistant with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA. He is currently a Research Assistant with Air lab, Carnegie Mellon University. His research interests include 3-D visual learning and reasoning, robot navigation, and reinforcement learning.

Anton Egorov received the B.S. degree in electronics engineering from Chuvash State University, Cheboksary, Russia, in 2018, and the M.S. degree in space and engineering systems from the Skolkovo Institute of Science and Technology, Moscow, Russia, in 2020.

From 2019 to 2020, he was a Visiting Student with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA. He is currently a Research Engineer with the Autonomous Transportation Systems Laboratory, Innopolis University, Innopolis, Russia. His current research interests include LiDAR simultaneous localization and mapping, robotics perception, and deep learning.

Jiafan Hou received the bachelor's degree from the Chinese University of Hong Kong, Shenzhen, China, in 2020.

She is currently a Research Assistant with the Robotics and Artificial Intelligence Laboratory, Chinese University of Hong Kong. Her research interests include LiDAR simultaneous localization and mapping, place recognition, perception, and reinforcement learning.

Zhenzhong Jia (Member, IEEE) received the B.E. and M.E. degrees in mechanical engineering from Tsinghua University, Beijing, China, and the M.S. degrees in applied math and mechanical engineering and the Ph.D. degree in naval architecture and marine engineering (focus in controls), all from the University of Michigan, Ann Arbor, MI, USA.

From 2014 to 2018, he was a Postdoctoral Fellow with the Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA. He is currently an Assistant Professor with the Southern University of Science and Technology, Shenzhen, China. His research interests include robot mobility (Mars/Lunar rover, intelligent driving under extreme conditions), robotic manipulation, and related topics on perception, planning, control, and learning.

Jianda Han (Member, IEEE) was born in Liaoning, China, in 1968. He received the Ph.D. degree from the Harbin Institute of Technology, Harbin, China, in 1998.

He is currently a Professor and the Vice Director of the State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences, Shenyang, China. His research interests include nonlinear estimation and control, robotics, and mechatronics systems.